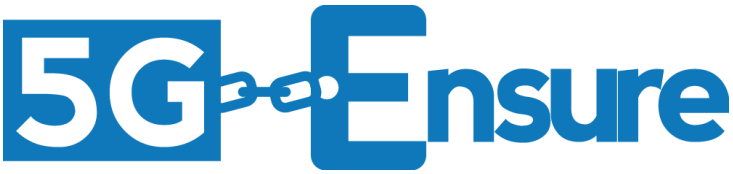


Deliverable D4.3

Test plan (final): Final description of how to evaluate the selected security enablers

Project name	5G Enablers for Network and System Security and Resilience	
Short name	5G-ENSURE	
Grant agreement	671562	
Call	H2020-ICT-2014-2	
Delivery date	20.02.2018	
Dissemination Level:	Public	
Lead beneficiary	PARTNER	Sergio Morant, Sergio.morant@b-com.com
Authors	b<>com: Michel Corriou, Sergio Morant Orange: José Sanchez, Jean-Philippe Wary TIIT: Madalina Baltatu TS: Edith Felix, Pascal Bisson	

Executive summary

5G-ENSURE aims at providing security proven enablers. In order to achieve this goal, a testbed has been designed within the scope of the project to host the enablers issued from the project. The enabler's security claims will be tested against the security threats previously identified within the project. This will prove efficiency of the features developed.

This document version provides the test plan version containing the basis to build the complete test plan, the procedures to deliver and integrate the software, the integration roadmap, the procedure to evaluate the coverage of evaluation scenarios, and the R1 evaluation scenarios validated by the project partners (described in chapter §4).

Another WP4 deliverable will arrive afterwards to provide the analysis results of the test plan execution (D4.4).

The D4.3 Test Plan document embeds the final version of the "Testbed Terms of Use", reviewed and accepted by all partners.

Foreword

5G-ENSURE belongs to the first group of EU-funded projects which collaboratively develop 5G under the umbrella of the 5G Infrastructure Public Private Partnership (5G-PPP) in the Horizon 2020 Programme. The overall goal of 5G-ENSURE is to deliver strategic impact across technology and business enablement, standardisation and vision for a secure, resilient and viable 5G network. The project covers research & innovation - from technical solutions (5G security architecture and testbed with 5G security enablers) to market validation and stakeholders' engagement - spanning various application domains.

This document provides the procedures to deliver and integrate the enablers issued from the 5G-ENSURE Software Releases into the testbed. It also establishes the test plan that will allow the validation of the enabler's security claims against the identified security threats.

An important document, the **Testbed Terms of Use** (final version)**,** is annexed as it provides the rules applied to the testbed usage that need to be respected by each partner working on the testbed.

Disclaimer

Copyright notice

Contents

Abbreviations 7
1 Introduction 8
1.1 Definitions 8
1.1.1 Enabler evaluation Scenario 8
2 Security requirements to cover use case needs 8
2.1 Enabler's delivered software features 9
2.1.1 Release R1 9
2.1.2 Release R2 10
2.2 Relevant use cases covered by each feature 12
2.2.1 Release R1 12
2.2.2 Release R2 13
2.3 Enabler's security claims against use cases 15
2.3.1 Release R1 15
2.3.2 Release R2 17
3 Enabler's integration roadmap 20
3.1 Release R1 20
3.2 Release R2 21
4 Testing procedures for the testbed 22
4.1 Enabler testbed lifecycle 22
4.2 Enabler deployment strategy 22
4.2.1 Delivery process 23
4.2.2 Integration workflow 25
4.3 Delivering an enabler on the catalogue 27
4.4 Running an enabler security evaluation 32
4.4.1 Overview 32
4.4.2 Scenario validation process 35
4.5 Project's evaluation metric definitions 38
5 Test plan 39
5.1 Roles 39
5.1.1 Role matching 39
5.1.2 Role endorsement 41
5.2 Structure 41

5.2.1 Enabler's feature sanity checks 42
5.2.2 Enabler's security evaluation Scenarios 47
5.2.3 Releases management 52
5.3 Test plan design and execution 53
5.3.1 Design 53
5.3.2 Execution 54
6 Conclusions 55
References 56
A Testbed Terms of Use 59
B Ansible roles for testbed enabler deployment 64
C Test plan design: Enabler's security evaluation (R1) 67
Use Cases cluster 1 - Identity Management 67
Use Cases cluster 2 - Enhanced Identity Protection and Authentication 73
Use Cases cluster 3 - IoT Device Authentication and Key Management 80
Use Cases cluster 5 - Software-Defined Networks, Virtualization and Monitor 88
Use Cases cluster 8 - Ultra-Reliable and Standalone Operations 116
Use Cases cluster 9 - Trusted Core Network and Interconnect 125
D Test plan design: Enabler's security evaluation (R2) 128
Use Cases cluster 1 - Identity Management 128
Use Cases cluster 2 - Enhanced Identity Protection and Authentication 137
Use Cases cluster 3 - IoT Device Authentication and Key Management 143
Use Cases cluster 5 - Software-Defined Networks, Virtualization and Monitor 147
Use Cases cluster 8 - Ultra-Reliable and Standalone Operations 176
Use Cases cluster 9 - Trusted Core Network and Interconnect 185
Use Cases cluster 10 - 5G Enhanced Security Services 187
E ANNEX : integration of enablers R1 189
F ANNEX : integration of enablers R2 190

Abbreviations

5G-PPP	5G Infrastructure Public Private Partnership
AAA	Authentication Authorisation Accounting
Dx.y	Deliverable x.y
E.O	Enabler Owner
ETSI	European Telecommunications Standards Institute
ID	Identifier
MANO	MANagement and Orchestration
NFV	Network Function Virtualisation
SDN	Software-Defined Networking
UC	Use Case
VNF	Virtualised Network Function
VPN	Virtual Private Network

1 Introduction

This deliverable covers the aspects of enabler's deployment and evaluation within the 5G-ENSURE testbed. This includes an analysis of the security requirements to cover the security use cases described in D2.1 [1]. This analysis consists in matching the enabler's security claims described in the enabler's open specifications (D3.2 [2] and D3.6 [3]) of each enabler's feature, against the different use cases defined in D2.1 [1], and their associated security threats identified in D2.3 [4]. In order to achieve the enabler's integration on the testbed, the integration roadmaps for the different enabler releases (i.e. R1 and R2) are provided, together with the procedures allowing to deploy and evaluate the enablers.

The test plan structure for the integration and evaluation of the enabler's security claims against the security use cases are provided as well. This includes the process defined to exchange information with other WPs (namely WP2 and WP3) and the test plan designs.

This deliverable provides the required documentation to evaluate the enablers through the features scheduled over the two major releases. This deliverable provides the test plan structures and the different test plan designs. The evaluation results from the test plan execution and the result analysis will be provided at the end of the project (M24) on the D4.4 "*Evaluation of the security enablers: Results and analysis of the Testbed runs*".

Annexed to this document is the final version of **Testbed Terms of Use (Annex A)** that has been signed by all the partners willing or having to use the testbed.

This document is based on outcomes from previous project's deliverables, and so naming or identification convention is thus re-used. This includes preserving the structure and identification used to organize the enablers and their features (D3.2 [2] and D3.6 [3]), the use cases (D2.1 [1]), and the threats (D2.3 [4]).

As a matter of fact, this document follows D4.2 "Test plan (draft)" that it extends and completes. All information present on the previous document is preserved and enriched, whenever necessary, with the new project outcomes about the testbed activities.

1.1 Definitions

Most of the definitions used in this document have been already defined in the Chapter 1 of D4.1 [5]. In this section only new definitions are added

1.1.1 Enabler evaluation Scenario

Description of (technical or theoretical) steps required to provide evidence for some claim.
For instance, a Scenario (set of technical steps description) is used by an Enabler Owner to demonstrate that its enabler features covers a specific threat.

Note: A Scenario never mitigates a threat, only an enabler feature mitigates a threat. A Scenario is used to validate and demonstrate that the enabler's threat coverage is more or less effective.

2 Security requirements to cover use case needs

This section shows the security requirements to cover the different needs with regard to the use cases. The section is structured in three subsections, the enablers to be delivered in both releases, the relevant use cases covered by those enablers, and their security claims.

2.1 Enabler's delivered software features

This section details the corresponding enablers and features to be delivered in both releases of the project, R1 and R2.

2.1.1 Release R1

Table 1 gathers all the enablers and their corresponding features initially planned to be integrated in the R1. Note that those enablers and features are classified as a function of the security group they belong, namely AAA, Privacy, Trust, Security monitoring, and network management & virtualisation isolation.

This classification determines the indexing of the enablers and corresponding features, where the indexing of the features and enablers will be consistent throughout this deliverable.

The list of enablers and security features is taken from the deliverable D3.1 "5G-PPP security enablers technical road map" [D3.1], where enablers and features are defined to be integrated in the 5G testbed R1.

Table 1: Initially planned set of Enabler's to be delivered in R1

Id	Security Group	Owner	5G-ENSURE security enablers	Features for 1st sw release (R1)
1	AAA	SICS	1.1 Internet of things (IoT)	1.1.1 Group authentication by extending the LTE-AKA protocol
		TASE	1.2 Fine-grained Authorization	1.2.1: Basic Authorization in Satellite systems 1.2.2: Basic distributed authorization Enforcement for RCDs
2	Privacy	TIIT	2.1 Privacy Enhanced Identity Protection	2.1.1: Encryption of Long Term Identifiers **(**IMSI public-key based encryption**)**
		OXFORD	2.2 Device identifier(s) privacy	2.2.1: Enhanced privacy for network attachment protocols
3	Trust	TCS	3.1 VNF Certification	3.1.1: VNF Trustworthiness Evaluation
		VTT	3.2 Trust Metric	3.2.1: Trust metric based network domain security policy management
		IT-INNOV	3.3 Trust Builder	3.3.1: 5G Asset Model 3.3.2: 5G Threat knowledge base v1
4	Security Monitoring	ORANGE	4.1 Generic Collector Interface	4.1.1: Log and Event Processing
		VTT	4.2 Security Monitor for 5G Micro-Segments	4.2.1: Complex Event Processing Framework for Security Monitoring and Inferencing
		TASE	4.3 Satellite Network Monitoring	4.3.1: Pseudo real-time monitoring 4.3.2 : Threat detection
		TS	4.4 PulSAR: Proactive Security Analysis and Remediation	4.4.1: 5G specific vulnerability schema
		IT-INNOV	4.5 System security state repository	4.5.1 : Deployment model ontology
5	Network Management and Virtualization Isolation	NEC	5.1 Access Control Mechanisms	5.1.1: Southbound Reference Monitor
		NEC	5.2 Component-Interaction Audits	5.2.1: Basic OpenFlow Compliance Checker
		SICS	5.3 Bootstrapping Trust	5.3.1 Integrity Attestation of virtual network components
		VTT	5.4 Micro Segmentation	5.4.1: Dynamic Arrangement of Micro-Segments

Table 2 gathers all those enablers and corresponding features which are candidate to be integrated in the Testbed. This means that all those enablers and features have triggered an enabler deployment ticket in Helpdesk. Those enablers and features which did not trigger the corresponding helpdesk ticked are not considered candidate and are discarded from the integration process and are depicted in red in the table. We can see that 15 enablers out of the initially planned 16 enablers are actual candidate to be integrated over the 5G security testbed, on the condition that the integration and evaluation process is successful. This number of candidate enablers corresponds to 18 candidate features.

Table 2: Candidate to be delivered enabler software features for R1

Id	Security Group	Owner	Helpdesk ticket	5G-ENSURE security enablers	Features for 1st sw release (R1)
1	AAA	SICS	YES	1.1 Internet of things (IoT)	1.1.1 Group authentication by extending the LTE-AKA protocol
		TASE	YES	1.2 Fine-grained Authorization	1.2.1: Basic Authorization in Satellite systems 1.2.2: Basic distributed authorization Enforcement for RCDs
2	Privacy	TIIT	YES	2.1 Privacy Enhanced Identity Protection	2.1.1: Encryption of Long Term Identifiers **(**IMSI public-key based encryption**)**
		OXFORD	YES	2.2 Device identifier(s) privacy	2.2.1: Enhanced privacy for network attachment protocols
3	Trust	TCS	YES	3.1 VNF Certification	3.1.1: VNF Trustworthiness Evaluation
		VTT	YES	3.2 Trust Metric	3.2.1: Trust metric based network domain security policy management
		IT-INNOV	YES	3.3 Trust Builder	3.3.1: 5G Asset Model 3.3.2: 5G Threat knowledge base v1
4	Security Monitoring	ORANGE	YES	4.1 Generic Collector Interface	4.1.1: Log and Event Processing
		VTT	YES	4.2 Security Monitor for 5G Micro-Segments	4.2.1: Complex Event Processing Framework for Security Monitoring and Inferencing
		TASE	YES	4.3 Satellite Network Monitoring	4.3.1: Pseudo real-time monitoring 4.3.2 : Threat detection
		TS	YES	4.4 PulSAR: Proactive Security Analysis and Remediation	4.4.1: 5G specific vulnerability schema
		IT-INNOV	NO	4.5 System security state repository	4.5.1 : Deployment model ontology
5	Network Management and Virtualization Isolation	NEC	YES	5.1 Access Control Mechanisms	5.1.1: Southbound Reference Monitor
		NEC	YES	5.2 Component-Interaction Audits	5.2.1: Basic OpenFlow Compliance Checker
		SICS	YES	5.3 Bootstrapping Trust	5.3.1 Integrity Attestation of virtual network components
		VTT	YES	5.4 Micro Segmentation	5.4.1: Dynamic Arrangement of Micro-Segments

2.1.2 Release R2

Table 3 provides the enablers and their corresponding features initially planned to be integrated in the R2.

Table 3: Initially planned set of Enabler's to be delivered in R2

Id	Security Group	Owner	5G-ENSURE security enablers	Features for 2nd sw release (R2)
1	AAA	SICS	1.1 Internet of things	1.1.1 : Group based AKA (R1/R2) 1.1.2 : Non-USIM based AKA (R2) 1.1.3 : BYOI (Bring Your Own Identity) (R2) 1.1.4 : vGBA (Vertical GBA) (R2)
		TASE	1.2 Fine-grained Authorization	1.2.3 : AAA integration with satellite systems (R2) 1.2.4 : Authorization and authentication for RCD based on ongoing IETF standardization (R2)
		SICS		
		EAB	1.3 Basic AAA enabler	1.3.1: Forward Secrecy (R1/R2) 1.3.2 : AAA aspects of trusted micro-segmentation (R1 /R2) 1.3.3 : Trusted interconnect and authorization (R2)
		TS	1.4 Federative authentication and identification enabler	1.4.1 :Storage of authentication level (R2) 1.4.2 :Usage of authentication level (R2)
2	Privacy	TIIT	2.1 Privacy Enhanced Identity Protection	2.1.2 :Home Network centric IMSI protection (R2) 2.1.3 :IMSI Pseudonymization (R2)
		OXFORD	2.2 Device identifier(s) privacy	2.2.2 :Anonymous and optimised address selection for network attachment protocols (R2)
		TIIT	2.3 Device-based Anonymization	2.3.1 :Format preserving anonymization algorithm (R2) 2.3.2 :Privacy configuration (R2)
		IT-INNOV	2.4 Privacy policy analysis	2.4.1 :Privacy policy specification (R2) 2.4.2 :Privacy preferences specification (R2) 2.4.3 :Comparison of policies and preferences (R2)

3	Trust	TCS	3.1 VNF Certification	3.1.1 :VNF Trustworthiness Evaluation (R1) 3.1.2 :VNF Trustworthiness Certification (R2)
		VTT	3.2 Trust Metric Enabler	3.2.1 :Improved trust metric based on extended data (R2)
		IT-INNOV	3.3 Trust Builder	3.3.2 :Graphical editor (R1/R2) 3.3.3:5G Threat knowledgebase (R2)
		IT-INNOV	3.4 Security Indicator	3.4.1.:Security indicator subscriber display (R2)
		ORANGE	3.5 Reputation based on Root Cause Analysis for SDN	3.5.1 :Root Cause Analysis for SDN (R2)
4	Security Monitoring	ORANGE	4.1 Generic Collector Interface	4.1.1: Log and Event Processing (R1)
		VTT	4.2 Security Monitor for 5G Micro-Segments	4.2.2 :Risk-based adaptation of micro-segments (R2) 4.2.3:Extended data gathering (R2) 4.2.4 :Cross-domain information exchange (R2)
		TASE	4.3 Satellite Network Monitoring	4.3.3 :Active security analysis (R2) 4.3.4 :Pre-emptive mitigation security actions (R2)
		TS	4.4 PulSAR: Proactive Security Analysis and Remediation	4.4.2 :5G specific vulnerability schema implementation (R2) 4.4.3 :PulSAR interface with Generic Collector (R2)
		ITINNOV	4.5 System Security State Repository	4.5.2 :System Security State Repository service (R2)
		NIXU	4.6 Malicious Traffic generation	4.6.1 Traffic generator engine 4.6.2 Malicious pattern library 4.6.3 Fuzzing engine
5	Network Management and Virtualization Isolation	NEC	5.1 Access Control Mechanisms	5.1.2 :Access Requirements for VNF Container Resources (R2)
		NEC	5.2 Component-Interaction Audits	5.2.2 :Basic NFV Reconfiguration Compliance Checker (R2)
		SICS	5.3 Bootstrapping Trust	5.3.2 :Integrity Attestation of VNFs running in Docker containers (R2)
		VTT	5.4 Micro Segmentation	5.4.2 :Extended Northbound API (R2) 5.4.3 :Support for multi-domain micro-segments (R2)
		NEC	5.5 Anti-Fingerprinting	5.5.1 :Controller-Switch-Interaction Imitator (R2)
		TCS	5.6 Flow Control: in-network Threat Detection and Mitigation for Critical Functions in Virtual Networks	5.6.1 :Detection of malicious behaviours in virtual networks (R2) 5.6.2 :Mitigation of detected network threats (R2)

As in the previous section, Table 4 gathers all those enablers and corresponding features candidate to be integrated in the Testbed. We can see that 16 enablers out of the initially planned 25 enablers are actual candidate to be integrated over the 5G security testbed, on the condition that the integration and evaluation process is successful. This number of candidate enablers corresponds to 30 candidate features.

Table 4: Candidate to be delivered enabler software features for R2

Id	Security Group	Owner	Helpdesk ticket	5G-ENSURE security enablers	Features for 2nd sw release (R2)
1	AAA	SICS	YES	1.1 Internet of things	1.1.1 : Group based AKA (R1/R2) 1.1.2 : Non-USIM based AKA (R2) 1.1.3 : BYOI (Bring Your Own Identity) (R2) 1.1.4 : vGBA (Vertical GBA) (R2)
		TASE	YES	1.2 Fine-grained Authorization	1.2.3 : AAA integration with satellite systems (R2) 1.2.4 : Authorization and authentication for RCD based on ongoing IETF standardization (R2)
		SICS	YES		
		EAB	NO	1.3 Basic AAA enabler	1.3.1: Forward Secrecy (R1/R2) 1.3.2 : AAA aspects of trusted micro-segmentation (R1 /R2) 1.3.3 : Trusted interconnect and authorization (R2)
		TS	NO	1.4 Federative authentication and identification enabler	1.4.1 :Storage of authentication level (R2) 1.4.2 :Usage of authentication level (R2)
2	Privacy	TIIT	YES	2.1 Privacy Enhanced Identity Protection	2.1.2 :Home Network centric IMSI protection (R2) 2.1.3 :IMSI Pseudonymization (R2)
		OXFORD	YES	2.2 Device identifier(s) privacy	2.2.2 :Anonymous and optimised address selection for network attachment protocols (R2)
		TIIT	NO	2.3 Device-based Anonymization	2.3.1 :Format preserving anonymization algorithm (R2) 2.3.2 :Privacy configuration (R2)

		IT-INNOV	YES	2.4 Privacy policy analysis	2.4.1 :Privacy policy specification (R2) 2.4.2 :Privacy preferences specification (R2) 2.4.3 :Comparison of policies and preferences (R2)
3	Trust	TCS	YES	3.1 VNF Certification	3.1.1 :VNF Trustworthiness Evaluation (R1) 3.1.2 :VNF Trustworthiness Certification (R2)
		VTT	YES	3.2 Trust Metric Enabler	3.2.1 :Improved trust metric based on extended data (R2)
		IT-INNOV	YES	3.3 Trust Builder	3.3.2 :Graphical editor (R1/R2) 3.3.3:5G Threat knowledgebase (R2)
		IT-INNOV	NO	3.4 Security Indicator	3.4.1.:Security indicator subscriber display (R2)
		ORANGE	NO	3.5 Reputation based on Root Cause Analysis for SDN	3.5.1 :Root Cause Analysis for SDN (R2)
4	Security Monitoring	ORANGE	NO	4.1 Generic Collector Interface	4.1.1: Log and Event Processing (R1)
		VTT	YES	4.2 Security Monitor for 5G Micro-Segments	4.2.2 :Risk-based adaptation of micro-segments (R2) 4.2.3:Extended data gathering (R2) 4.2.4 :Cross-domain information exchange (R2)
		TASE	YES	4.3 Satellite Network Monitoring	4.3.3 :Active security analysis (R2) 4.3.4 :Pre-emptive mitigation security actions (R2)
		TS	YES	4.4 PulSAR: Proactive Security Analysis and Remediation	4.4.2 :5G specific vulnerability schema implementation (R2) 4.4.3 :PulSAR interface with Generic Collector (R2)
		ITINNOV	NO	4.5 System Security State Repository	4.5.2 :System Security State Repository service (R2)
		NIXU	NO	4.6 Malicious Traffic generation	4.6.1 Traffic generator engine 4.6.2 Malicious pattern library 4.6.3 Fuzzing engine
5	Network Management and Virtualization Isolation	NEC	NO	5.1 Access Control Mechanisms	5.1.2 :Access Requirements for VNF Container Resources (R2)
		NEC	YES	5.2 Component-Interaction Audits	5.2.2 :Basic NFV Reconfiguration Compliance Checker (R2)
		SICS	YES	5.3 Bootstrapping Trust	5.3.2 :Integrity Attestation of VNFs running in Docker containers (R2)
		VTT	YES	5.4 Micro Segmentation	5.4.2 :Extended Northbound API (R2) 5.4.3 :Support for multi-domain micro-segments (R2)
		NEC	NO	5.5 Anti-Fingerprinting	5.5.1 :Controller-Switch-Interaction Imitator (R2)
		TCS	YES	5.6 Flow Control: in-network Threat Detection and Mitigation for Critical Functions in Virtual Networks	5.6.1 :Detection of malicious behaviours in virtual networks (R2) 5.6.2 :Mitigation of detected network threats (R2)

2.2 Relevant use cases covered by each feature

This section details the use cases to be covered by each proposed feature. This section is also split in the two releases, R1 and R2.

2.2.1 Release R1

In this section we present the relationships between each of the enablers' feature to be integrated in the testbed and the Use Cases (UCs). The indexing of the features is the same as shown in the previous section and will be consistent throughout this deliverable to ensure the coherence.

Each enabler's feature of this table is coming from D3.2 "Security enablers open specifications (early) and is related to the different use cases as specified in D3.1 "5G-PPP security enablers technical roadmap" [6].

Table 5: Relevant use cases covered by each feature in R1

ID Feature	Relevant use cases
1.1.1	UC3.1 : Authentication of IoT Devices in 5G
1.2.1	UC1.3 : Satellite Identity Management for 5G Access
1.2.2	UC4.1 : Authorization in Resource-Constrained Devices Supported by 5G Network

2.1.1	UC2.2: Subscriber Identity Privacy UC2.3: Enhanced Communication Privacy
2.2.1	UC2.1: Device Identity Privacy UC2.2: Subscriber Identity Privacy
3.1.1	UC5.2: Adding a 5G Node to a Virtualized Core Network UC5.4: Verification of the Virtualized Node and the Virtualization Platform UC5.5: Control and Monitoring of Slice by Service Provider UC9.3: Authentication of New Network Elements
3.2.1	UC5.2: Adding a 5G Node to a Virtualized Core Network UC 5.5: Control and Monitoring of Slice by Service Provider UC7.1: Unprotected Mobility Management Exposes Network for Denial of Service UC9.1: Alternative Roaming in 5G
3.3.1 3.3.2	UC1.1 : Factory Device Identity Management for 5G Access UC3.1 : Authentication of IoT Devices in 5G UC3.2 : Network-Based Key Management for End-to-End Security UC5.1 : Virtualized Core Networks, and Network Slicing UC9.3 : Authentication of New Network Elements UC11 .1: Lawful Interception in a Dynamic 5G Network UC11.2: End-to-end Encryption in LI-aware network
4.1.1	UC5.1: Virtualized Core Networks, and Network Slicing UC5.4: Verification of the Virtualized Node and the Virtualization Platform UC5.5: Control and Monitoring of Slice by Service Provider UC5.6: Integrated Satellite and Terrestrial Systems Monitor UC7.1: Unprotected Mobility Management Exposes Network for Denial of Service UC8.1: Satellite-Capable eNB UC8.2: Standalone EPC UC9.3: Authentication of New Network Elements UC10.1: Botnet Mitigation UC10.2: Privacy Violation Mitigation UC11.1: Lawful Interception in a Dynamic 5G Network UC8.2: Standalone EPC UC10.1: Botnet Mitigation UC10.2: Privacy Violation Mitigation UC11.1: Lawful Interception in a Dynamic 5G Network
4.2.1	UC5.5: Control and Monitoring of Slice by Service Provider UC10.1: Botnet Mitigation
4.3.1 4.3.2	UC5.6: Integrated Satellite and Terrestrial Systems Monitor UC8.1: Satellite-Capable eNB
4.4.1	UC5.5: Control and Monitoring of Slice by Service Provider
4.5.1	UC5.1: Virtualized Core Networks, and Network Slicing UC5.4: Verification of the Virtualized Node and the Virtualization Platform UC5.5: Control and Monitoring of Slice by Service Provider
5.1.1	UC4.2: Authorization for end-to-end IP connections UC5.2: Adding a 5G node to a virtualized core network UC9.3: Authentication of new network elements UC11.1: Lawful interception in a dynamic 5G network
5.2.1	UC5.2: Adding a 5G node to a virtualized core network UC5.4: Verification of the virtualized node and the virtualization platform UC9.3: Authentication of new network elements UC11.1: Lawful interception in a dynamic 5G network
5.3.1	UC5.1: Virtualized core networks, and network slicing UC5.2: Adding a 5G node to a virtualized core network UC5.4: Verification of the virtualized node and the virtualization platform UC9.3: Authentication of new network elements
5.4.1	UC5.1: Virtualized core networks and network slicing UC5.2: Adding a 5G node to a virtualized core network UC3.1: Authentication of IoT devices in 5G UC3.2: Network-based key management for end-to-end security UC1.3: Satellite identity management for 5G access

2.2.2 Release R2

The table 6 shows the relevant use cases covered by the features to be integrated in R2. The use cases were extracted from the documents with the open specifications in D3.6 [3].

Each enabler's feature of this table is coming from D3.6 "Security enablers open specifications (early) [3] and is related to the different use cases as specified in D3.5 "5G-PPP security enablers technical roadmap - Update" [7].

Table 6: Relevant D2.1 use cases covered by R2 features

ID Feature	Relevant Use Cases
1.1.1	UC3.1: Authentication of IoT Devices in 5G
1.1.2	UC1.2 : Using Enterprise Identity Management for Bootstrapping 5G Access
1.1.3	UC 1.1:Factory Device Identity Management for 5G Access UC1.2: Using Enterprise Identity Management for Bootstrapping 5G Access
1.1.4	UC3.1 : Authentication of IoT Devices in 5G
1.2.3	UC3.1 : Authentication of IoT Devices in 5G
1.2.4	UC3.1 : Authentication of IoT Devices in 5G
1.3.1	UC2.3: Enhanced Communication Privacy
1.3.2	UC5.1: Virtualized Core Networks, and Network Slicing
1.3.3	UC9.3: Authentication of new network elements
1.4.1	UC 1.4 : MNO Identity Management Service
1.4.2	UC 1.4 : MNO Identity Management Service
2.1.2	UC2.2: Subscriber Identity Privacy
2.1.3	UC2.2: Subscriber Identity Privacy
2.2.2	UC2.1: Device Identity Privacy
2.3.1	UC 10.3: SIM-based and/or Device-based Anonymization
2.3.2	UC10.3: SIM-based and/or Device-based Anonymization
2.4.1	UC10.2: Privacy Violation Mitigation
2.4.2	UC10.2: Privacy Violation Mitigation
2.4.3	UC10.2: Privacy Violation Mitigation
3.1.1	UC5.2: Adding a 5G Node to a Virtualized Core Network UC5.4: Verification of the Virtualized Node and the Virtualization Platform
3.1.2	UC5.2: Adding a 5G Node to a Virtualized Core Network UC5.4: Verification of the Virtualized Node and the Virtualization Platform
3.2.1	UC3.1: Authentication of IoT Devices in 5G UC5.5: Control and Monitoring of Slice by Service Provider
3.3.2 3.3.3	UC1.1 : Factory Device Identity Management for 5G Access UC3.1 : Authentication of IoT Devices in 5G UC3.2 : Network-Based Key Management for End-to-End Security UC5.1 : Virtualized Core Networks, and Network Slicing UC9.3 : Authentication of New Network Elements UC11 .1: Lawful Interception in a Dynamic 5G Network UC11.2: End-to-end Encryption in LI-aware network
3.4.1	UC9.3: Authentication of New Network Elements UC10.2: Privacy Violation Mitigation UC2.3: Enhanced Communication Privacy
3.5.1	UC5.1 : Virtualized Core Networks, and Network Slicing
4.2.2	UC5.5: Control and Monitoring of Slice by Service Provider
4.2.3	UC5.4: Verification of the Virtualized Node and the Virtualization Platform
4.2.4	UC5.5: Control and Monitoring of Slice by Service Provider
4.3.3	UC 5.6: Integrated Satellite and Terrestrial Systems Monitor
4.3.4	UC 5.6: Integrated Satellite and Terrestrial Systems Monitor
4.4.2	UC5.1: Virtualized Core Networks, and Network Slicing UC5.5: Control and Monitoring of Slice by Service Provider
4.4.3	UC5.5: Control and Monitoring of Slice by Service Provider
4.5.2	UC5.1: Virtualized Core Networks, and Network Slicing UC5.4: Verification of the Virtualized Node and the Virtualization Platform UC5.5: Control and Monitoring of Slice by Service Provider

4.6.1	UC5.3: Reactive Traffic Routing in a Virtualized Core Network; network reconfiguration messages UC5.4: Verification of the Virtualized Node and the Virtualization Platform; attacking the SDN controller UC6.1: Attach Request During Overload UC7.1: Unprotected Mobility Management Exposes Network for Denial of Service
4.6.2	UC1.1: Factory Device Identity Management for 5G Access UC1.2: Using Enterprise Identity Management for Bootstrapping 5G Access UC1.3: Satellite Identity Management for 5G Access UC1.4: MNO Identity Management Service UC4.1: Authorization in Resource-Constrained Devices Supported by 5G Network; attacks against the AAA servers' vulnerabilities UC4.2: Authorization for End-to-End IP Connections; direct IP connection without authorization UC4.3: Vehicle-to-Everything (V2X) UC5.1: Virtualized Core Networks, and Network Slicing; significant attack surface UC5.2: Adding a 5G Node to a Virtualized Core Network UC5.5: Control and Monitoring of Slice by Service Provider UC5.6: Integrated Satellite and Terrestrial Systems Monitor; signalling messages outside of the normal sequences UC6.2: Unprotected User Plane on Radio Interface UC9.1: Alternative Roaming in 5G; spoofing of signalling messages UC9.2: Privacy in Context-Aware Services; User traffic can be enriched in various ways UC9.3: Authentication of New Network Elements
4.6.3	Generic use case: Any use case may contain a vulnerability when the interface is flooded with random traffic, the aim is to investigate whether an interface is vulnerable to the interface overload.
5.1.2	UC5.2:Adding a 5G node to a virtualized core network
5.2.2	UC5.2:Adding a 5G node to a virtualized core network UC5.4: Verification of the virtualized node and the virtualization platform
5.3.2	UC5.1:Virtualized Core Networks, and Network
5.4.2	UC5.1: Virtualized core networks and network slicing UC5.2: Adding a 5G node to a virtualized core network UC5.5: Control and Monitoring of Slice by Service Provider
5.4.3	UC5.1: Virtualized core networks and network slicing UC 5.2: Adding a 5G node to a virtualized core network
5.5.1	UC5.3: Reactive traffic routing in a virtualized core network
5.6.1	UC5.4: Verification of the Virtualized Node and the Virtualization Platform
5.6.2	UC5.4: Verification of the Virtualized Node and the Virtualization Platform

2.3 Enabler's security claims against use cases

This section presents the security claims done by the enabler owners to mitigate the threats in the different use cases. This section is split in the two releases R1 and R2.

2.3.1 Release R1

This section contains all the features per enabler to be integrated in the 5G-ENSURE platform in R1 as well as the goal of each feature and a detailed description of those threats covered by the features. The threats listed in this table are with regard to the identified relevant use cases shown in the previous section.

Table 7: R1 Enabler's security claims against use cases

ID Feature	Goal	Covered Threats
1.1.1	*Enable 5G to support massive deployments of IoT devices by adding explicit support for group authentication of devices .*	• **T_UC3.1_1 :** Authentication traffic spikes • **T_UC1.4_1 :** Compromised Data. • **T_UC2.2_2 :** Mobile user interception and information interception
1.2.1	*To support access control of multiple users with different rights in satellite devices and services.*	• **T_UC1.3_1 :** Unauthorized activities related to satellite devices or (satellite) network resources

		• **T_UC1.3_2 :** Fake roaming from terrestrial network into satellite network • **T_UC5.6_1 :** Security threats in a satellite network
1.2.2	*To support access control on RCDs based on existing http solutions using ABAC and adapted for these devices.*	• **T_UC3.1_1 :** Authentication traffic spikes • **T_UC3.1_2 :** Compromised authentication gateway • **T_UC4.1_1 :** Unauthorized data access • **T_UC5.6_1 :** Security threats in a satellite network
2.1.1	*Limit (preferably totally avoid) exposing user identities on (at least) the air interface*	• **T_UC2.2_1** : Tracking of device's (user's) location. • **T_UC2.2_2** : Mobile user interception and information interception
2.2.1	*Limit exposure of device identifiers and prior points of attachment, and therefore, limit the ability to track a device.*	• **T_UC2.1_1 :** Tracking of device's (user's) location • **T_UC2.1_2 :** Mobile user interception and information interception
3.1.1	*Certify the trustworthy implementation of the VNF and to expose their characteristics through a Digital Trustworthiness Certificate.*	• **T_UC5.2_1 :** Add malicious nodes into core network • **T_UC5.2_2 :** Forwarding logic leakage • **T_UC5.2_3 :** Manipulation of forwarding logic • **T_UC5.5_1 :** Misuse of open control and monitoring interfaces • **T_UC5.5_2 :** Unauthorized access to a network slice • **T_UC5.5_3 :** Bogus monitoring data • **T_UC5.5_4 :** No control of Cyber-attacks by the Service providers • **T_UC9.3_1 :** Hardening or patching of systems is not done • **T_UC9.3_2 :** Unauthentic device installed into the system
3.2.1	*Enable service providers to offer trust based services for customers in mass market and industry.*	• **T_UC3.1_1** : Authentication traffic spikes • **T_UC5.5_4 :** No control of Cyber-attacks by the Service providers
3.3.1	*Allow the modelling of 5G networks using the information gathered.*	• **T_UC3.1_1** : Authentication traffic spikes • **T_UC3.1_2** : Compromised authentication gateway • **T_UC3.2_1 :** Leaking keys • **T_UC5.1_1 :** Misbehaving control plane • **T_UC9.3_1 :** Hardening or patching of systems is not done • **T_UC9.3_2 :** Unauthentic device installed into the system • **T_UC11.1_1 :** Compromised / malicious LI (Lawful Interception) function • **T_UC11.2_1 :** Nefarious activities (manipulation of information, interception of information) over LI-aware network
3.3.2	*Allow the mapping of a limited subset of threats to the designed 5G system.*	• **T_UC3.1_1** : Authentication traffic spikes • **T_UC3.1_2** : Compromised authentication gateway • **T_UC3.2_1 :** Leaking keys • **T_UC5.1_1 :** Misbehaving control plane • **T_UC9.3_1 :** Hardening or patching of systems is not done • **T_UC9.3_2 :** Unauthentic device installed into the system • **T_UC11.1_1 :** Compromised / malicious LI (Lawful Interception) function • **T_UC11.2_1 :** Nefarious activities (manipulation of information, interception of information) over LI-aware network
4.4.1	*Extension of the Cyber Attack modelling.*	• **T_UC5.1_1 :** Misbehaving control plane • **T_UC5.5_1 :** Misuse of open control and monitoring interfaces
4.3.1	*Provide pseudo real-time monitoring of the satellite network*	• **T_UC5.6_1 :** Security threats in a satellite network • **T_UC8.1_1 :** Service failure over satellite capable eNB • **T_UC5.5_4 :** No control of Cyber-attacks by the Service providers • **T_UC5.5_3 :** Bogus monitoring data • **T_UC1.3_2 :** Fake roaming from terrestrial network into satellite network (and vice versa)
4.3.2	*Include rules in the monitoring system that correlate different incidents to detect specific threats and vulnerabilities in the satellite network.*	• **T_UC5.6_1 :** Security threats in a satellite network • **T_UC8.1_1 :** Service failure over satellite capable eNB • **T_UC5.5_4 :** No control of Cyber-attacks by the Service providers • **T_UC5.5_3 :** Bogus monitoring data • **T_UC1.3_2 :** Fake roaming from terrestrial network into satellite network (and vice versa)
4.1.1	*Interoperability between events and logs format, in order to allow FastData technologies to be deployed inside the 5G Network*	• **T_UC1.4_1 :** Compromised Data • **T_UC5.1_1 :** Misbehaving control plane • **T_UC7.1_1 :** Denial of service due to Unprotected Mobility Management Exposes Network • **T_UC5.5_1 :** Misuse of open control and monitoring interfaces • **T_UC5.5_3 :** Bogus monitoring data

		• **dfT_UC5.5_4 :** No control of Cyber-attacks by the Service providers • **T_UC9.3_1 :** Hardening or patching of systems is not done • **T_UC9.3_2 :** Unauthentic device installed into the system • **T_UC5.6_1 :** Security threats in a satellite network • **T_UC8.1_1 :** Service failure over satellite capable eNB • **T_UC10.2_1 :** Nefarious activities (malicious software, unauthorized activities, interception of information): privacy violations
4.2.1	*Enable distributed security monitoring and reactions to security incidents.*	• **T_UC5.1_1 :** Misbehaving control plane • **T_UC5.5_1 :** Misuse of open control and monitoring interfaces • **T_UC5.5_2 :** Unauthorized access to a network slice • **T_UC9.3_2 :** Unauthentic device installed into the system
5.1.1	*Enforce access control policies that account for the southbound API of an SDN controller.*	• **T_UC5.1_1 :** Misbehaving control plane • **T_UC5.2_1 :** Add malicious nodes into core network • **T_UC5.2_3 :** Manipulation of forwarding logic
5.2.1	*Verification of the interaction between multiple network components with respect to simple policies about the components' exchanged OpenFlow messages.*	• **T_UC5.1_1 :** Misbehaving control plane • **T_UC5.2_1 :** Add malicious nodes into core network • **T_UC5.2_3 :** Manipulation of forwarding logic
5.3.1	*Implement the strictly minimal functionality of software components and protocols necessary to validate the concept of deploying SDN components in isolated execution environments with a hardware root of trust.*	• **T_UC5.2_1 :** Add malicious nodes into core network • **T_UC9.3_2 :** Unauthentic device installed into the system
5.4.1	*Enable dynamic arrangement (create, delete) of micro-segments in the network.*	• **T_UC5.2_1 :** Add malicious nodes into core network

2.3.2 Release R2

This section contains all the features per enabler to be integrated in R2. The information to complete the table has been extracted from D2.3 [4] (threats) and D3.5 [8] (feature's goals).

Table 8: R2 Enabler's security claims against use cases

ID Feature	Goal	Covered Threats
1.1.1	*Improve the support of group authentication of IoT devices in 5G.*	• **T_UC3.1_1 :** Authentication traffic spikes • **T_UC3.1_2 :** Compromised authentication gateway • **T_UC2.2_2** : Mobile user interception and information interception
1.1.2	*Enable 5G to support massive deployments of IoT devices by adding support for alternative AKA procedures than EPS-AKA (e.g. EAP-TLS, using certificates instead of USIM, etc.).*	• **T_UC3.1_1 :** Authentication traffic spikes • **T_UC9.3_2 :** Unauthentic device installed into the system • **T_UC2.2_2** : Mobile user interception and information interception
1.1.3	*Allow enterprises that already have an existing AAA infrastructure in place for devices and/or employees to re-use pre-existing identities as a basis for 5G network access*	• **T_UC9.3_2 :** Unauthentic device installed into the system • **T_UC3.1_2 :** Compromised authentication gateway • **T_UC3.1_1 :** Authentication traffic spikes
1.1.4	*Enhance the classic GBA protocol to achieve better signalling efficiency*	• **T_UC3.1_1 :** Authentication traffic spikes • **T_UC3.1_2 :** Compromised authentication gateway
1.2.3	*To support policies for decision per user, resource and action; and integrate the authentication and authorization mechanism with the satellite system*	• **T_UC3.1_1 :** Authentication traffic spikes • **T_UC3.1_2 :** Compromised authentication gateway
1.2.4	*Enable standards-based, fine-grained access control and authentication on resource constrained devices connected at the edge via low power lossy networks.*	• **T_UC3.1_1 :** Authentication traffic spikes • **T_UC3.1_2 :** Compromised authentication gateway
1.3.1	*Limit and/or recover from impact of compromised long-term keys, preferably with backward compatibility. Provide a high-level description of which concepts to use for key agreement and authentication.*	• **T_UC2.2_1 :** Tracking of device's (user's) location • **T_UC2.2_2 :** Mobile user interception and information interception • **T_UC9.3_2 :** Unauthentic device installed into the system
1.3.2	*Provide a high-level description of micro-segmentation and its potential benefits for 5G.*	• **T_UC5.1_1 :** Misbehaving control plane
1.3.3	*Ensure authenticity of interconnecting parties, provide explicit authorization to actions with security impact*	• **T_UC9.3_1 :** Hardening or patching of systems is not done • **T_UC9.3_2 :** Unauthentic device installed into the system

1.4.1	*To store in a dedicated database the authentication level (in LDAP for example)*	• **T_UC1.4_1** : Compromised data
1.4.2	*Usage, at node level, of the authentication level*	• **T_UC1.4_1 :** Compromised data
2.1.2	*Limit (preferably totally avoid) exposing user permanent or long term identities on (at least) the air interface (i.e., in Attach requests, Identity responses).*	• **T_UC2.2_1 :** Tracking of device's (user's) location**T_UC2.2_2 :** Mobile user interception and information interception
2.1.3	*Complement the "Encryption of Long Term Identifiers" feature to totally avoid exposing user permanent or long term identities on (at least) the air interface (i.e., in Attach Requests, Identity Responses, Paging Responses) by avoiding user traceability*	• **T_UC2.2_1 :** Tracking of device's (user's) location • **T_UC2.2_2 :** Mobile user interception and information interception
2.2.2	*Enhanced address anonymity providing for protection of device identifiers and prior points of attachment, and therefore, limit the ability to track a device*	• **T_UC2.2_1 : T**racking of device's (user's) location. **T_UC2.2_2 :** Mobile user interception and information interception
2.3.1	*Provide an anonymization algorithm for data received in input (e.g., the IMSI, IMEI, telephone number, etc.), with the preservation of the input data format.*	• **T_UC10.3_1 :** Nefarious activities (manipulation of information, interception of information): personal information disclosure
2.3.2	*The mediator between the caller and the anonymizing SIM.*	• **T_UC10.3_1 :** Nefarious activities (manipulation of information, interception of information): personal information disclosure
2.4.1	*Encoding service privacy policy.*	• **T_UC10.2_1 :** Nefarious activities (malicious software, unauthorized activities, interception of information): privacy violations
2.4.2	*Encoding users' preferences.*	• **T_UC10.2_1 :** Nefarious activities (malicious software, unauthorized activities, interception of information): privacy violations
2.4.3	*Compare the selected service policies with the user's expressed preferences.*	• **T_UC10.2_1 :** Nefarious activities (malicious software, unauthorized activities, interception of information): privacy violations
3.1.1	*To certify the trustworthy implementation of the VNF and to expose their characteristics through a Digital Trustworthiness Certificate*	• **T_UC5.2_1** Add malicious nodes into core network
3.1.2	*Delivery of a trustworthy Digital Trustworthiness Certificate*	• **T_UC5.2_1** Add malicious nodes into core network
3.2.1	*Collecting monitoring data and KPI from the micro-segment and from eNodeB to enable near real-time operation*	• **T_UC3.1_1:** Authentication traffic spikes • **T_UC5.5_4 :** No control of Cyber-attacks by the Service providers
3.3.2	*Provide a tool to analyse the threats present in a 5G system design*	• **T_UC3.1_1 :** Authentication traffic spikes • **T_UC3.1_2 :** Compromised authentication gateway • **T_UC5.1_1 :** Misbehaving control plane • **T_UC9.3_1 :**Hardening or patching of systems is not done • **T_UC9.3_2 :** Unauthentic device installed into the system • **T_UC11.1_1 :** Compromised / malicious LI (Lawful Interception) function • **T_UC11.2_1 :** Nefarious activities (manipulation of information, interception of information) over LI-aware network
3.3.3	*Encode threat and trust data so that it can be inferred from the models and displayed in the modelling tool*	• **T_UC3.1_1 :** Authentication traffic spikes • **T_UC3.1_2 :** Compromised authentication gateway • **T_UC5.1_1 :**Misbehaving control plane • **T_UC9.3_1 :** Hardening or patching of systems is not done • **T_UC9.3_2 :** Unauthentic device installed into the system • **T_UC11.1_1 :**Compromised / malicious LI (Lawful Interception) function • **T_UC11.2_1 :** Nefarious activities (manipulation of information, interception of information) over LI-aware network
3.4.1	*Provide a new security indicator to be displayed to subscribers, whilst complying with operators' requirements to local regulations.*	• **T_UC9.3_1 :**Hardening or patching of systems is not done • **T_UC2.2_2 :** Mobile user interception and information interception • **T_UC10.2_1 :** Nefarious activities (malicious software, unauthorized activities, interception of information): privacy violations
3.5.1	*Reputation calculation block based on a RCA, taken into account all changes at physical and virtual resource level*	• **T_UC5.1_1 :** Misbehaving control plane
4.2.2	*Dynamic control of micro-segments topology and defences based on determined security threats and risk levels*	• **T_UC5.5_4 :** No control of Cyber-attacks by the Service providers
4.2.3	*Collecting monitoring data and KPI from the micro-segment and from eNodeB.*	• **T_UC5.5_4 :** No control of Cyber-attacks by the Service providers
4.2.4	*Exchanging monitoring data (with respect to the format described in D3.2) between the GCI enabler and micro-segmentation enabler*	• **T_UC5.5_4 :** No control of Cyber-attacks by the Service providers

4.3.3	*Provide the complete solution including the active security analysis to detect, investigate and response to the threats identified.*	• **T_UC5.6_1 :** Security threats in a satellite network
4.3.4	*Provide predictive capabilities to the system in order to execute mitigations actions before possible security threats happened.*	• **T_UC5.6_1 :** Security threats in a satellite network
4.4.2	*Extension of the Cyber-attack modelling*	• **T_UC5.1_1 :** Misbehaving control plane • **T_UC5.5_1 :** Misuse of open control and monitoring interfaces • **T_UC5.5_1 :** Misuse of open control and monitoring interfaces
4.4.3	*Provide an integration with Generic Collector enabler*	• **T_UC5.5_1 :** Misuse of open control and monitoring interfaces • **T_UC5.5_1 :** Misuse of open control and monitoring interfaces
4.5.2	*Software to create, update and query the runtime model*	• **T_UC5.1_1 :** Misbehaving control plane • **T_UC5.5_1 :** Misuse of open control and monitoring interfaces • **T_UC5.5_1 :** Misuse of open control and monitoring interfaces
4.6.1	*Generate an overload of traffic to the gNB or eNodeB in order to cause a DoS attack on the radio interface by flooding it with connects, or to generate traffic overload by a rogue gNB or eNodeB to UEs to prevent connectivity*	• **T_UC3.1_1 :** Authentication traffic spikes • **T_UC3.1_2 :** Compromised authentication gateway • **T_UC5.1_1 :** Misbehaving control plane • **T_UC9.3_2 :** Unauthentic device installed into the system
4.6.2	*Compose a collection of syntactically correct messages in protocols supported by the 5G node protocol stacks that would cause the node to either malfunction, drop, or surrender to unauthorized access.*	• **T_UC3.1_2 :** Compromised authentication gateway • **T_UC5.1_1 :** Misbehaving control plane • **T_UC9.3_2 :** Unauthentic device installed into the system • **T_UC11.1_1 :** Compromised / malicious LI (Lawful Interception) function • **T_UC5.2_1** Add malicious nodes into core network • **T_UC5.3_1** : Fingerprinting attack
4.6.3	*Generate random input to node interfaces in order to crash the interface or induce a memory leak*	• **T_UC3.1_2 :** Compromised authentication gateway • **T_UC5.1_1 :** Misbehaving control plane • **T_UC3.2_1:** Leaking keys • **T_UC5.2_2:** Forwarding logic leakage
5.1.2	*Enforce policies for containers that host VNFs and restrict their access to other network resources.*	• **T_UC5.2_1** Add malicious nodes into core network • **T_UC5.5_1** Misuse of open control and monitoring interfaces
5.2.2	*Verification of reconfigurations on NFV deployments with respect to policies or workflows.*	• **T_UC5.2_1** Add malicious nodes into core network • **T_UC5.5_1** Misuse of open control and monitoring interfaces
5.3.2	*Verification of the virtual switch configuration using trust agents running in trusted execution environments.*	• **T_UC3.2_1:** Leaking keys • **T_UC5.2_1:** Add malicious nodes into core network • **T_UC5.2_2:** Forwarding logic leakage
5.4.2	*Northbound micro-segmentation API extension*	• **T_UC5.5_4 :** No control of Cyber-attacks by the Service providers
5.4.3	*Add support for multi-domain micro-segments and include secure communication between two micro-segments (and different operators).*	• **T_UC5.5_4 :** No control of Cyber-attacks by the Service providers • **T_UC5.2_1:** Add malicious nodes into core network
5.5.1	*Prevent the leakage of timing information that would reveal whether a network packet received by a data plane component (e.g., a switch) triggers an interaction with the control plane (i.e., the SDN controller).*	• **T_UC5.3_1** : Fingerprinting attack • **T_UC5.2_2 :** Forwarding logic leakage
5.6.1	*Detection of malicious network-based attacks.*	• **T_UC5.5_4 :** No control of Cyber-attacks by the Service providers • **T_UC5.2_1:** Add malicious nodes into core network • **T_UC5.5_1** Misuse of open control and monitoring interfaces
5.6.2	*Detection of malicious network-based attacks.*	• **T_UC5.5_4 :** No control of Cyber-attacks by the Service providers • **T_UC5.2_1:** Add malicious nodes into core network

3 Enabler's integration roadmap

In this section we present the integration roadmap for releases R1 and R2.

3.1 Release R1

As stated in deliverable D4.1 [5], the enabler integration procedure is split in two: the R1 and the R1.1 (shown in Figure 1). The reason for this split, is that it is preferable to integrate the first set of enablers, which are easier to integrate, and schedule the more complex enablers once the integration process is mature enough.

The first enabler to be integrated in 5G-ENSURE testbed was the Generic Collector Interface (GCI). Indeed, this enabler collects information that will be sent to some other enablers, that is why its integration was one of the main priorities in the roadmap introduced in D4.1. The roadmap shows the 15 enablers to be integrated in R1.

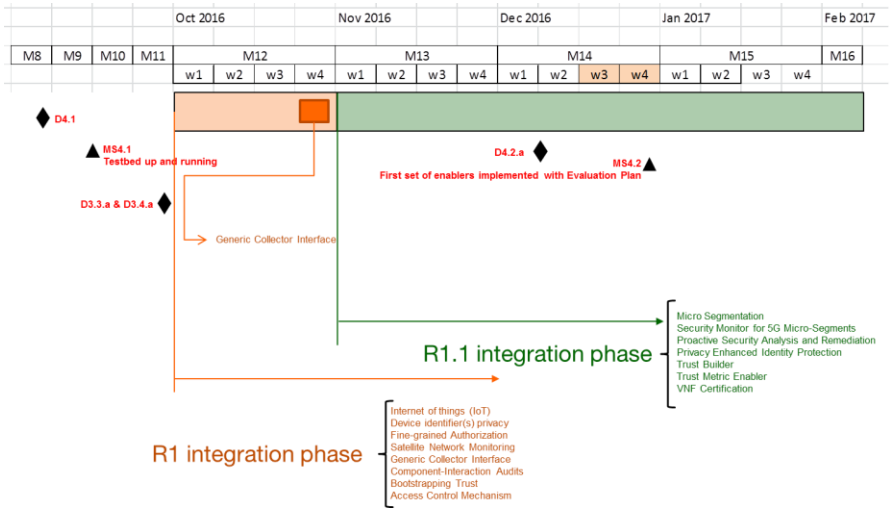


Figure 1: Initial testbed R1 integration roadmap

Figure 2 shows the final integration roadmap for enablers in R1. As it can be seen, all the features were integrated by the end of March. A total of 13 out of 15 enablers have been fully integrated (86% of integration rate).

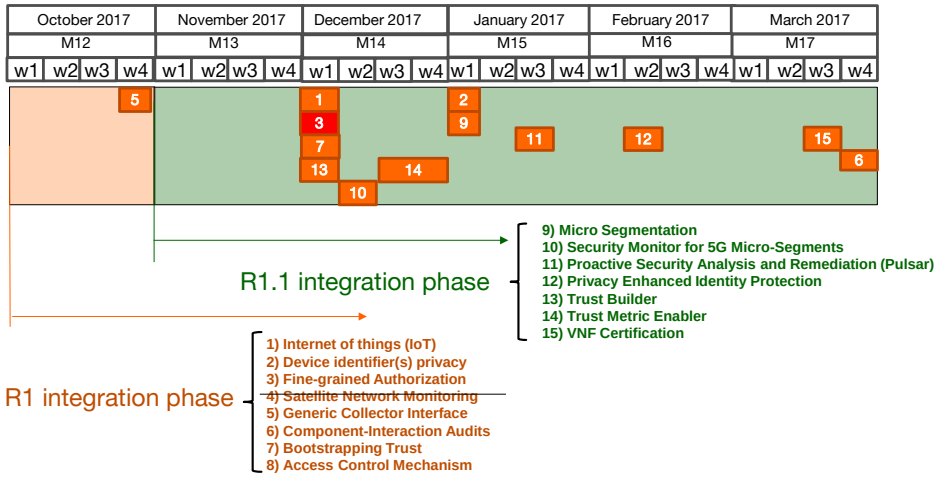


Figure 2: Final testbed R1 integration roadmap

The final status of R1 Enablers integration is given in ANNEX E. This status is based on the evidences collected from testbed tools (helpdesk, catalogue and test plan).

3.2 Release R2

Figure 3 presents the planed R2 enablers' integration on the testbed, which planes to anticipate early delivery of R2 enablers. Due to the enabler's software delivery planning with regard to the end of the project (M24 October 2017), the integration planning deadline has been set to the 31 of August. The enablers requesting integration after the deadline will be managed in Best Effort mode.

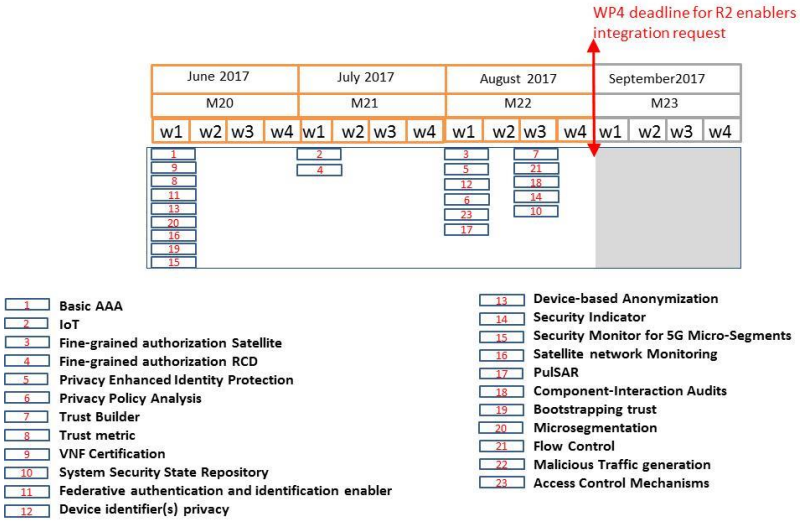


Figure 3 original R2 integration roadmap

Note : only integrated enablers could run under evaluation WP2/WP4 process.

Figure 4 shows the final integration roadmap for enablers in R2. As it can be seen, features were integrated by the end of September. A total of 9 out of 16 enablers have been fully integrated (56% of integration rate).

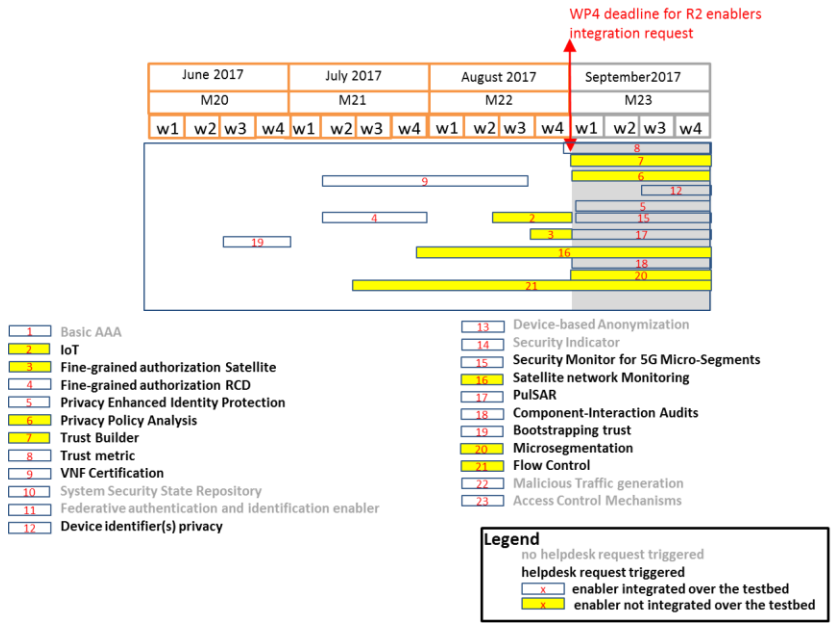


Figure 4: Final testbed R2 integration roadmap

The final status of R2 Enablers integration is given in ANNEX F. This status is based on the evidences collected from testbed tools (helpdesk, catalogue and test plan).

4 Testing procedures for the testbed

This section provides the procedures in support of the enabler testbed lifecycle. These procedures are enhanced and documented on the project's workspace (wiki). This way, the procedure is able to evolve in time without compromising the concordance with the content described in this chapter.

Notice: the tools referred in this document (TestLink, Artifactory, Ansible, etc) have already been introduced in D4.1 [5]. Please refer to this document for more detailed information about the tool description and their use in the scope of the testbed.

4.1 Enabler testbed lifecycle

The testbed lifecycle has been split in three main stages as shown on Figure 5

Figure 5: Enabler testbed lifecycle

- **Delivery** of the enabler to the testbed
- **Integration** of the enabler in the testbed allowing the assertion of enabler's testbed acceptance
- **Evaluation** of the enabler against the security threats related to the security UCs

The first two stages (delivery and integration) constitute the **deployment process** of the enabler on the testbed, which ends up with the enabler acceptance. The last stage (evaluation) allows to evaluate and grade to which extent the security claims of the enabler are covered.

The web-based TestLink [9] system is used to describe each unitary test (or acceptance test) and evaluation Scenarios. Each project entity needing to access a specific test or Scenario description should refer to TestLink.

4.2 Enabler deployment strategy

This section proposes the workflows and procedures for the delivery and integration of an enabler over the testbed as opposed to the evaluation of the enabler, which takes place later in the process and checks the coherence of the enabler with respect to the expressed requirements. The process of delivery and integration of an enabler requires the collaboration and exchange of information among several actors for an optimal result.

As stated in D4.1, "*in order to provide the required degree of conformity for a telco grade platform, the deployment of the testbed instances will be handled by the Testbed Operator who will ensure that the required engineering rules are applied to all the instances running on the testbed.*"

Therefore, the process of delivery and acceptance of an enabler consists of several procedures whose goal is to ensure the good transfer of information between the Enabler Owner and the Testbed Operator.

4.2.1 Delivery process

The enabler delivery process is composed of three steps as depicted in Figure 6. This process is led by the Enabler Owner (EO) who is supported by the Testbed Operator.

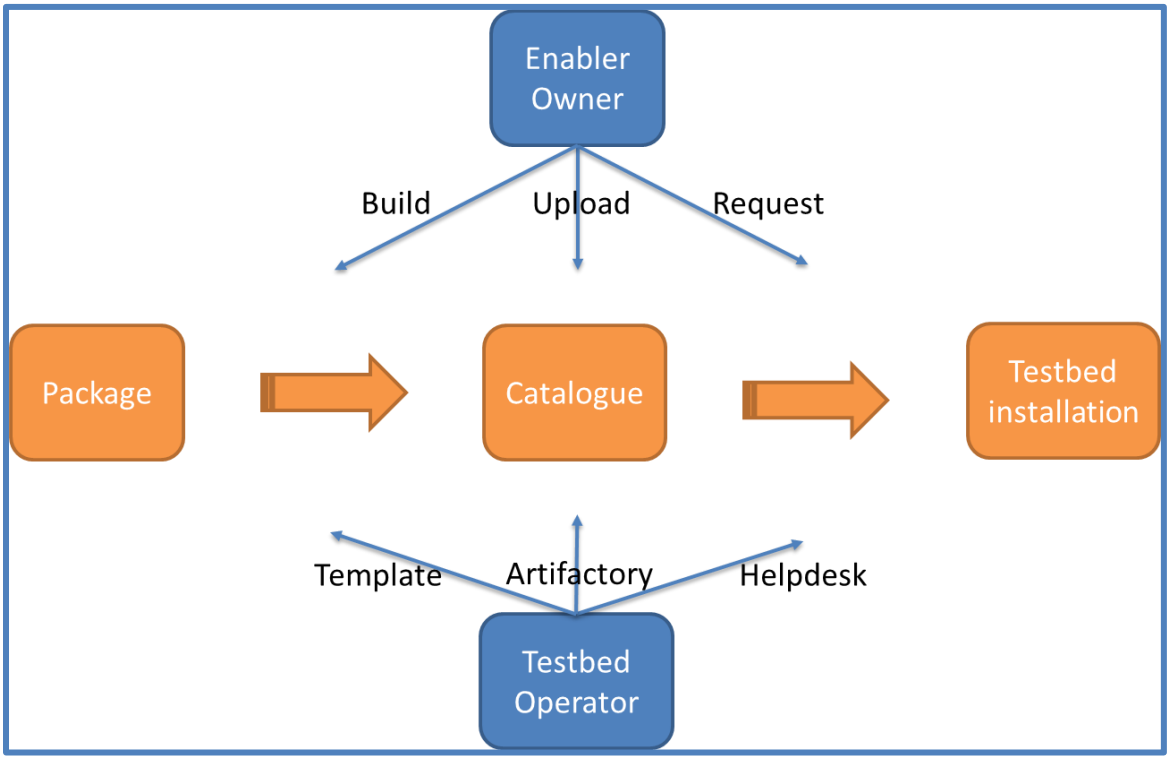


Figure 6: Delivery workflow

- The *package* build: activity where the software, package and documentation is made ready.
- The *catalogue* upload: activity where the enabler is being uploaded on the testbed.
- The *testbed installation* request: permits to trigger the integration process by means of a deployment request through the helpdesk.

As concerns the Package build, the Enabler Owner builds a package containing:

- The dependencies.
- The enabler's object code.
- The configuration file(s).
- The Ansible [10] configuration role (optional).

The Testbed Operator provides templates to simplify this task (the packaging and the Ansible [10] role definition). The list of Ansible roles used for the enabler testbed integration is defined in the annex B of this document.

After the package has been built by the Enabler Owner, the Enabler Owner uploads the package to the 5G-ENSURE testbed catalogue. The catalogue is based on Artifactory [11] and has been provided by the testbed Operator. A dedicated repository is used for 5G-ENSURE enablers. The Enabler Owner provides the dependencies if they are not available as standard distribution packages. This procedure is detailed in section 4.3.

Once the package(s) is(are) made available on the repository, the Enabler Owner requests the enabler deployment through the helpdesk. A dedicated ticket template is available for this specific request. This communication channel is important for managing these requests and track resource allocation.

Figure 7 illustrates the helpdesk deployment request template:

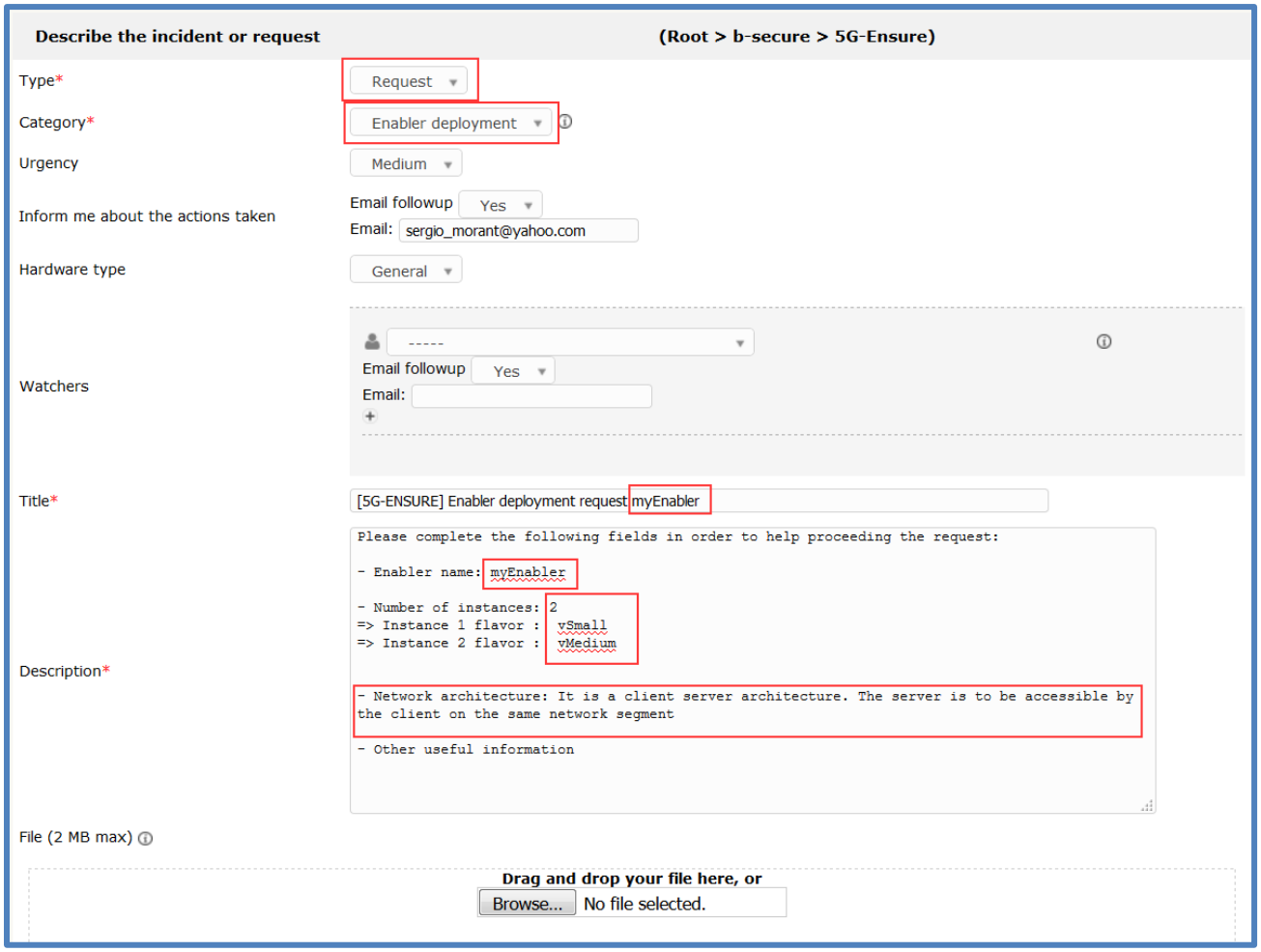


Figure 7: Helpdesk deployment request template

In order to trigger this template, the Enabler Owner needs to create a new **Request** ticket on the helpdesk and choose the **Enabler Deployment** category.

Then, the template will pre-set the required fields with the default information. The Enabler Owner should complete the ticket, before submitting it, with the following information:

Title: add the enabler name as defined on the enabler's open specifications (D3.2 [2], D3.6 [3]).

Description: Provide as much information as possible to help preparing the deployment, namely:

- The number of instances to deploy and their flavours.
- If the enabler is composed of several packages, specify in which instance they should be deployed.

- The requested network architecture allowing the interconnection of all requested instances, and with any other required equipment. Architecture can be delivered as an attached document in the deployment request.
- Any other information that could help the Testbed Operator improves the understanding of the request.

Hereunder, Figure 8 resumes the request created for the Generic Collector Interface deployment as example:

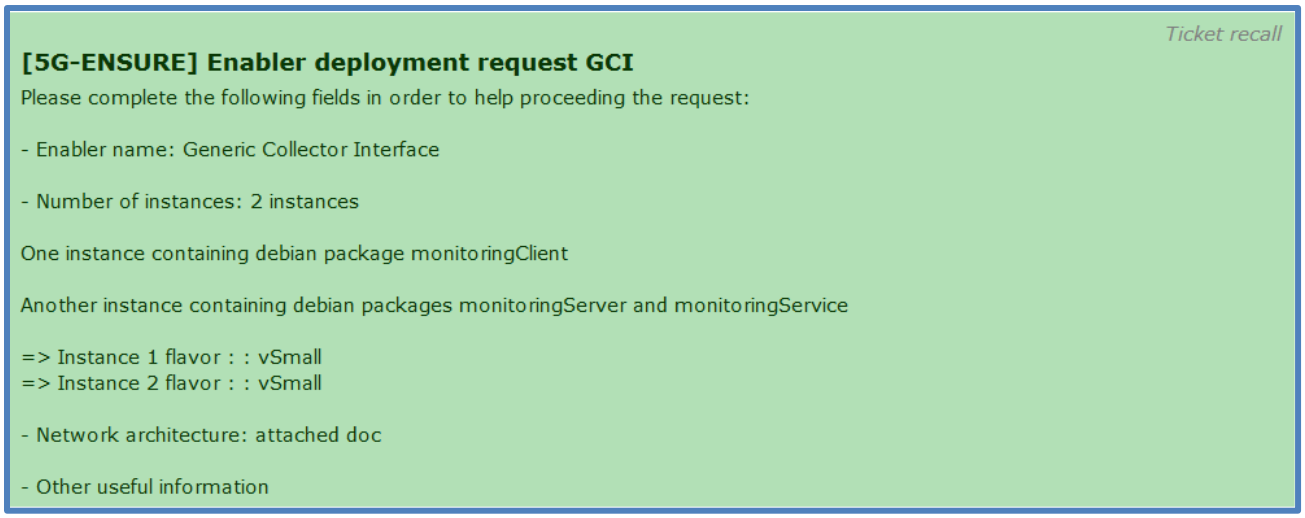


Figure 8: GCI helpdesk deployment request

4.2.2 Integration workflow

Figure 9 depicts the steps that need to be performed to complete the integration on the testbed. In this case, the process is driven by the Testbed Operator with the support of the Enabler Owner.

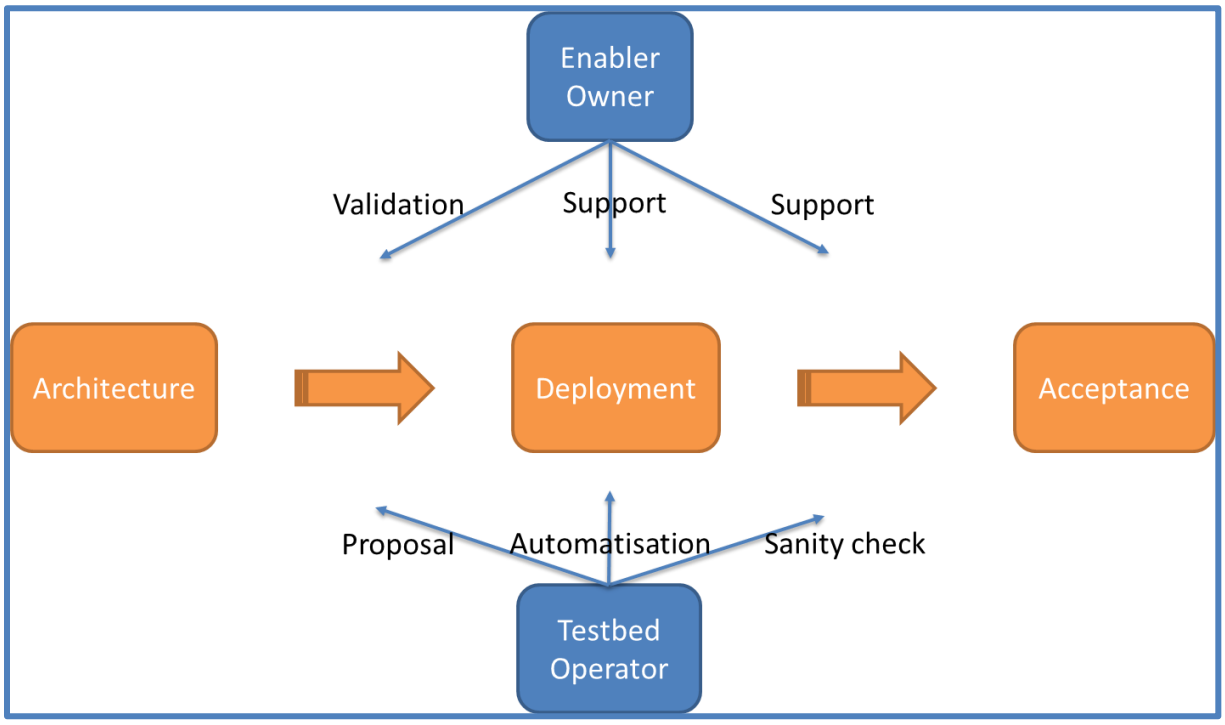


Figure 9: Integration workflow

The architecture proposal allows the specification of a target deployment architecture for the enabler and its associated components. The proposal will be based on the following inputs:

- The enabler User Manual present in D3.4 [12].
- The content of the deployment request generated by the Enabler Owner through the helpdesk.

The Testbed Operator will provide, by answering the helpdesk request, a deployment architecture proposal containing the information required by the Enabler Owner to validate the correctness of the deployment.

The following example (Figure 10 and Figure 11) contains the proposal for the hosting of the Generic Collector Interface.

Please find attached a reviewed version of the architecture in agreement with what was discussed on Friday.
The main change is that the enablers will communicate through the management (OAM) interface as it should on an operational network.
All configuration regarding the user network is suppressed. Here is the resume:

Services :
gci-client : **10.102.8.52:4444**
gci-server: **10.102.8.53:8888**
gci-service: **10.102.8.53:5555**

Client instance
=> Hostname: vbsc-5gesrv002.b-secure.local
=> Management IP address: **10.102.8.52/24**
=> Routing:
==> Default gw **10.102.8.1**

Server instance
=> Hostname: vbsc-5gesrv003.b-secure.local
=> Management IP address: **10.102.8.53/24**
=> Routing:
==>Default gw **10.102.8.1**

Added document: Document Ticket 25 - 5G-Ensure_testbed_architecture-GCI.png

Figure 10: GCI Network configuration proposal

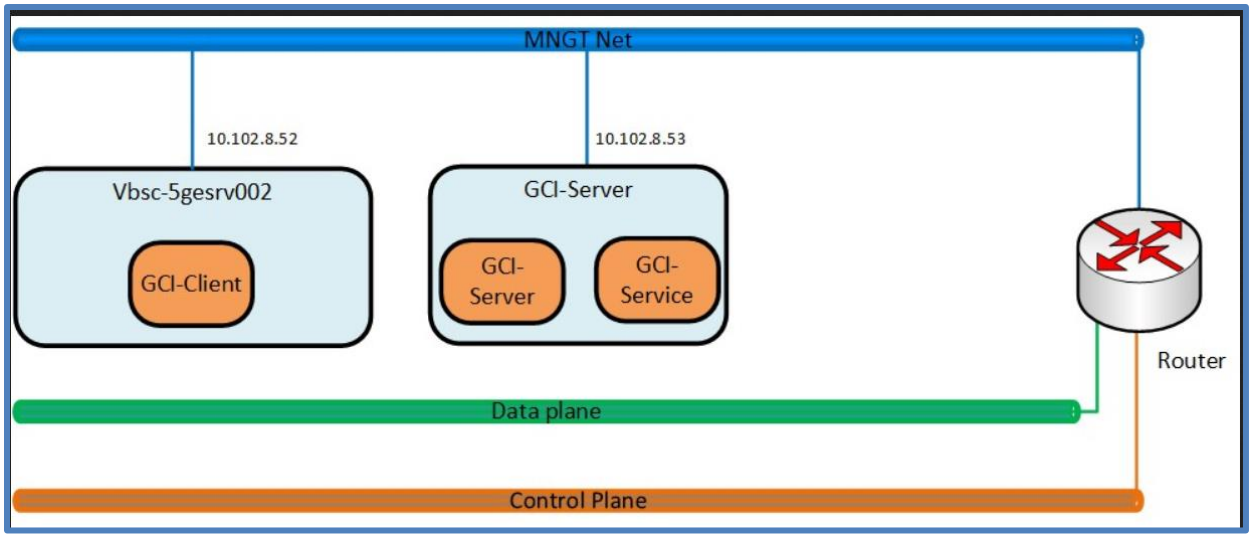


Figure 11: GCI network architecture proposal

Upon validation by the Enabler Owner, which is to be done through the on-going helpdesk request, the deployment can be triggered.

At this stage the Testbed Operator will map all the collected information to the Orchestration and configuration management tools. Once this step is done the deployment process will be held automatically.

At the end of the process, the systems will be deployed with the identified enabler components and the requested configuration. If for any reason there are issues to deploy the target architecture, the Enabler Owner will support the Testbed Operator to identify a solution. The main communication channel to support this action is the helpdesk.

Once the enabler and its associated components are deployed, the acceptance procedure can take place. The goal at this stage is just to run the enabler's unitary tests described in D3.4 [12], which have been integrated as part of the test plan (see section 5.2.1). Running these tests in the testbed, functions as enabler's sanity checks. If the enabler passes the tests, it can be considered as integrated in the testbed. The testbed acceptance of the enabler is announced by means of an official mail to the Enabler Owner and the project Technical Manager.

4.3 Delivering an enabler on the catalogue

This procedure was described in a high level in D4.1 [5]. This section aims at describing the procedure in more details now that the testbed and the catalogue tool are fully operational.

A catalogue tool (Artifactory [11]) is provided within the testbed. It centralizes and manages the delivery and deployment of the enablers within the testbed. Enabler packaging is an operational requirement for the enablers to be deployed on the testbed.

Hereunder the complete procedure to deliver an enabler on the catalogue is specified:

- Connect to the catalogue repository: https://artifact.b-com.com
- Login using the personal testbed credentials. A web page looking like the following should appear:

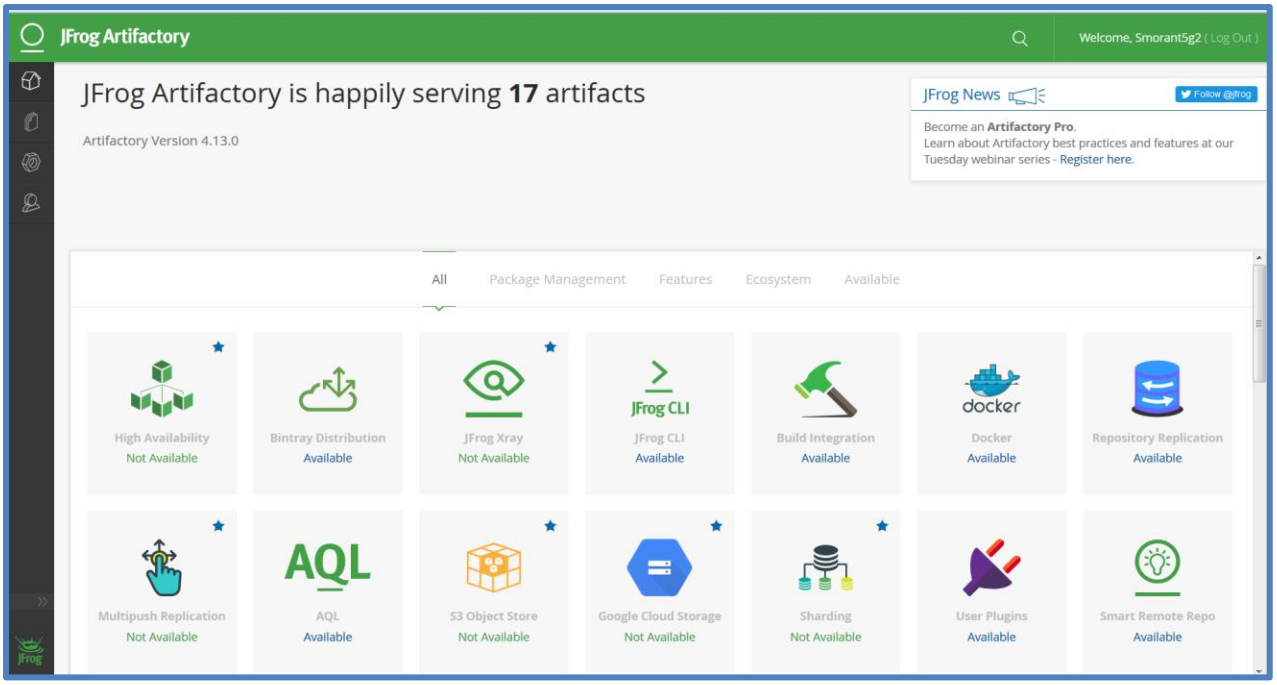


Figure 12: Catalogue home page

- Go to the Artifacts menu. A screen looking like Figure 13 should appear.

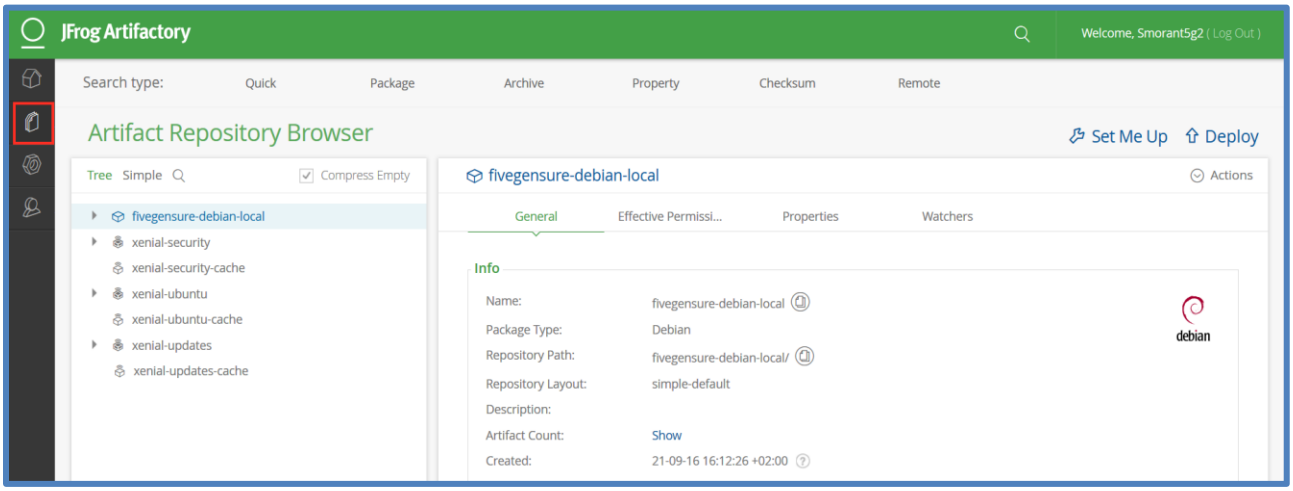


Figure 13: Catalogue repository page

- The following repositories are available at the time of the writing of this deliverable (see Figure 14):
  - **Fivegensure-debian-local**: Repository dedicated for the 5G-ENSURE project enablers for Debian / Ubuntu distributions.
  - **Xenial-xxxxx**: Repositories used to cache Ubuntu Xenial distribution packages. This allows to install the system packages on the testbed from a local repository.
- Choose the target path on the left hand side of the webpage, taking into account the considerations regarding the operating system (Ubuntu Xenial), the architecture (amd64) and the nature of the enablers regarding their Intellectual property (restricted). This would provide the following target path "*fivegensure-debian-local/dists/xenial/restricted/binary-amd64/*".

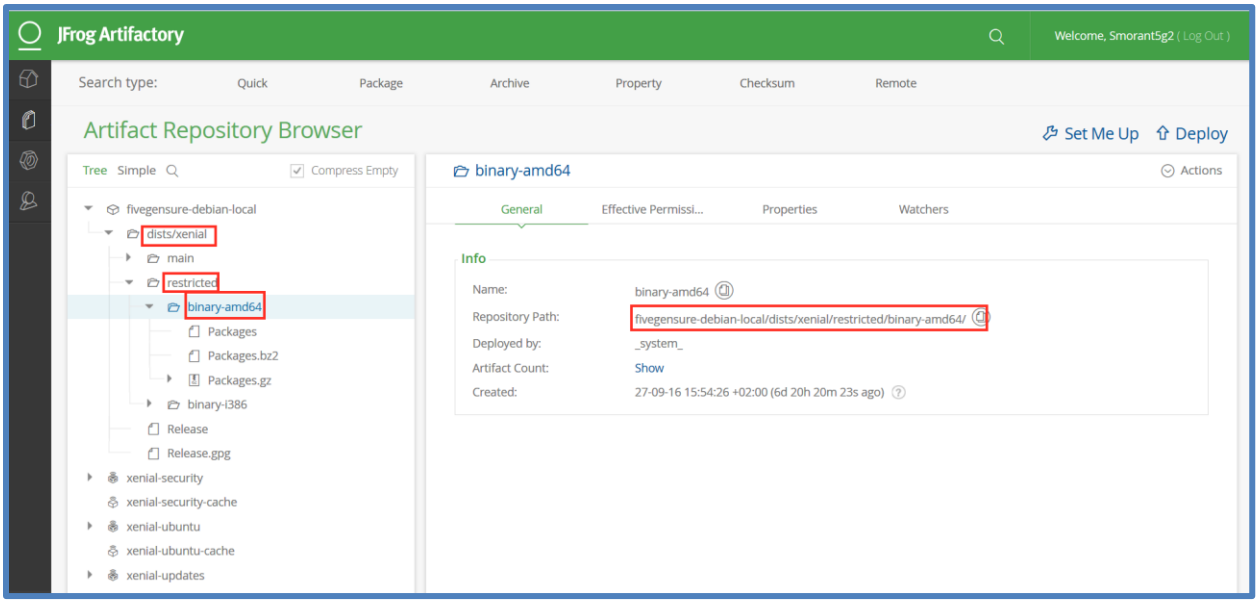


Figure 14: Catalogue 5G-ENSURE Debian / Ubuntu repository

- In order to upload a new package on the catalogue, click on the Deploy button. The following menu will appear (see Figure 15):

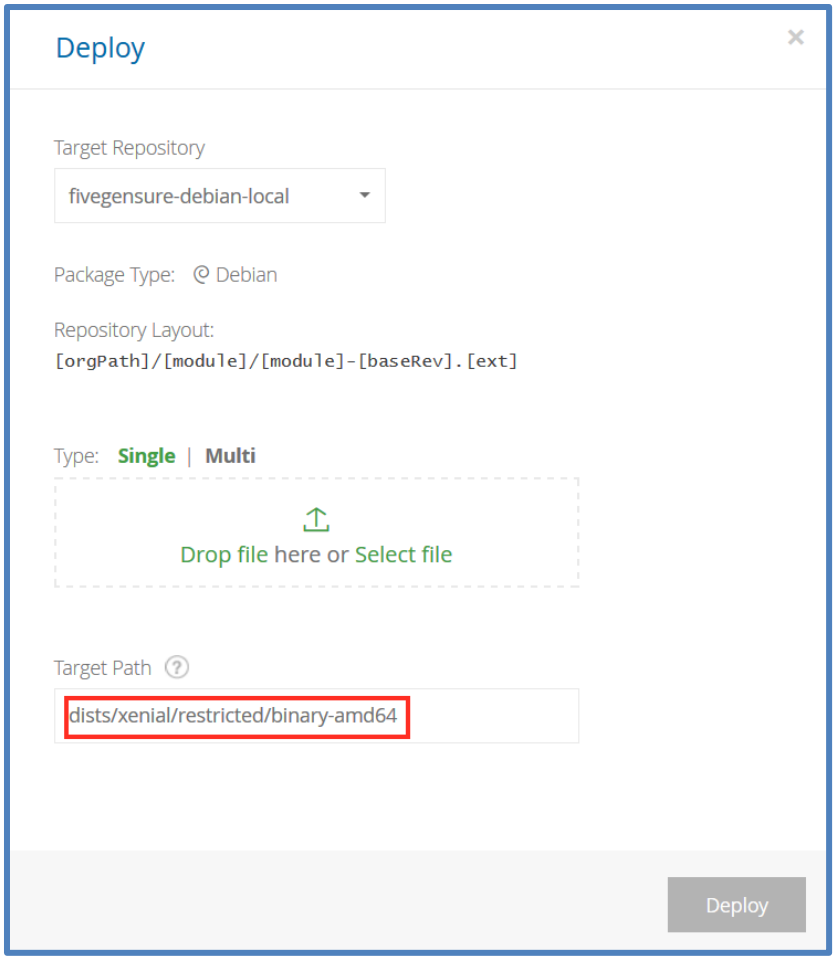


Figure 15: Catalogue deploy menu (1/2)

- Select the file containing the enabler and check that the target path is set as expected. Then click on the "Deploy" button (see Figure 16).

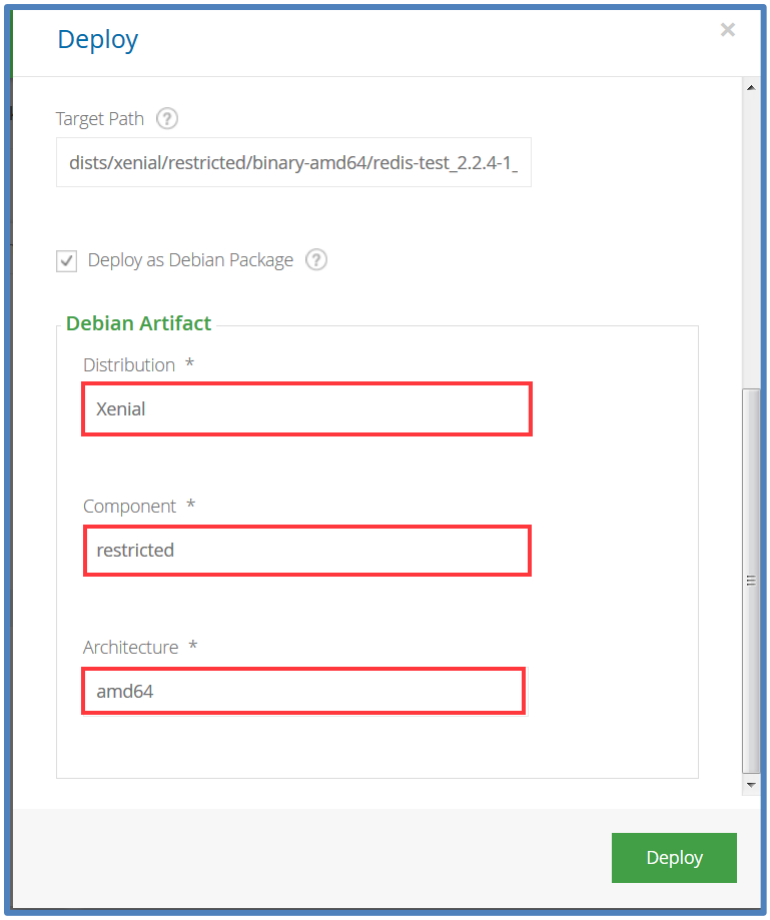


Figure 16: Catalogue deploy menu (2/2)

Alternatively, it is also possible to perform the action using the Artifactory Rest API. In order to get the right format, use the Set Me Up button on the Artifacts menu. This will provide the curl command template. The output should look like as in Figure 17.

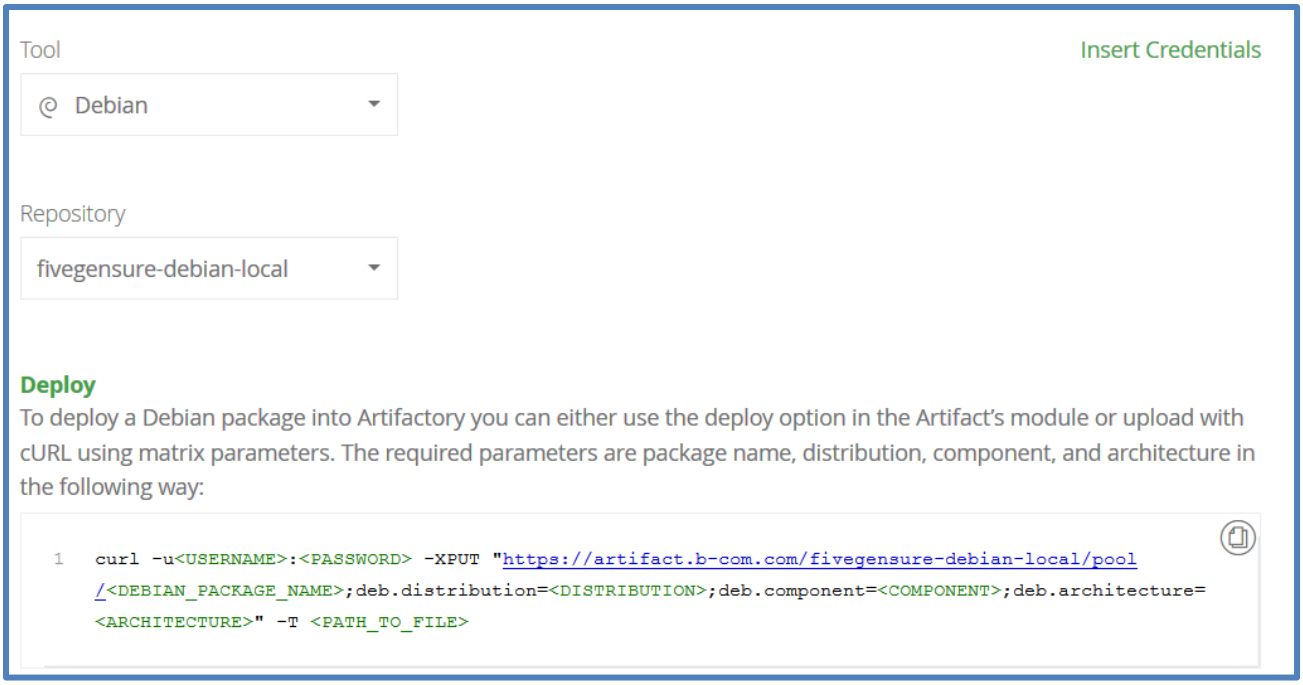


Figure 17: Catalogue Curl template for package upload

At this point, it should be possible to install / update the enabler from any testbed host system by using the repository management tool, or deploy a new Docker container.

For the instances deployed on the testbed, the configuration management tool will configure their repository to point to the catalogue.

4.4 Running an enabler security evaluation

Note: to enter in a **security evaluation stage**, an enabler should have finalized its **integration stage** as described in the section 4.1.

4.4.1 Overview

The evaluation stage will be performed for a specific pair of elements defined as (enabler feature, threat), as it was stated in (D3.5 [8], D3.6 [3] and D2.3 [4]).

The Enabler Owner has to describe how its enabler may mitigate some of the identified threats (see Table and Table 8). This description will be based on the enabler technical specification, threat and uses case, and the testbed's available nodes and resources (see D4.1 [5] testbed architecture description).

The evaluation scenarios proposed for testbed execution must comply with Testbed Term of Use (see appendix A) policies

Hereafter is shown the evaluation Scenario validation:

- **WP2 "Security requirements & Architecture" also in charge of the Use Cases is responsible** for validating if the proposed Scenario, delivered by E.O., is sufficient to demonstrate that the enabler addresses and mitigates the

identified threat. It is not the WP2 responsibility to look neither at the enabler implementation details, nor penetration test, nor configuration / software security evaluation of the proposed enabler feature.

- **WP4 "Testbed" is responsible** for validating if the proposed Scenario (after WP2 validation) is technically compatible with the testbed architecture (see D4.1 [5]).

  WP4 will thus assess technical feasibility of the scenario on the Testbed and report it to the Enabler Owner. If doable WP4 would further interact with EO to agree on final version of the scenario. In case the scenario proposed by EO can't be supported by the Testbed, WP4 may then request the E.O. to proceed with another type of metrics among the ones proposed (see below).

The joint Evaluation process delivers a metric on the nature of proposed scenario.

Scenario Evaluation Metric:

0: no evidence of coverage of the threat is delivered

1: theoretical evidence (scientific article) of coverage of the threat is delivered

2: implementation delivered (integration phase on the testbed achieved and evaluation test described inside TestLink have been validated by WP2 without performing it, see 4.4 Running an enabler security evaluation)

3: Evaluation Tests performed on the testbed, based on simulated environment, achieved and positive.

4: Evaluation Tests performed on the testbed have been done over the real testbed flows as described in the evaluation Scenario validated by WP2 and corresponding test description (TestLink).

5: Scientific paper (formal proof) and verification that the tested code and Scenario conform to the scientific paper. This could only happen once level 4 is achieved for the specific (enabler feature, threat).

Important note: in some case the (enabler.feature, claims) could not be evaluated over the testbed, but the enabler owner could deliver to WP2 scientific evidences of this coverage. This can happen for instance for the enabler's features not having software implementation, or requiring non available components (i.e NFV MANO). Based on these evidences, WP2 will validate the evaluation Scenario, without any implementation over the testbed (see section 4.5).

The project Evaluation Scenario process is described hereafter

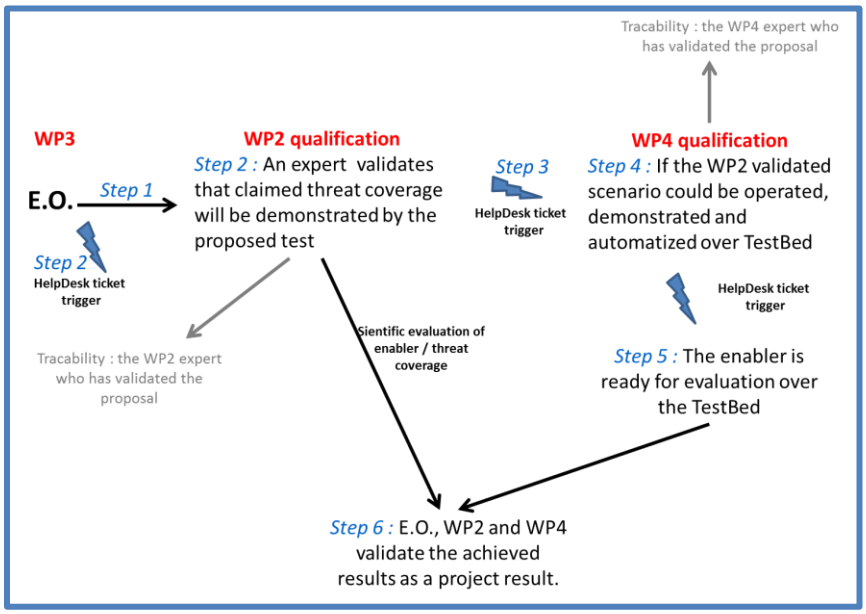


Figure 18: project Evaluation Scenario process

Step 1: To deliver evidences and facts of threats coverage, the E.O. delivers a description of the Scenario allowing to demonstrate the coverage of the identified threats.

Note: in some cases an evaluation scenario may also involve more than one feature, or potentially more than one threat.
For the pair (enabler feature, threat), the Enabler Owner (E.O.) description aims to cover:

- The relations and differences between use cases, attacks, threats and the proposed evaluation Scenario.
- Clear explanation on how the E.O. interprets the threat(s) to be covered by the enabler feature (interpretation of the threat objectives)
- Motivation of the chosen techniques, algorithms, heuristics to cover the threat. There is the need of adding reference of publication supporting evidence
- Description of its Scenario in Testlink, which:
    - Relates the steps of the Scenario to the threat interpretation.
    - Explains the choice (and motivation) of attacks (real or simulated) and the strategy of evidence for the coverage
    - Describes the strategy in a step by step manner
    - Identifies the required preconditions of each step
    - Describes the attack scenario proposed in a step by step manner
    - Describes the evidences of threat coverage (related to threat interpretation) in a step by step manner
    - Describes the enabler feature's property used and why it is an evidence of threat coverage (and potential links with other threats)

Step 2: The Scenario proposal will be then reviewed by WP2. The scenario review process would be triggered by creating a ticket (review requirement) on the Help desk. Ticket that would also encompass a short description of the scenario itself for WP2 receiving person to possibly dispatch it to expert of the domain/field covered. WP2 notifies the Enabler Owner that their evaluation scenario with regard to the threat coverage claims has been processed (reviewed).

Step 3: WP2 triggers WP4 to evaluate the potential demonstration of Scenario proposed by the E.O. as WP2 as finalized the threat coverage evaluation.

Step 4: WP4 assess technical feasibility of the proposed tests and Scenario (in case of issue, we go back to step 1 or finalize the evaluation procedure based on theoretical evidences)

- Establish if the proposed pre-conditions, steps and attacks are feasible and reproducible on the testbed.
- Identify the building blocks required to describe / operate the test.
- Validate the technical strategy to be implemented over the testbed.

The 2 next steps are outside of the evaluation process, they are related to the technical evaluation and associated measures performed over the TestBed.

Step 5: WP4 **runs tests** based on the description in TestLink (under the E.O. responsibility) and performs the evaluation of test result.

Step 6: E.O., WP2 and WP4 validate the achieved results as a project result.

Note: the evaluation performed on the testbed for a specific pair (enabler feature, threat) will be based on the proposed Scenario (defined by E.O and validated by WP2/WP4)), but nothing prevents extra Scenarios from being defined and run after evaluation phase of one enabler feature, regarding the acquired information inside the whole project.

4.4.2 Scenario validation process

In order to achieve the implementation of the workflow allowing the evaluation of the Scenarios proposed by the Enabler Owners, it is required to use several tools. The choice made by the project partners is to re-use the already deployed tools, and customize them in order to support the new workflow. The chosen tools are:

- **Helpdesk**: It is the central element allowing tracking the workflow status for a given feature and the associated evaluation Scenarios.
- **Test plan tool**: Allows to record the scenario definition, its status (draft, under review, final…), its evaluation, and the execution results (when executed on the testbed).

The main stages of the workflow are shown in the Figure 19. They are identified together with the main actions driven on each of the tools.

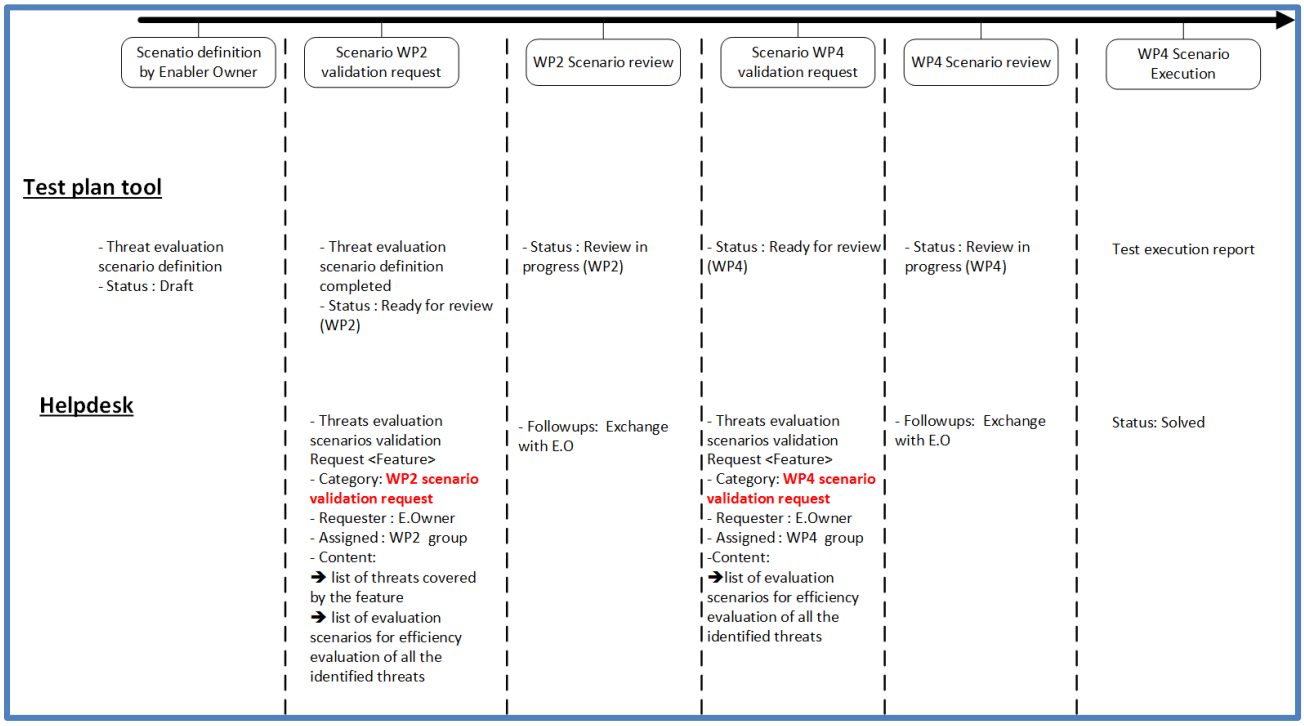


Figure 19: Enabler evaluation Scenario timeline

The following tasks have been defined to implement the workflow, namely, scenario definition, WP2 scenario validation request, WP2 scenario review, WP4 scenario validation request, WP4 Scenario review, and WP4 Scenario Execution. These tasks are further detailed hereafter.

Scenario definition

- **Task owner**: Enabler Owner
- **Task description**: This is the initial stage of the workflow. This stage aims at providing the evaluation Scenario by the means of one or several Test Cases defined in the Test plan tool.
- **Task actions**:

- Test Plan tool: The E.O. is responsible for adding the definition of the evaluation Scenario on the Test Plan tool and to provide it. The Scenario must be introduced as a threat test case as explained in section 5.2.2. The **Scenario status** at this stage is set to "**Draft**"
- Helpdesk tool : No action

WP2 scenario validation request

- **Task owner**: Enabler Owner
- **Task description**: The Enabler Owner considers that the scenario proposed to validate the threat coverage for a given feature is ready to be evaluated by the WP2. The Enabler Owner will trigger the review by creating a helpdesk ticket containing the following information:
  - The relations and differences between use cases, attacks, threats and the proposed evaluation Scenario.
  - Clear explanation on how the E.O. interprets the threat(s) to be covered by the enabler feature (threat objectives interpretation)
  - Motivation of the chosen techniques, algorithms, heuristics to cover the threat. There is the need of adding reference of publication supporting evidence
- **Task actions**:
  - Test Plan tool: Once the Scenario has been completely defined, the **status** of the associated test case has to be set to "**ready for review**"
  - Helpdesk tool: A new ticket request must be created for the validation of the scenarios associated to feature (one ticket per feature). The ticket will be with the following information
    - **Category**: WP2 scenario validation request (triggers a template)
    - **Type**: Request
    - **Title**: [5G-ENSURE] WP2 Scenario evaluation request *<Feature>*
    - **Description: brief introduction to the scenario in object for WP2 receiving person to figure and allocate to right expert person**

The helpdesk ticket will be automatically affected to the WP2 reviewers based on its category.

WP2 scenario review

- **Task owner**: WP2 Reviewers
- **Task description**: This task encompasses the review and the evaluation of the Scenarios proposed by the Enabler Owner for a given feature.

  WP2 reviewers must provide the following verdicts:

  - *Scenario validation:* The WP2 reviewer must state that the proposed scenario covers the claimed threat. It might happen that the evaluation Scenario must be reworked in order to better specify how it covers the identified threat.
  - *Scenario scoring*: Based on the Scenario definition and the claims provided in the ticket request, the WP2 reviewers should provide a score for the scenario based on the evaluation metrics available in section 4.5. This score is to be recorded on the test plan tool as described on the task actions hereunder.
- **Task actions**:

- *Test Plan tool*: The test case **status** would be set to "**Review in progress**". If the WP2 review for the given Scenario is not satisfactory, then the status would be set to "**Rework**". The evaluation of the scenario is to be defined in the **Scenario evaluation result.**
- *Helpdesk tool*: On the ticket it is possible to add comments related to the different scenarios to help the Enabler Owner improve the Scenario definition. Once the review process is considered as finished for the given feature, the helpdesk **ticket status** can be set to "**solved**"

WP4 scenario validation request

- **Task owner**: Enabler Owner
- **Task description**: Once the Enabler Owner has met the expectations of the WP2 reviewer regarding the scenario definition, he/she can trigger the WP4 scenario validation request. In order to do so, the E.O will create a new ticket on the helpdesk that will be, based on its category, and automatically affected to the WP4 reviewers. Besides the list of the evaluation Scenarios that are requested for implementation on the testbed, the E.O. should provide, as much as possible, the information allowing the implementation of the required architecture on the testbed. It is important to recall that, in order to ensure that scenario can be played on the testbed, a step by step testing procedure should be included on the test case definition.
- **Task actions**:
  - *Test Plan tool*: Once the Scenario has been completely defined, the **status** of the associated test case has to be set to "**ready for review**"
  - *Helpdesk tool*: A new ticket request must be created for the validation of the scenarios associated to feature (one ticket per feature). The ticket will be with the following information
    - **Category**: WP4 scenario validation request (triggers a template)
    - **Type**: Request
    - **Title**: [5G-ENSURE] WP4 Scenario evaluation request <*Feature*>
    - **Description**: <*Complete the template*>

WP4 scenario review

- **Task owner**: WP4 Reviewers
- **Task description**: This task encompasses the review and the evaluation of the scenarios proposed by the Enabler Owner for a given feature requiring a validation on the testbed. The main goals for the review are :
  - Validate whether the proposed scenario can be executed in the testbed.
  - In the case that the Evaluation scenario can be implement in the testbed, make an architecture proposal for the scenario validation.
  - In case the scenario execution could not be performed on the testbed, this may lead to a requalification of the scenario evaluation result.
- **Task actions**:
  - *Test Plan tool*: The test case **status** would be set to "**Review in progress**". If the WP4 review for the given Scenario is not satisfactory, then the status would be set to "**Rework**". The **Scenario evaluation result** may be reviewed as described previously.
  - *Helpdesk tool*: On the ticket it is possible to add comments related to the different scenarios to help the Enabler Owner improve the Scenario definition. Once the review process is considered as finished for the given feature, the helpdesk **ticket status** can be set to "**solved**"

Whenever the scenario is completely reviewed and ready for implementation, its **status** has to be set to "**Final**".

Figure 20 shows the overall validation workflow and the possible transitions from one task to the next one

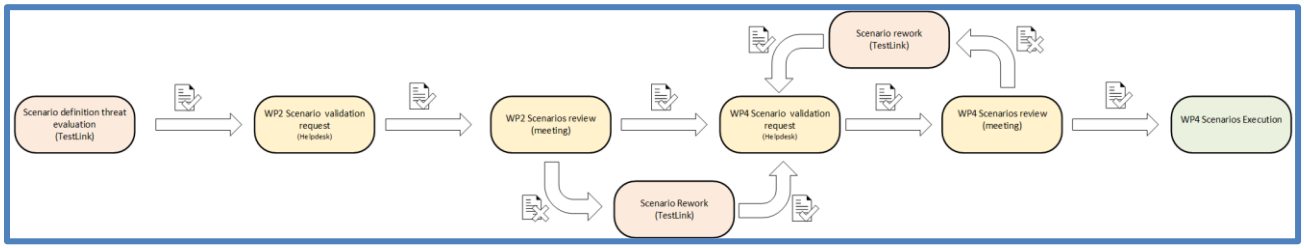


Figure 20: Enabler evaluation Scenario flowchart

Notice that some transitory stages are added with regard to the timeline in Figure 19 and the previous definitions. The aim is to better illustrate the transitions where it is required to rework the evaluation Scenario.

4.5 Project's evaluation metric definitions

We provide hereafter a set of elementary metrics to evaluate the coverage of different threats. The metric integrates 2 dimensions, the first is composed by the WP2 evaluation steps performed (how and on which evidence the security claims of the enablers are demonstrated by the evaluation Scenario proposed) combining with the WP4 evaluation performed (feasibility to perform the described Scenario using the testbed). The second dimension is related to the results and evidences collected on testbed based on the execution of validated (WP2/WP4) scenario implementation.

Note: an Enabler feature could be evaluated even if no implementation was proposed, through a scientific qualification by WP2 of evidences collected:

Global Evaluation Metric (based on **Scenario Evaluation Metric** delivered by WP2/WP4 process)

0: **No evidence of coverag**e of the threat is delivered

1: **Theoretical evidence** (scientific article) of coverage of the threat is delivered. (See hereafter for detailed description of s values).

2: **Implementation delivere**d (integration phase on the testbed achieved and evaluation test described inside TestLink have been validated by WP2 without performing it, see 4.4 Running an enabler security evaluation)

3: **Evaluation Tests** performed on the testbed, based on **simulated environment**, achieved and positive.

4: **Evaluation Tests** performed on the testbed have been done over the **real testbed flows** as described in the evaluation Scenario validated by WP2 and corresponding test description (TestLink).

A proposal to improve the scientific evaluation could be to apply the following metrics could to score the scientific publications:

0: no reference to a scientific paper

1 : scientific Paper independent of 5G-ENSURE work (like state of the art or external specification) no impact on enabler

2 : scientific Paper related on enabler work, establishing the concept and protocol (without implementation) without scientific committee review

3 : scientific Paper related on enabler work, establishing the concept and protocol (without implementation) submit to scientific review and qualification

4 : scientific paper based on the enabler's concept, implementation and assessment / measurement without scientific committee review

5 : scientific paper based on the enabler's concept, implementation and assessment / measurement, submit to scientific review and acceptance

Note: This scoring have not been applied to the evaluation process due to its late arrival in the project timeframe.

Some evaluation examples:

- The following pair (enabler feature, threat), where only unitary tests are performed (integration phase) but there is not any theoretical, nor technical, nor scientific evidence on how it covers the claimed threats will be scored with the value “0”.
- The following pair (enabler feature, threat), where only theoretical, technical or scientific paper based evidence on how it covers the claimed threats will be scored with the value “s”.
- The following pair (enabler feature, threat), where theoretical or scientific evidence is delivered and unitary test(s) are performed (integration phase) will be scored with the value “s1”.
- The following pair (enabler feature, threat), where evaluation scenario have been validated by WP2 and WP4, and for which evaluation Tests have been performed and achieved over the real testbed flows as described in the evaluation Scenario will be scored with the value “3”.

Those metrics are delivered for each pair (enabler feature, threat).

5 Test plan

This chapter covers the way the test plan has been structured and how this structure is matched against the TestLink [9] web tool, which is provided by the testbed to build the test plan, drive the tests, and collect the results.

The complete user manual of TestLink is available at [13] and a screencast is available at [14] also.

5.1 Roles

In D4.1 [5] the following roles related to the test plan are defined:

Test plan Editor

It is a (testbed) user that contributes to the edition of the test plan for the project’s enabler security validation.

Test plan Executor

It is a (testbed) user that participates to the execution of the test plan and the collection of the results.

In this section, these definitions will be extended in two directions:

- Provide the relationship between these roles and those existing on TestLink.
- Identify the partner’s role endorsement

5.1.1 Role matching

TestLink is bundled with 6 different default permission levels built in, as described in [13]. These permission levels are the following:

• **Guest:** *A guest only has permission to view test cases, reports and metrics. He cannot modify anything.*

• **Test Executor:** *A tester has permissions to see and run tests allocated to them.*

• **Test Designer**: *A user can fully work (view and modify) with Test Specification and Requirements.*

• **Test Analyst (or senior tester)**: *A tester can view, create, edit, and delete test cases as well as execute them. Testers lack the permissions to manage test plans, manage Test projects, create milestones, or assign rights. (Initially Senior tester).*

• **Test Leader**: *A leader has all of the same permissions as a tester but also gains the ability to manage test plans, assign rights, create milestones, and manage keywords.*

• **Administrator**: *An administrator has all possible permissions (leader plus the ability to manage test projects and users)*

The roles above are resumed in the Figure 21

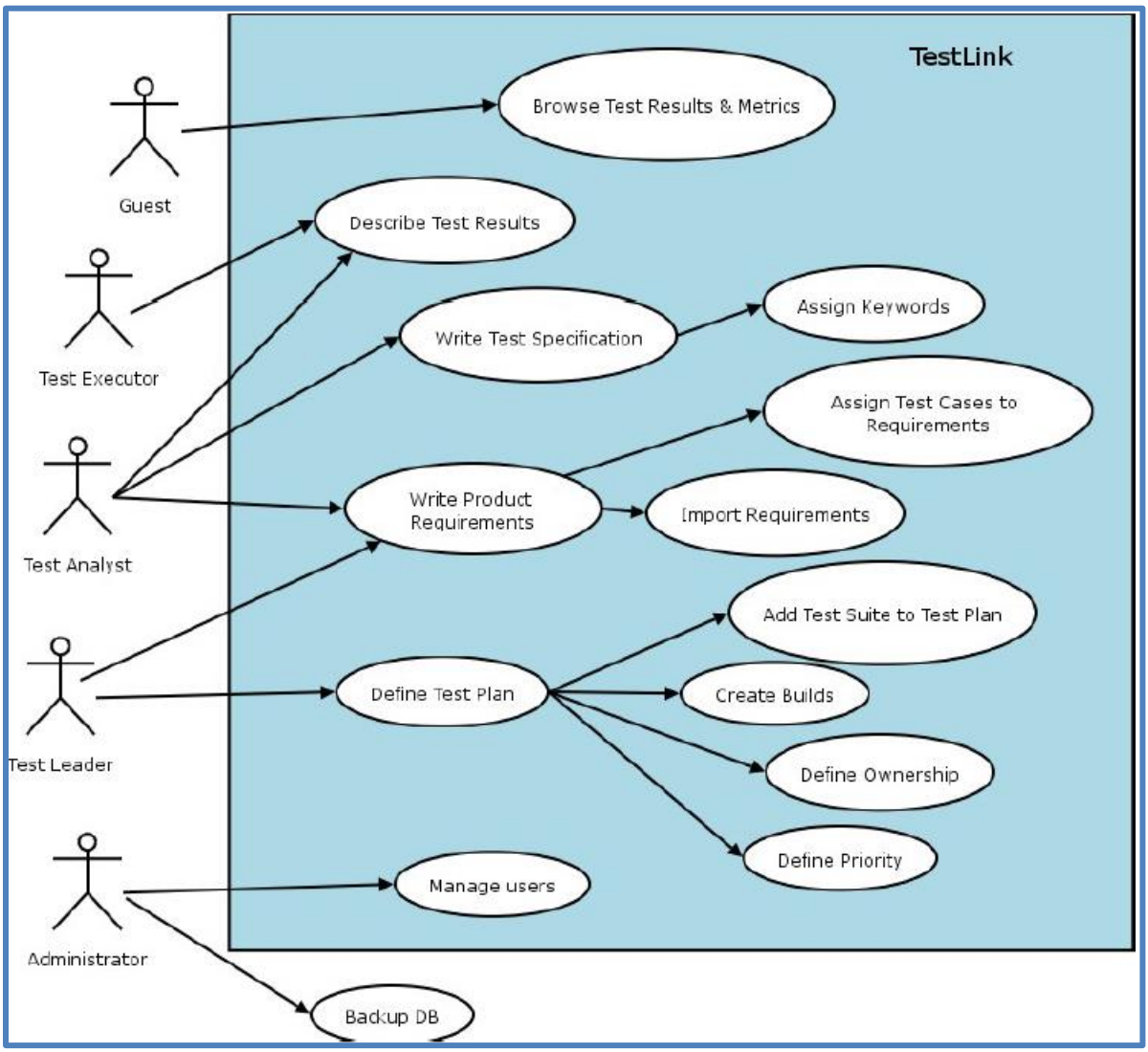


Figure 21: TestLink roles (source [13])

In order to preserve the coherence between the deliverables and for the sake of simplicity, the number of roles defined will be preserved. Here is the proposed matching:

- **Testbed Test plan Editor** ➔ TestLink Test Analyst (Senior tester)
- **Testbed Test plan Executor** ➔ TestLink Test Executor

There is a third role, not directly related to the testing strategy, which is the administrator role. It will be played by the Testbed Operator as for any other service provided within the testbed.

5.1.2 Role endorsement

As described in the next section, the test plan will be divided in two threads: enabler feature sanity check and enabler security evaluation. Depending on the threat, endorsement will differ.

Enabler feature sanity check

The main goal is to validate the integration of the feature in testbed.

- **Testbed Test plan Editor** ➔ Enabler Owner
- **Testbed Test plan Executor** ➔ Testbed Operator

The tests are based on the unitary test cases defined on D3.4 [12]

Enabler security evaluation

- **Testbed Test plan Editor** ➔ Enabler Owner
- **Testbed Test plan Executor** ➔ Partners involved in 5G-ENSURE testbed test plan activities

In this case, the goal is that the enabler owner, in collaboration with WP2 members (see 4.4 Running an enabler security evaluation), establishes the test cases that would allow for evaluation of its enabler against the security threats covered by the enabler. The testbed operator will afterwards check the feasibility of the test case within the testbed, and will support the enabler owner to describe them within the scope of the testbed.

5.2 Structure

This section will cover the test plan structure and its mapping against the test plan web tool. As described previously on the document, the goal of the test plan is to provide the means to evaluate the enabler's security claims against the identified security use cases, and their associated security threats. However, it is important also to check that the enablers have been properly integrated on the testbed, prior to start the security evaluation. All the project partners have agreed in structuring the test plan to cover both, the integration and the evaluation tests using TestLink [9].

In a first stage, the unitary tests will be driven as sanity checks. They will be run at the end of the testbed integration phase. Then, security evaluation Scenarios related tests will take place during the enabler security evaluation.

In order to use a single tool to collect all test results, the enablers' unitary tests will be added to TestLink. This step will enhance their description in order to correspond with the deployment of the enabler within the testbed.

The internal structure used by TestLink is described in details in the user manual [13]. Here, the focus is on the most important concepts that have been applied to create the test plan. Figure 22 provides the relationship between the objects composing a test plan based on requirements specification as described in [13]. This approach is particularly adapted to the 5G-ENSURE project, as there has been a considerable effort to describe the enablers feature requirements in the deliverables from WP3, and the UCs and the Threat requirements in the deliverables from WP2.

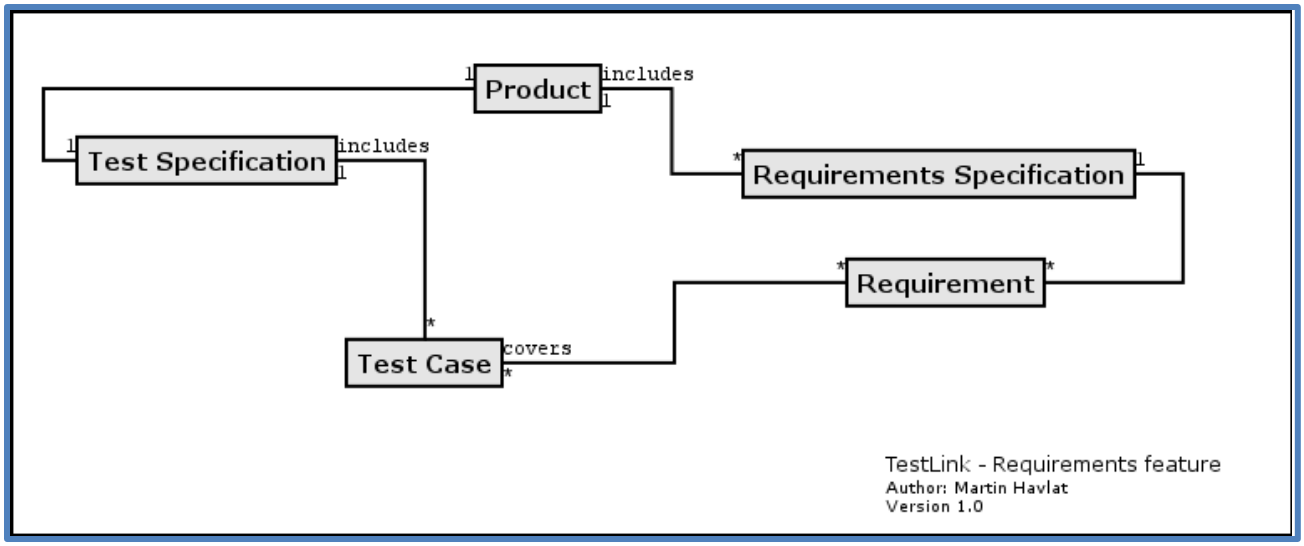


Figure 22: Requirement based test plan (source [13])

In Figure 22, the following elements are depicted:

- **Requirement**: It describes a requirement which can be related to a feature, a use case, a constraint, etc. In the current test plan, the enabler features will be described as feature requirements, and the Security UCs as use case requirements.
- **Requirement specification**: It is a group of related requirements. In the current test plan they are either related to an enabler or a use case cluster.
- **Test case**: It is the testing unit. For each test that needs to be executed on the testbed, there should be a test case providing scope, pre-conditions, steps to perform the test, and expected results.
- **Test specification (or test suite)**: It defines a group of related test cases. In the scope of the 5G-ENSURE testbook, it is either related to enabler features or to a use cases.

5.2.1 Enabler's feature sanity checks

Figure 23 depicts the structure for the testbook with regard to the enabler features (sanity checks)

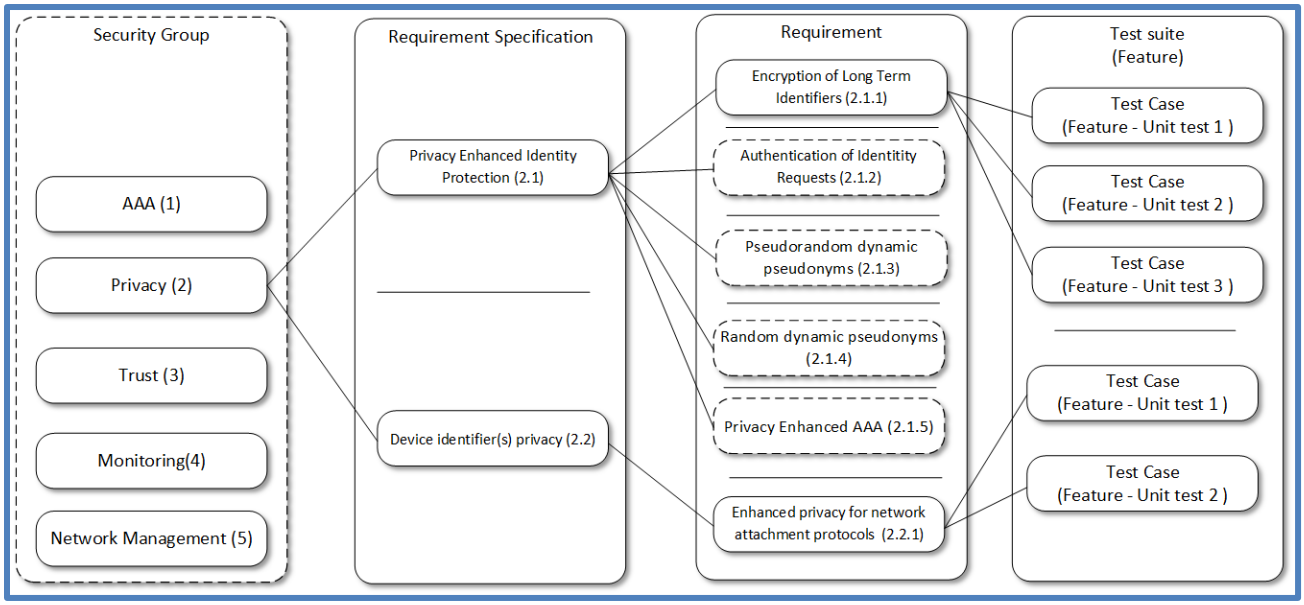


Figure 23: Testbook structure based on enabler features

The goal is to map the structure that has been defined by WP3 for the enablers and their features on the test plan structure. This structure should allow having an eanbler's product features based validation approach. This should be compliant with the feature sanity check to be run at the end of the enabler integration on the testbed.

In the rest of this section it is described the way to actually map the illustration of Figure 23 with the objects inside TestLink [9].

Requirement Specification

Figure 24 provides an example of requirement specification for the "*Privacy Enhanced Identity Protection enabler*"

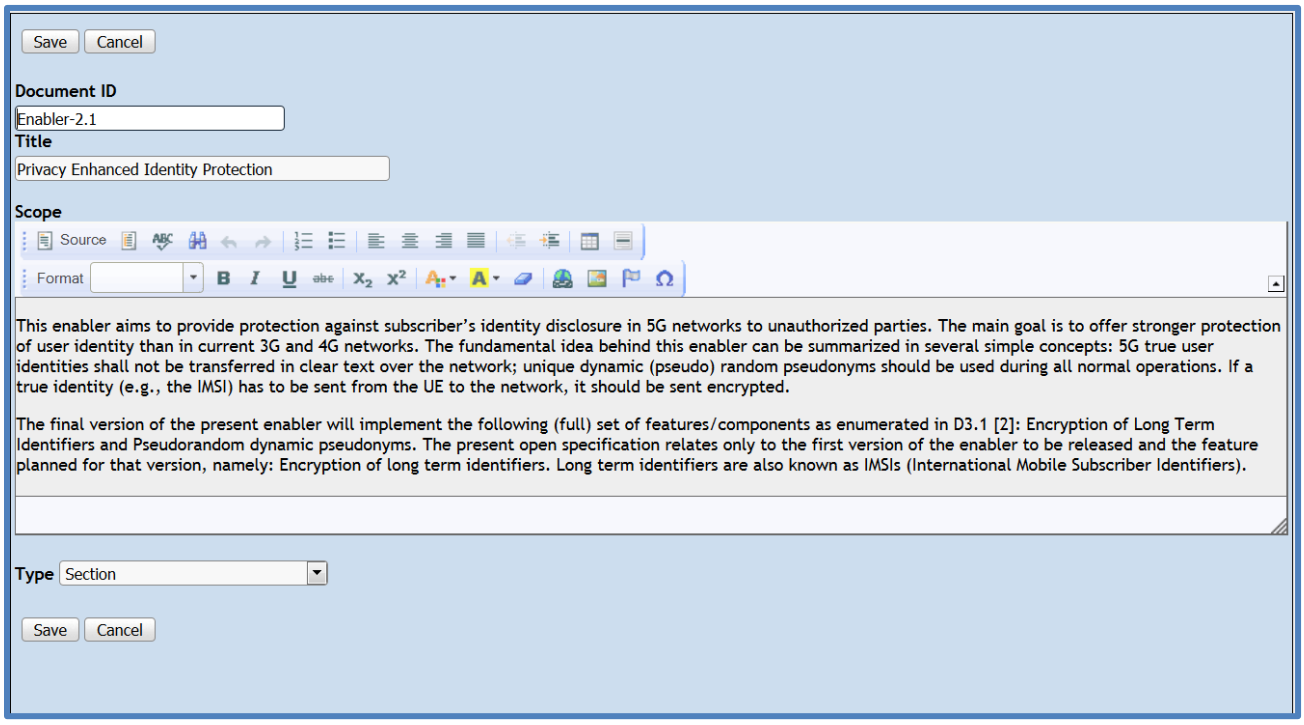


Figure 24: Requirement Specification for "Privacy Enhanced Identity Protection" enabler

The following fields are required to create a requirement specification object:

- **Document id**: Enabler-<enabler_id>.
  - The enabler id must correspond to the identifier assigned in D3.2 [2] or D3.6 [3]
- **Title**: <Enabler name>.
  - As defined in D3.2 [2] or D3.6 [3].
- **Scope**: A description of the enabler's scope. In the current example, the text has been extracted from the enabler's Preface section in D3.2 [2].
- **Type**: Section.
  - It is just used to group the features related to the same enabler.

Requirement

Once the Requirement Specification is defined, it is possible to add new requirements inside. Figure 25 provides an example of feature requirement definition based on "*Encryption of Long Term Identifiers*".

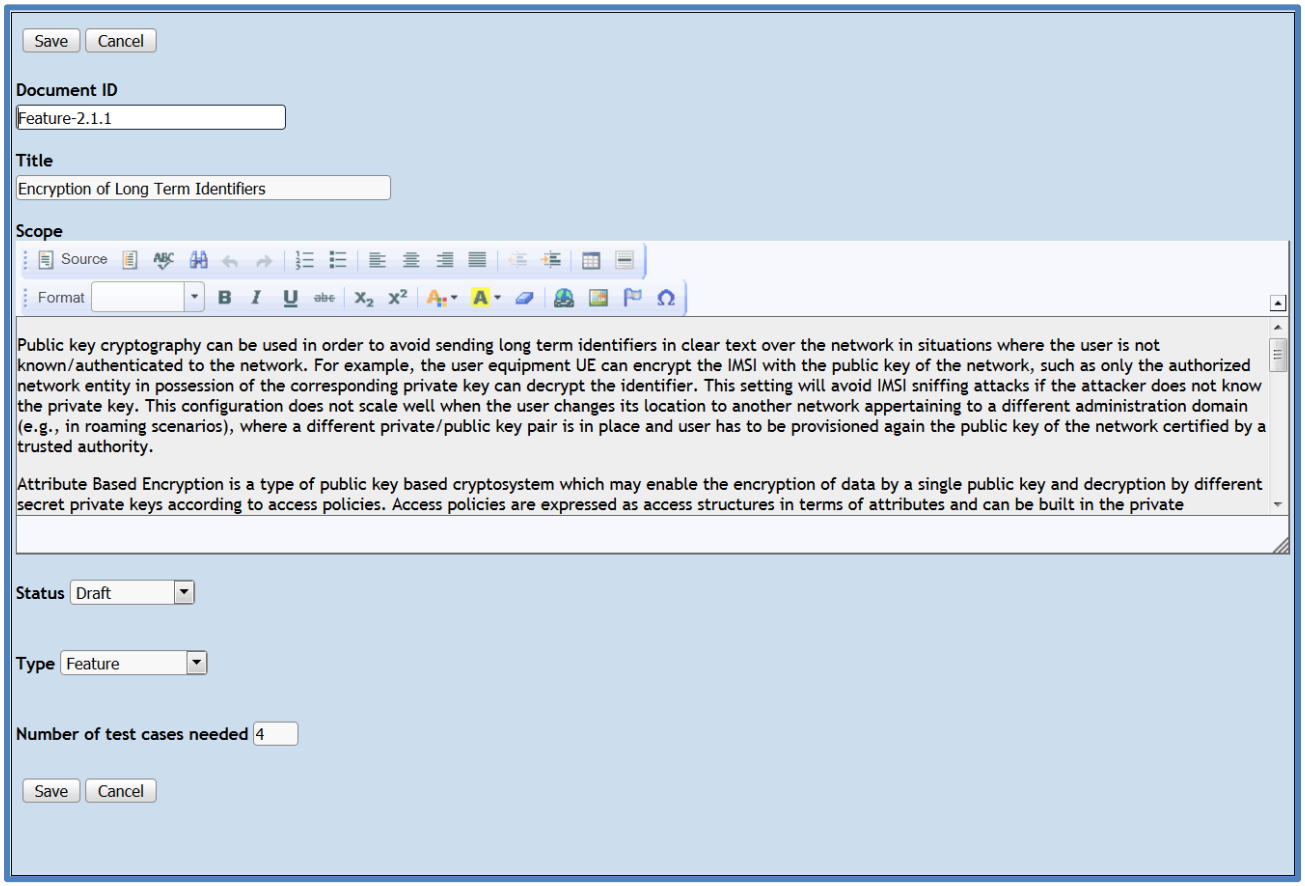


Figure 25: Feature "Encryption of Long Term identifiers" requirement

The following fields are required to create a requirement object:

- **Document id**: Feature-<feature_id>.
  - The feature id must correspond to the identifier assigned in D4.1 [5]
- **Title**: <Feature name>.
  - As defined in D3.2 [2] or D3.6 [3]. .

- **Scope**: A description of the feature's scope. In the current example, the text has been extracted from the Feature basic concepts section in D3.2 [2].
- **Type**: Feature.

Test Suite

Figure 26 depicts an example of test suite definition for the tests related to "*Encryption of Long Term Identifiers*".

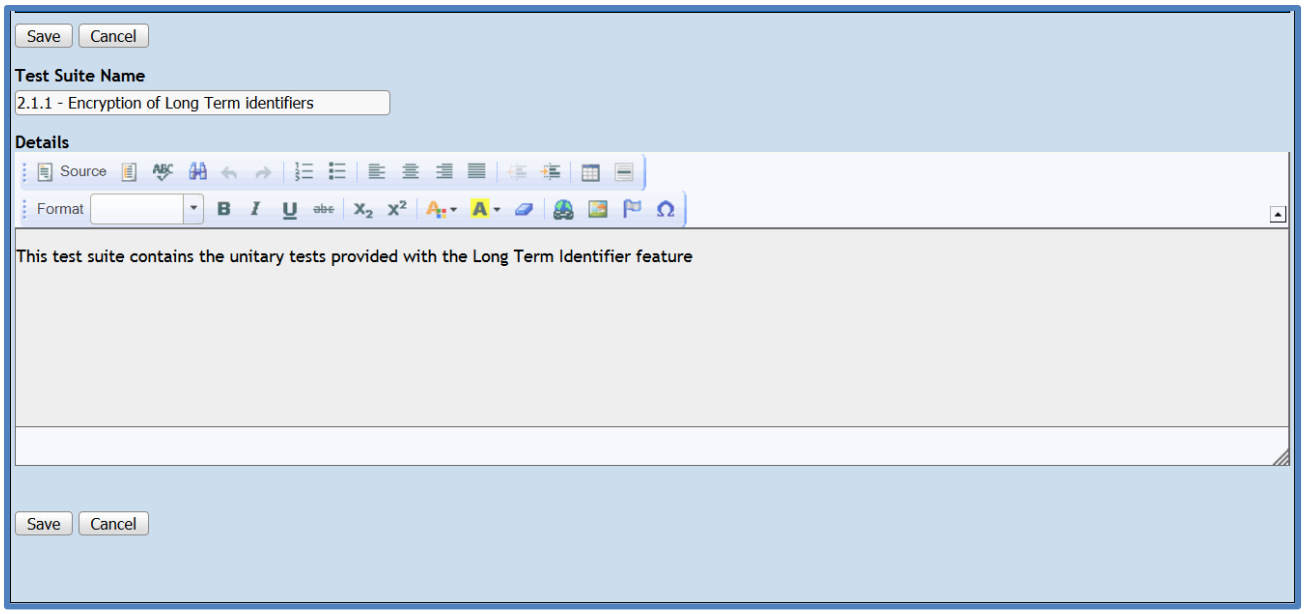


Figure 26: "Encryption of Long term identifiers" feature's Test Suite

The following fields are required to create a Test Suite object:

- **Test Suite name**: <Feature_id>-<Feature name>.
- **Details**: A brief description of the scope of the tests cases that will be grouped on the Test Suite.

Note that the Test Suites within TestLink are structured in a way to preserve the organization of enablers / feature defined by the WP3. In this way some hierarchical sections have been added in order to preserve the classification established by the project. Figure 27 illustrates this organization inside the tool.

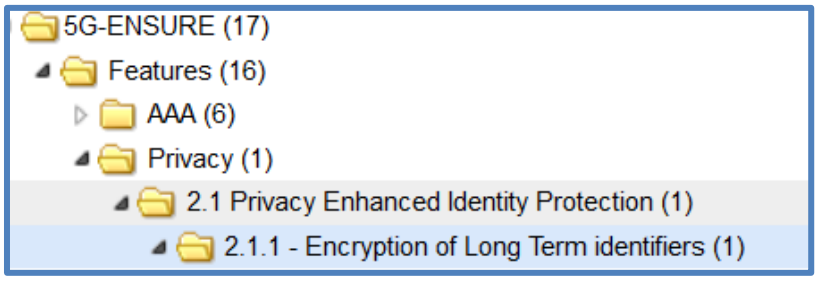


Figure 27: Test cases suites hierarchical organisation

Test case

Once the Test Suite has been defined, it is possible to start adding the test cases inside.

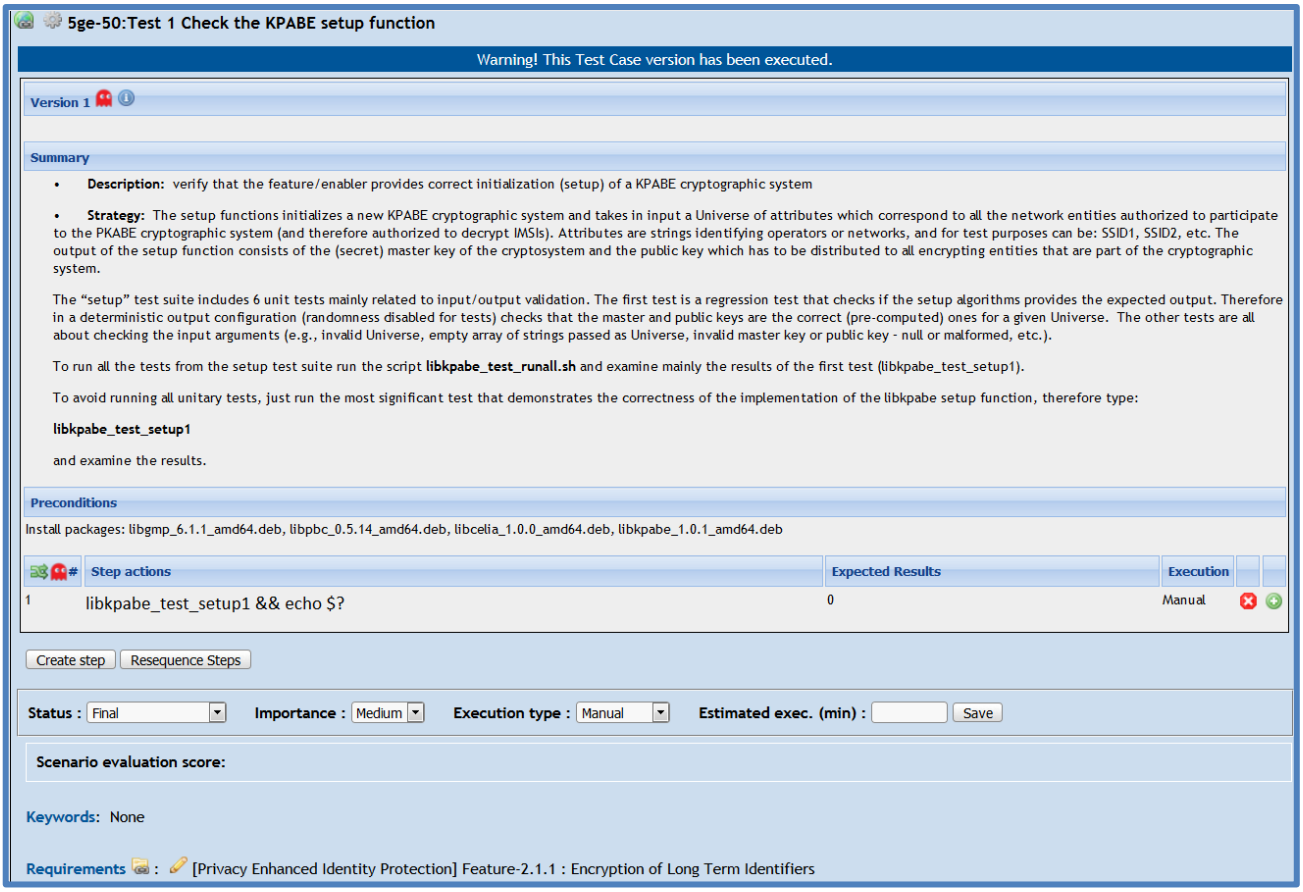


Figure 28 illustrates an example based on the "*Check Function Setup*" unitary test defined in D3.4 [12] for the feature "*Encryption of Long Term Identifiers*".

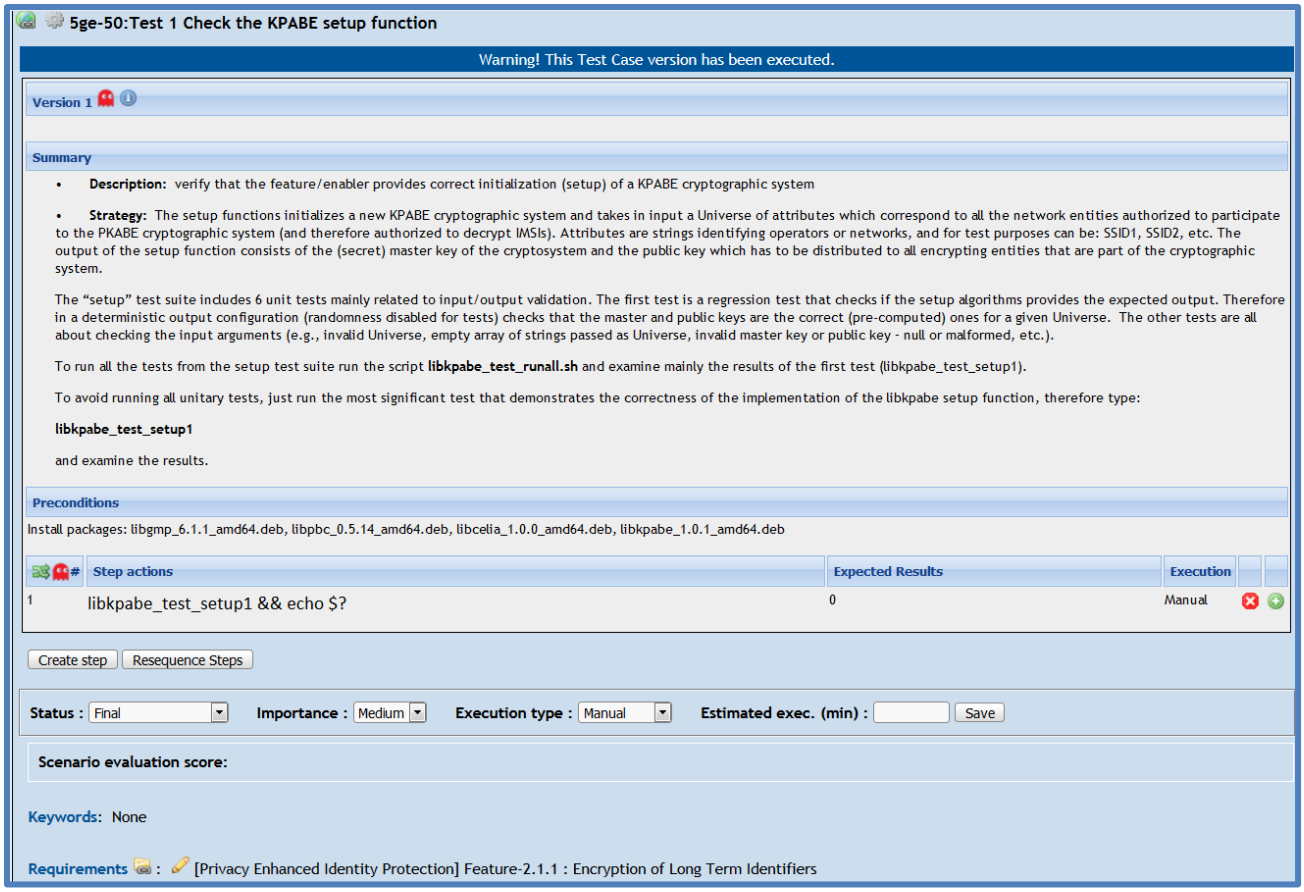


Figure 28: "Check the KPABE key generation function" test case

The following fields are required to create a test case object:

- **Test case name**: <Test name>.
- **Summary**: A description of the test scope and the expected results.
- **Preconditions**: This section should address all the requirements needed in order to grant the correct execution of the test case.
- **Steps**: A step by step sequence providing the actions to perform and the expected results for each action.
- **Requirements**: **Link to the feature** for which the test case has been defined.
- **Relations**: It is possible to put in relation several test cases if needed.

Optionally the **Status** field should reflect the status of the given test case:

- Draft: This test case is under definition.
- Ready for review: The test case can be reviewed by the testbed operator.
- Review in progress: The testbed operator checks that the test case can be executed on the testbed.
- Final: The test is in its final version and ready to be executed.

5.2.2 Enabler's security evaluation Scenarios

Figure 29 shows the test plan structure based on the security use cases and their associate threats.

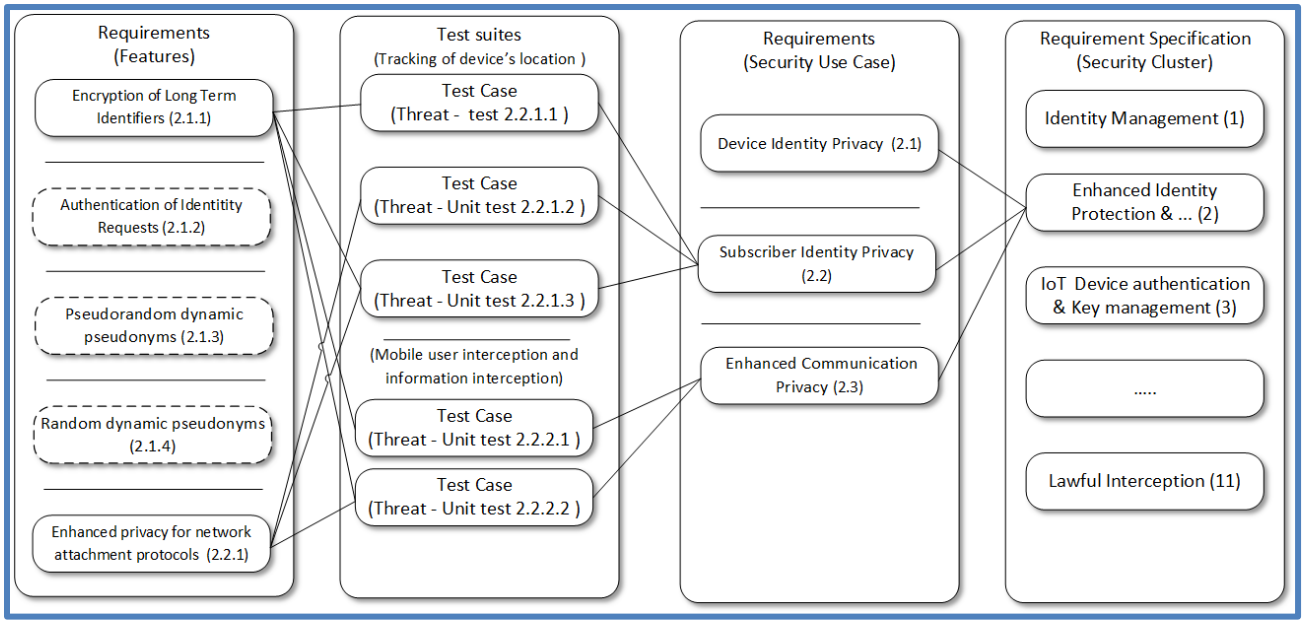


Figure 29: Testbook structure based on security use cases

Figure 29 proposes the structure for security threats "*Tracking of device's location*" and "*Mobile user interception and information interception*" from the requirements of the use case 2.2 "*Subscriber Identity Privacy*" as example. Note that the test cases will also be in relation with the enabler(s) feature(s) requirements that are supposed to tackle the security threat. By doing so, there will be an established relationship between the security use case and the enabler feature, by means of the threat test case.

The rest of the section illustrates how the proposed structure is mapped against TestLink.

Requirement Specification

Figure 30 provides an example of requirement specification for the "*Privacy Enhanced Identity Protection enabler*".

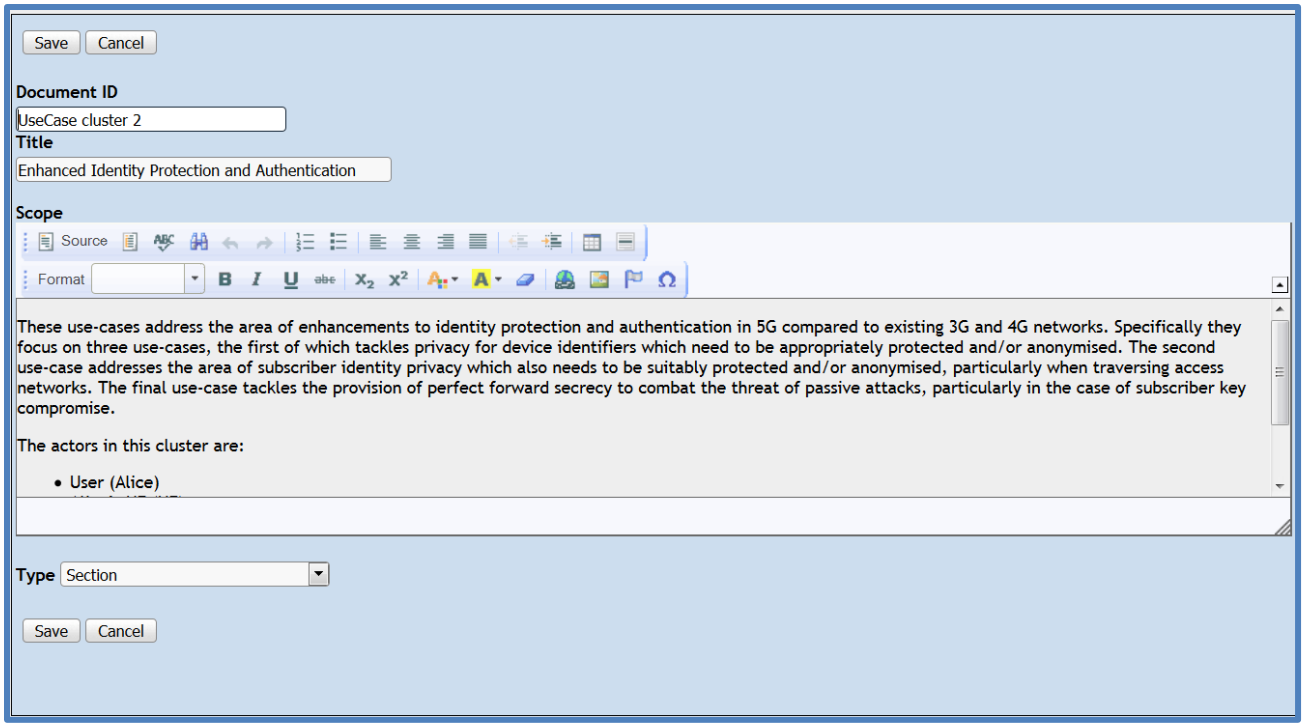


Figure 30: Requirement Specification for "Enhanced Identity Protection and Authentication" security cluster

The following fields are required to create a security cluster object:

- **Document id**: use case cluster<cluster_id>.
  - Where cluster id corresponds to the cluster defined in D2.1 [1]
- **Title:** <use case cluster name>
  - As defined in D2.1 [1]
- **Scope:** A description of the security cluster scope. In the current example, the text has been extracted from the Introduction section of the Security Cluster in D2.1 [1].
- **Type:** Section.
  - Used only to group UCs from the same cluster

Requirement

Once the requirement specification is defined, it is possible to add new requirements inside. Figure 31 provides an example of the use case definition.

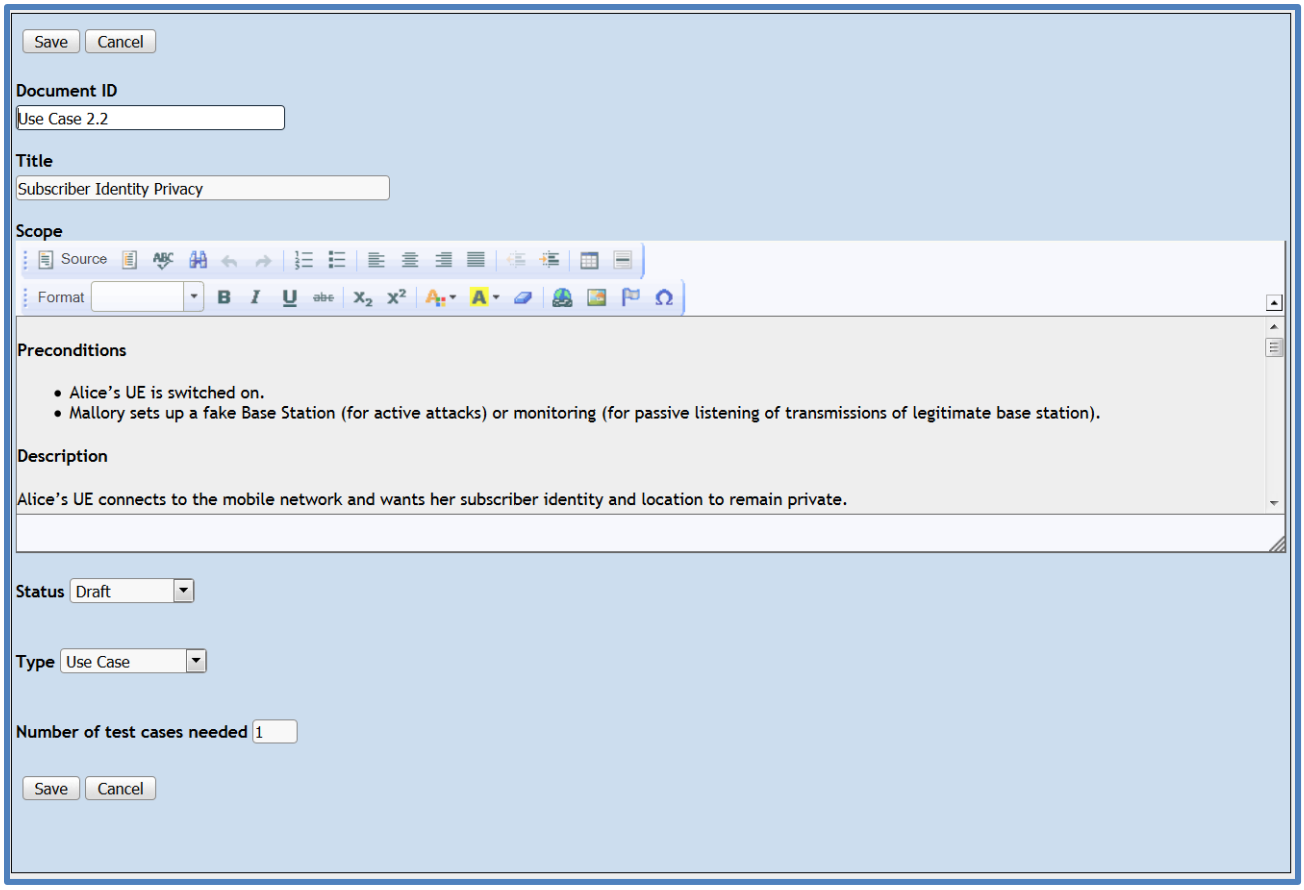


Figure 31: Use case "Subscriber Identity Privacy" requirement

The following fields are required to create a requirement object:

- **Document id**: use case <use case id>.
  - As defined in D2.1 [1].
- **Title**: <use case name>.
  - As defined in D2.1 [1].
- **Scope**: A description of the use case scope. In the current example, the text has been extracted from the use case definition present in D2.1 [1].
- **Type**: use case.

Test Suite

The test suites refers to the threats identified in D2.3 [4]. Figure 32 provides an example of the "*Mobile user interception and information interception*" threat.

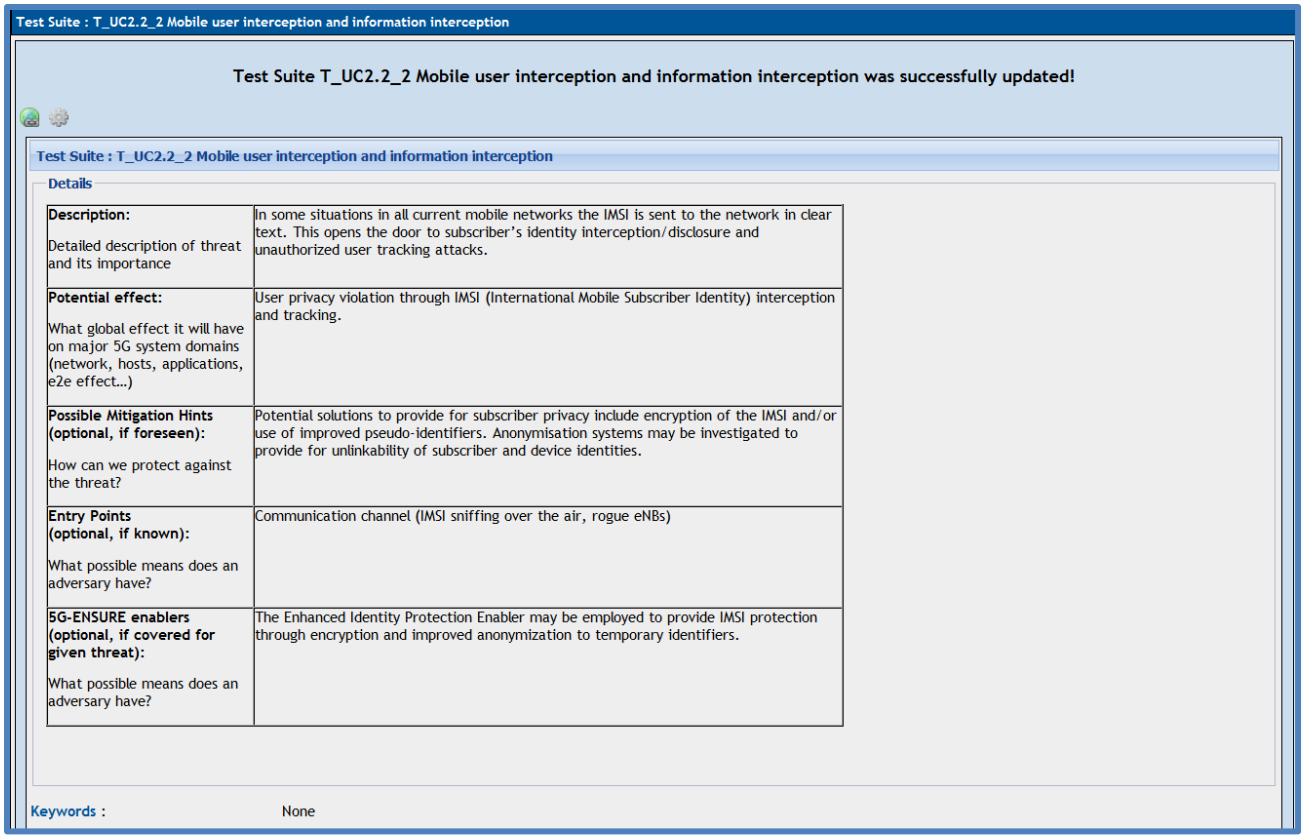


Figure 32: "T_UC2.2_2 Mobile user interception and information interception" threat test suite

The following fields are required to create a test suite object:

- **Test Suite name**: <Threat ID> <Threat name> as specified in D2.3 [4].
- **Details**: The threat description as identified in D2.3 [4]. Only the relevant fields providing a scope for the test cases definition are included.
    - **Description**
    - **Potential effect**
    - **Possible mitigation**
    - **Entry points**
    - **5G-Ensure enablers**

Test case

Once the test suite has been defined, it is possible to start adding the test cases. Test cases, as previously explained, will refer to the threat identified in the test suite. The cases should illustrate the way the enablers developed within 5G-ENSURE will cover the given threat. Figure 33 illustrates an example of what could be a test case under the "*Mobile user interception and information interception" threat,* which is called "*IMSI protection on air interface*". This test case should be tackled by the feature "*Encryption of Long Term Identifiers*"

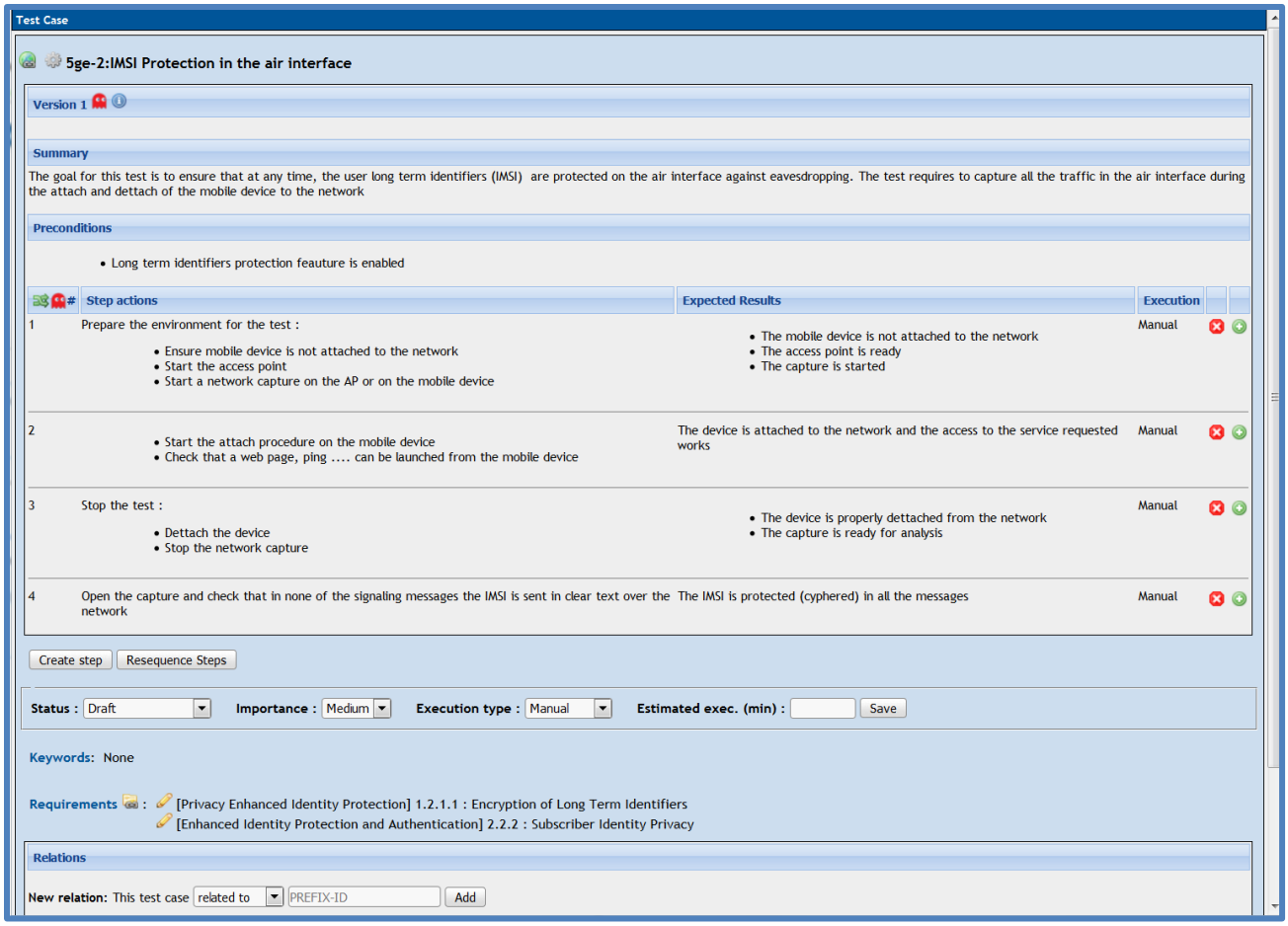


Figure 33: "IMSI protection in the air interface" test case

The following fields are required to create a test case object:

- **Test case name**: <Test name>.
- **Summary**: A description of test scope and the expected results.
- **Preconditions**: This section should address all the requirements needed in order to grant the correct execution of the test case.
- **Steps**: A step by step sequence providing the actions to perform and the expected results for each action.
- **Requirements**: **Link to feature(s) and threat** requirements for which the test case has been defined.
- **Relations**: It is possible to in relation several test cases if needed.

5.2.3 Releases management

In this section it is explained the way in which the different releases versions of a given enabler would be managed within the test plan tool.

As a premise, it is required to identify the different possible scenarios:

- The enabler proposes features only in R1.
- The enabler proposes features in R1 and R2.
- The enabler proposes features only in R2.

The first and the third case are quite straight forward as the only thing to do is to create the right objects (test cases, features requirements) and attach them to the right test plan.

In the case where the enabler provides features for both releases, there are also two different situations:

- The feature on R2 is different from R1.
- The feature in R2 is an evolution of the one of R1.

In the first case, the actions to perform are the same as for the previous case. In the second case the following things require attention:

- The requirements definition (feature) may have evolve between the D3.2 [2] and D3.6 [3]. The feature requirement needs to be updated with regard to the content of most recent document (D3.6 [3]). Anyhow it's important to create a new version of the requirement definition to keep track of the update

In any case, for an enabler that is present on both releases, it is possible that the test cases (unitary tests, evaluation Scenarios) are to be used for both test plans.

- If the test case do not need any modification, then it can be used as it is.
- If the test case requires some evolution, then it is important create a new test case version. This allows to have one version on the R1 test plan, and another version on the R2 test plan.

In order to avoid any kind of manipulation errors, all objects used for Release 1 have been upgraded in version. This allows users requiring to modify the objects, to do it without having to worry about potential issues.

5.3 Test plan design and execution

On the previous section 5.2, it is described how the different elements are mapped to Testlink objects and the information that should be provided for each of them. Once all the test cases have been defined and linked to their requirements, they can be added to one test plan in order to be executed. In this section, it will be covered the way the test plans have been designed and executed.

5.3.1 Design

The test plan campaigns are organized in a way to best fit in the project organisation. As such there are four test plans:

1. Enabler's testbed integration (Release 1).
2. Enabler's security evaluation (Release 1). Test plan included in Appendix C.
3. Enabler's testbed integration (Release 2).
4. Enabler's security evaluation (Release 2). Test plan included in Appendix 0.

Only the test plans associated to the enabler evaluation scenarios have been included. In order to improve the readability of this document, test plans designs are provided in the annexes of this document.

The test plan design follows the structures proposed in section 5.2 for each type of validation (integration, evaluation). On the Annexes of the document, only the test plan designs for enabler evaluation have been included. The following information has been exported for each test case:

- Test suite name
- Test case summary
- Test case scenario
- Test case requirements

5.3.2 Execution

During the test plan execution, for each test case, it is possible to collect the execution results of each step detailed in the test procedure, and provide a final test evaluation among the following possibilities:

- **Blocked:** the enabler cannot be run on the 5G Security testbed. This is due to the fact that the enabler was not integrated over the TestBed, or scenario description needs additional technical information in order to perform the described test (not compatible with the 5G Security testbed itself).
- **Failed**: the enabler has not passed the test suite, due to incoherence on its result or unexpected results.
- **Passed**: the enabler has successfully passed the test suite. All the theoretical scenarios which do not need to be executed on the 5G Security testbed are considered in this state on the condition that have followed the evaluation workflow mentioned in D4.3 for the WP2/WP3 evaluation process.

It is also possible to add notes for each step (i.e. execution results) and also regarding the final test evaluation.

The results collected out of the execution of the **Enablers Security Evaluation** test plans will be reported and analysed in the deliverable D4.4 *"Evaluation of the security enablers: Results and analysis of the testbed runs"*.

6 Conclusions

This deliverable defines the procedures required to evaluate the enablers' features in the testbed. It provides the test plan structures and some test case examples. The evaluation results from the test plan execution and the result analysis will be provided at the end of the project through D4.4 "Evaluation of the security enablers: Results and analysis of the Testbed runs".

At the time this document is delivered, the TestBed team and project partners demonstrate the TestBed capacity to industrially integrate enablers(delivered by WP3), in a replicable way, with traceability, procedures and process.

The proposed evaluation scheme (TCE/TFE) has been validated as operational over more than 10 enablers / features see D4.4 (§5.3 and §5.4) for detailed figures.

The work reported in this document was performed in close technical collaboration with WP2 and WP3, and based on all applicable technical deliverables already produced by the project. More specifically at the heart of work reported were considered here: an analysis of Enabler's security claims described in D3.2 [2], D3.6 [3] of each enabler's feature, but also their check against the different use cases defined in D2.1 [1], and their associated security threats identified in D2.3 [4]. Finally, this document delivers the consolidated list of threats coverage by the Enabler features (both R1 and R2 enablers).

References

[1] 5G-ENSURE, "5G-ENSURE D2.1 Uses Cases," [Online]. Available: http://5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.1-UseCases.pdf.

[2] 5G-ENSURE, "5G-ENSURE D3.2 5G-PPP security enablers open specifications (v1.0)," [Online]. Available: http://5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D3.2-5G-PPPSecurityEnablersOpenSpecifications_v1.0.pdf.

[3] 5G-ENSURE, "5G-ENSURE D3.6 5G-PPP security enablers open specifications (v2.0)," [Online]. Available: http://www.5gensure.eu/sites/default/files/5G-ENSURE_D3.6%205G-PPP%20security%20enablers%20open%20specifications%20%28v2.0%29.pdf.

[4] 5G-ENSURE, "5G-ENSURE D2.3 Risk assessment, mitigation and requirements (draft)," [Online]. Available: http://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.3-RiskAssessmentMitigationRequirements.pdf.

[5] 5G-ENSURE, "5G-ENSURE D4.1 5G Security testbed architecture," [Online]. Available: http://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D4.1-5G_Security_testbed_architecture_v1.0.pdf.

[6] 5G-ENSURE, "5G-ENSURE D3.1 5G-PPP Security Enablers Technical Roadmap early vision," [Online]. Available: http://5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D3.1-5G-PPPSecurityEnablersTechnicalRoadmap_early_vision.pdf.

[7] 5G-ENSURE, "5G-ENSURE D3.5 5G-PPP security enablers technical roadmap (Update)," [Online]. Available: http://www.5gensure.eu/sites/default/files/5G-ENSURE_D3.5%205G-PPP%20security%20enablers%20technical%20roadmap%20%28Update%29.pdf.

[8] 5G-ENSURE, "5G-ENSURE D3.5 5G-PPP security enablers technical roadmap (update)," [Online]. Available: http://www.5gensure.eu/sites/default/files/5G-ENSURE_D3.5%205G-PPP%20security%20enablers%20technical%20roadmap%20%28Update%29.pdf.

[9] "TestLink home page," [Online]. Available: http://testlink.org/.

[10] Ansible, "Ansible home page," [Online]. Available: https://www.ansible.com/.

[11] "Artifactory home page," [Online]. Available: https://www.jfrog.com/confluence/display/RTF/Welcome+to+Artifactory.

[12] 5G-ENSURE, "5G-ENSURE D3.4 5G-PPP_Security_Enablers_Documentation," [Online]. Available: http://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D3.4_5G-PPP_Security_Enablers_Documentation.pdf.

[13] “Testlink user manual,” [Online]. Available: https://wiki.openoffice.org/w/images/1/1b/Testlink_user_manual.pdf.

[14] “TestLink Screencast,” [Online]. Available: https://www.youtube.com/watch?v=6s48WGuX2WE.

[15] W. Rudin, Functional Analysis, McGraw-Hill, 1973.

[16] 5G-ENSURE, “5G-ENSURE D2.2 Trust Model (draft),” [Online]. Available: http://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.2-TrustModel.pdf.

[17] “Artifactory as a Debian repository,” [Online]. Available: https://www.jfrog.com/video/setting-up-artifactory-4-as-a-debian-repository-in-minutes/.

[18] “Artifactory as a YUM repository,” [Online]. Available: https://www.jfrog.com/video/artifactory-yum-repository/.

[19] “Artifactory as a Docker registry,” [Online]. Available: https://www.jfrog.com/video/install-artifactory-docker-registry-one-minute-less/.

[20] “Artifactury user manual,” [Online]. Available: https://www.jfrog.com/confluence/display/RTF/Welcome+to+Artifactory.

[21] 5G-ENSURE, “5G-ENSURE D3.1 5G-PPP Security Enablers Technical Roadmap early vision,” [Online]. Available: http://5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D3.1-5G-PPPSecurityEnablersTechnicalRoadmap_early_vision.pdf.

[22] 5G-ENSURE, “5G-ENSURE D2.1 Uses Cases,” [Online]. Available: http://5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.1-UseCases.pdf.

[23] 5G-ENSURE, “5G-ENSURE D3.2 5G-PPP Security Enablers Open Specifications,” [Online]. Available: http://5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D3.2-5G-PPPSecurityEnablersOpenSpecifications_v1.0.pdf.

[24] “KVM4FV,” [Online]. Available: http://artifacts.opnfv.org/kvmfornfv/docs/all/all.pdf.

[25] “IPSecS2S vpn template,” [Online]. Available: https://workspace.vtt.fi/sites/5g-ensure/Shared%20Documents/Workpackages/WP4/T4.1/Testbed/Nodes_interconnection/IPsecS2S-vpn-template.docx.

[26] “Open Air Interface,” [Online]. Available: http://www.openairinterface.org/.

[27] A. Diez, “Understanding NFV Management and Orchestration,” 2015.

[28] “ETSI NFV home page,” [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/nfv.

[29] “RHEL Virtualisation KVM timing management,” [Online]. Available: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Virtualization_Deployment_and_Administration_Guide/chap-KVM_guest_timing_management.html.

[30] “Cisco Anyconnect,” [Online]. Available: http://www.cisco.com/c/en/us/support/security/anyconnect-secure-mobility-client/tsd-products-support-series-home.html.

[31] “IPerf home page,” [Online]. Available: https://iperf.fr/.

[32] “OpenEPC Home Page,” [Online]. Available: http://www.openepc.com.

[33] VTT, “QoSMet home page,” [Online]. Available: http://www.vttresearch.com/qosmet.

[34] Wikipedia, “Wikipedia - Orchestration,” [Online]. Available: https://en.wikipedia.org/wiki/Orchestration_(computing).

[35] Wikipedia, “Wikipedia - Blackbox definition,” [Online]. Available: https://en.wikipedia.org/wiki/Black_box.

[36] B. Dictionary, “Businees Dictionary - White box,” [Online]. Available: http://www.businessdictionary.com/definition/white-box.html.

[37] 5GNorma, “5G Norma D3.1 Functional network architecture and security requirements,” [Online]. Available: https://5gnorma.5g-ppp.eu/wp-content/uploads/2016/01/5G_NORMA_D3.1.pdf.

[38] 5GNorma, “5G Norma D2.1 Use cases, scenarios and requirements,” [Online]. Available: https://5gnorma.5g-ppp.eu/wp-content/uploads/2015/11/5G-NORMA_D2.1.pdf.

[39] 5G-PPP, “5G-PPP 5G Architecture White Paper,” [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-5G-Architecture-WP-For-public-consultation.pdf.

[40] 5G-ENSURE, “5G-ENSURE D4.2 Test plan (draft),” [Online]. Available: http://www.5gensure.eu/sites/default/files/5G-ENSURE_D4.2_TestPlan_v1.0.pdf.

A Testbed Terms of Use

Foreword

The present document (hereinafter referred to as "**Terms of Use**") describes the terms of use and participation to the Testbed provided by Testbed Owners to the other participants in the 5G-ENSURE Project, as per the work packages 3 and 4 of Annex 1 of the Grant Agreement n°671562. The Project participants that have signed the present Terms of Use will hereinafter, jointly or individually, be referred to as "**Parties**" or "**Party**".

The mission of the 5G-ENSURE Testbed is to develop and test a set of useful and usable security enablers for 5G for the implementation of the Project (hereinafter referred to as "**Purpose**"). Participation to the 5G-ENSURE Testbed is subject to the rules of the Grant Agreement n°671562 and the Consortium Agreement, completed by the present Terms of Use's rules.

Two copies of the signed Terms of Use have to be submitted to both Project Coordinator and Technical Project Manager by any participant who wishes to be involved in Testbed activities.

Disclaimer

The EC flag in this deliverable is owned by the European Commission and the 5G PPP logo is owned by the 5G PPP initiative. The use of the flag and the 5G PPP logo reflects that 5G-ENSURE receives funding from the European Commission, integrated in its 5G PPP initiative. Apart from this, the European Commission or the 5G PPP initiative have no responsibility for the content.

Copyright notice

1. Definitions

As used herein and throughout these Terms of Use, the following capitalized terms, in singular or plural, shall have the meanings respectively ascribed to them below:

1. **Consortium Agreement**: the 5G-ENSURE Consortium Agreement executed on the 1st November 2015 by the Parties and the other participants of the Project as part of the European Union's Horizon 2020 research and innovation programme under the Grant Agreement n°671562.
2. **Input**: all data sets, security enablers, tools & methodologies or any other test materials input in the Testbed by the Parties.
3. **Output**: all results of the Parties' use and tests conducted on the Testbed.
4. **Project**: the research project "5G Enablers for Network and System Security and Resilience", which is part of the European Union's Horizon 2020 research and innovation programme and is governed by the Grant Agreement n°671562 executed on July 28 2015, and by the Consortium Agreement.
5. **Testbed**: the 5G-ENSURE security testbed object of these Terms of Use, implemented by the Parties as described in the scope of the work package 4 of Annex 1 of the Grant Agreement n°671562.
6. **(Testbed) Node**: set of hardware and software resources provided and operated by some Parties for the other Parties for the implementation of the Project. For the sake of clarity, a Testbed Owner's corporate infrastructure which is interconnected with the Testbed remains outside of the scope of these Terms of Use, and no Party except that Testbed Owner is authorized to access it.
7. **Testbed User**: a Party requiring to run activities on or have access to the 5G-ENSURE Testbed.
8. **Testbed Owner**: a Party providing at least one of the Nodes hosting the 5G-ENSURE security Testbed.
9. **Testbed Operator**: a Party managing enablers' deployment within Testbed and the operational status of the infrastructure.
10. **Enabler Owner**: a Party owning one or more of the 5G security enablers produced in the course of the Project.

Other capitalized words of the Terms of Use that are not defined in the present Terms of Use shall have the meaning attributed to them in the Consortium Agreement or, if not defined in the Consortium Agreement, in the Grant Agreement n°671562.

2. Technical undertakings

2.1. Use

Any involvement in the Testbed entails the following commitments:

- Limited use: The Parties may only use the Testbed in accordance with the Purpose. The Testbed Operators have the possibility to restrict the access of a Party to resources if the Testbed is not used in compliance with the present Terms of Use by that Party.
- Property rights: A Party shall not incorporate in the Testbed or use any information or intellectual property rights that are owned by a third party, unless that Party has first secured a right to do so.
- Input inventory: Any Party that considers providing Input to the Testbed has to send the Technical Project Manager and the Work Package 4 Leader a written comprehensive description of the Input considered, of its characteristics and of how to use it (i.e. for enablers this includes the software release and the documentation including Installation and Administration Guides; User and Programmers Guides; Unit Testing Plan as well the Unit Tests Report to certify that the enabler successfully passed the sanity checks). Once an Input has been validated by Technical Project Manager and the Work Package 4 Leader and assigned to the Testbed, this Input cannot be removed without the prior approval of Technical Project Manager. Input shall not contain any personal data.
- Configuration: The configuration of the Testbed is implemented by the Testbed Operators in accordance with the agreed test plan between Enabler Owners, Testbed Operators and Testbed Owners. All Testbed Users shall be given by the Testbed Operators the possibility to check that the required configuration is effective (right to access and read the configuration).
- Additional documentation: Documentation (such as user manuals or detailed interface descriptions) about the Testbed, restricted to essential information required to prepare and execute the tests, can be provided by Testbed Owners to Testbed Users which ask for it during the setup phase.

- Login credentials: A Party's login credentials for connection to the Testbed are provided by b<>com as primary Testbed Operator on a confidential basis and may not be disclosed to anyone.
- Planning: Any tests and experiments must be planned in advance during time periods agreed by the involved Testbed Operator, in order for the latter to book the resources required by the test session.

2.2. Nodes interconnection

Testbed Owners may interconnect their own remote testbed Nodes to the 5G-ENSURE Testbed subject to the following rules:

- Nodes inventory: Any Testbed Owner that considers providing Nodes to the Testbed has to send the Parties a written comprehensive description of the Node, the manpower that it will commit to operating the Node, and the elements considered, of their characteristics and of how to use them. It is the responsibility of the 5G-ENSURE Steering Committee to validate and manage any change in the Testbed Nodes allocation. Once assigned to the Testbed, Nodes cannot be removed without the prior approval of Technical Project Manager.
- Accurate development / availability of resources: The Testbed Owners shall do their reasonable efforts to ensure that their Nodes operate properly that they are reliable and secure, and that they are available in the timetable that they provided to the Parties.
- Data update cycle: The Testbed Owners must do their reasonable efforts to provide Nodes that are coherent, completely operative and regularly updated.
- Audit procedure: Technical Project Manager may choose to conduct an audit procedure to observe and inspect the controls, compliance, performance, etc. of a Testbed Owner's Node and Input. All Testbed Owners are required to comply with any audit procedure and any recommended corrective actions, which are compatible with the Purpose, that Technical Project Manager may come to suggest.
- Disclaimer of liability: Testbed Users understand that the Testbed Nodes are provided for experimental purposes. Even if the Testbed has been designed to meet Enabler Owners requirements especially in term of hosting criteria, Testbed Owners do not warrant that the network functions hosted in the Testbed Nodes will meet any Testbed User's expectation in term of end-to-end integration or performance. Testbed Users shall understand that the Testbed Node may be experimental, may have not been thoroughly tested, and may contain defects. The Testbed Node and related materials are provided "as is," and Testbed Owners make no warranties or representations, express or implied, written or oral, statutory or otherwise, regarding the use, operation, or performance, including, without limitation, warranties of non-infringement, merchantability, or fitness for a purpose. Testbed Users' use of the Testbed Nodes is at Testbed Users' sole risk, and Testbed User has sole responsibility for adequate protection and back-up of its data used in connection with the tests, even if the Testbed is providing tools to achieve this back-up.

2.3. Access and support

The Access to the Testbed and the Testbed Support are subject to the following rules:

- Access: The Testbed Operators undertake to do their reasonable efforts to provide a reliable remote access to the Testbed from a Party's remote facility, and to provide inter-connectivity with Testbed Owners' Nodes, as well as comply with all their obligations derived from the Grant Agreement n°671562 and the Consortium Agreement. Testbed Operators cannot guarantee however a 24/7 availability of the Testbed, especially as the operation of all the Testbed's Nodes relies on several different entities at the same time.
- Support: The Testbed Owners and the Testbed Operators undertake to do their reasonable efforts to provide the other Parties with assistance in case of technical problems in relation to the Testbed.
    - Ticket is classified in five levels to be used for notification of Testbed Owners and Operators in case of assistance request:
        - "Very high": A request is "Very high" when it leads to the inoperability of the service and no fallback or workaround solution is available.
        - "High", "Medium": A request is "High" or "Medium" when it leads to a limitation of the functionalities or the performances of the service, or to the necessity to use fallbacks mechanisms or workarounds.
        - "Low", "Very Low": A request is "Low" or "Very Low" when it has no operational impact but leads to difficulties to operate the service.
    - In order for Parties to access the helpdesk and submit Problem Reports, b<>com as primary Testbed Operator provides an online web portal accessible at: https://helpdesk.b-secure.irt-b-com.org.

- Testbed Owners' and Operators' operation staff is on duty 5 days / week (week-ends, bank holidays and days-off not included), at 9:30/12:00 – 14:00/17:30 CET/CEST Time.
- If a Testbed Owner or a Testbed Operator is not able to fix a notified issue related to one of the Nodes it provides or operates within ten (10) calendar days, it has to warn Technical Project Manager and the other Parties on a timely basis about the impact of the issue on the Testbed. After this period and if the issue is critical, the Parties can decide to refer the issue to Technical Project Manager.

3. Confidentiality

Confidential information exchanged in the context of the present Terms of Use remains subject to the confidentiality obligations of the 5G-ENSURE Consortium Agreement. Consequently, all information in whatever form or mode of communication, which is disclosed by a Party or another participant in the 5G-ENSURE Project (the "**Disclosing Party**") to any other Party or participant in the 5G-ENSURE Project (the "**Recipient**") in connection with the Project or with the Testbed, and which is marked or otherwise identified as confidential at or prior to the time of disclosure, or when disclosed orally has been identified as confidential at the time of disclosure and has been confirmed and designated in writing within thirty (30) days from oral disclosure at the latest as Confidential Information by the Disclosing Party, is "**Confidential Information**".

For the sake of clarity, Confidential Information shall be deemed to include also all Input and all Output of the Testbed, as well as all information provided by Testbed Owners and Operators in order to enable the Parties to operate and use the Testbed.

As stipulated in the Consortium Agreement, the confidentiality obligations undertaken by the Parties for the Project shall remain in force during the Project and for a period of four (4) years after the end of the Project.

4. Intellectual property & Access Rights

The Parties' intellectual property and Confidential Information (be it Background, Input, Output, Results, etc.) that is used in the context of the present Terms of Use remain subject to the relevant obligations of the 5G-ENSURE Consortium Agreement.

Consequently, Output shall be owned by the Party who generated it.

For the sake of clarity, when a Testbed User carries out a test by itself, with its own Input, and if the Testbed is simply put at its disposal by the Testbed Owners without further operation, the Testbed User shall remain the owner of the Output that it generates.

Two or more Parties shall own Output jointly if:

- They have jointly generated the Output in question; and
- It is not possible to:
  - establish the respective contribution of each Party; or
  - separate each Party's part of the Output for the purpose of applying for, obtaining or maintaining protection.

Each joint owner shall have an equal, undivided interest in and to a joint Output as well as in and to resulting Intellectual Property Rights in all countries. Each of the joint owners and their Affiliated Entities shall be entitled to Exploit the jointly owned Output as they see fit, and shall be entitled to grant non-exclusive licenses, without obtaining any consent from, paying compensation to, or otherwise accounting to any other joint owner(s).

For the sake of clarity, Access Rights applicable to Input and Output in accordance with the 5G-ENSURE Consortium Agreement are the following:

- Access Rights to Input and Output of the Testbed needed for the implementation of the Project are hereby requested, and shall be deemed granted, as of the Effective Date, on a royalty-free basis to and by all Parties, and shall either terminate upon completion of the Project or upon termination of a Party's participation to the Testbed.
- Access Rights to Output of the Testbed needed for internal research, development and teaching are hereby requested and shall be deemed granted as of the date of the Output arising, on a royalty-free basis to and by all Parties.
- Access Rights to Output of the Testbed needed for any other Exploitation (including as needed for Use of a Party's own Results) shall be granted on Fair and Reasonable Conditions subject to the conditions of the Consortium Agreement.

5. Duration

The present Terms of Use shall have effect from 1st of August 2016 ("**Effective Date**"). The Terms of Use's obligations will remain in force for the duration of the Project or, where relevant, the duration assigned to them by the Consortium Agreement or the Grant Agreement n°671562, whichever is greater.

6. Applicable law and disputes

These terms of Use shall be construed in accordance with and governed by the Laws of Belgium excluding its conflict of laws principles.

The Parties shall reasonably endeavor to settle their disputes amicably. If however no settlement of disputes under these Terms of Use has been possible to achieve, the relevant provisions of the Consortium Agreement shall apply

****

The undersigned hereby acknowledges that it has read and understood the present Terms of Use and agrees to be bound by all their rules as well as by the other rules applicable to the 5G-ENSURE Project (i.e. Grant Agreement n°671562 and Consortium Agreement).

Party: ……..…..

_____

By: ……..…..

Title: ……..…..

Date: ……..…..

B Ansible roles for testbed enabler deployment

Ansible is the software used to automate both the testbed deployment and the enabler's integration.

Each configuration step is described in a task. The configured tasks are listed in what are called plays along with the hosts targeted by the tasks. The plays are regrouped in playbooks. To keep tasks organized and facilitate reusability, they can be regrouped in roles.

Hereunder are listed the roles thatare so far created/available. Altogether they relate to admin roles created for enablers integrated/deployed on the testbed:

Table 7: Ansible roles for enablers integration

Role name	Role description
5ge-enabler-acm_controller	Role to install and configure the Access Control Mechanism Enabler controller : o Install Onos o Start Onos service and enable it on boot
5ge-enabler-bootstrapping_trust	Role to install and configure 5G-Ensure Bootstrapping Trust Enabler o Enable kernel ima feature on boot o Install opensgx and openvswitch-tswitch packages o Add openvswitch to /etc/modules
5ge-enabler-cia_compliance_checker	Role to install and configure the 5G-Ensure Components Interactions Audit enabler Compliance checker feature on a host o Install runverif package
5ge-enabler-cia_controller	Role that installs and configures the 5G-Ensure Components Interactions Audit Enabler Controller feature on a host o Give higher priority to packages from Artifactory o Install openvswitch
5ge-enabler-device_identifier_privacy	Role to deploy and configure 5G-Ensure Device Identifier Privacy Enabler o Install required packages o isc-dhcp-server o iw o wireless-tools o lxc o rfkill o wpasupplicant o hostapd o dhcpd-dip o dip-integrate-scripts o add hwsim to /etc/modules o configure hwsim o configure isc-dhcp-server o configure wpa-supplicant o configure dhcpd o configure virtual wireless interfaces
5ge-enabler-fga_rcd	Role to install the 5G-Ensure Fine Grained Authorization (RCD Feature) Enabler. o Install enabler packages: o fga-rcd-authentication o fga-rcd-authorization

5ge-enabler-fga_satellite_server	Role to install and configure 5G-ENSURE Fine Grained Authorization Enabler Satellite Feature server on a host. o Install the finegrainedauthorization package
5ge-enabler-iot_hss	Role to deploy HSS component of 5G-Ensure IoT Enabler o install enabler packages: o group-auth-freediameter o group-auth-asn1c o group-auth-hss o adapt /etc/hosts file to the vm fqdn o provision database o setup hss configuration files o generate openssl certificates
5ge-enabler-iot_mme	Role to install and configure MME component of 5G-Ensure IoT Enabler o install enabler packages: o group-auth-freediameter o group-auth-asn1c o group-auth-gtpu o group-auth-mme o adapt /etc/hosts file to the vm fqdn o setup freediameter configuration files o setup epc configuration files o generate openssl certificates
5ge-enabler-iot_sim	Role to install and configure SIM component of 5G-Ensure IoT Enabler o install enabler packages: o group-auth-oaisim o setup oaisim configuration files
5ge-enabler-microsegmentation	Role to install the 5G-Ensure Microsegmentation enabler. o Install microsegmentation package
5ge-enabler-microsegment_monitoring	Role to install and configure 5G-Ensure Microsegment Monitoring Enabler o Install microsegmentmonitoring package
5ge-enabler-peip_elti	Role to install and configure 5G-ENSURE Privacy Enhanced Identity Protection Enabler Encryption of Long Term Identifiers feature o Install required packages o wpa-supplicant o hostapd o libcelia o libgmp o libkpabe o libpbc o libglib2.0-dev o libgmp3-dev
5ge-enabler-pulsar	Role to install and configure 5G-Ensure Pulsar Enabler o pull docker image pulsar o run docker container from image and expose ports
5ge-enabler-satellite_network_monitoring_server	Role that installs and configures 5G-Ensure Satellite Network Monitoring Enabler server. o Install package satellitenetworkmonitoring
5ge-enabler-trustbuilder	Role to install the 5G-Ensure Trust Builder enabler. o Install Java8 o Install mongodb o Install Tomcat 7

	○ Make tomcat use java8 ○ Install system-modeller from Artifactory ○ Ensure tomcat is running and enabled on boot
5ge-enabler-trustmetric	Role to install and configure the 5G-Ensure Trust Metric Enabler ○ Install trustmetric package

Notice: The above described roles are available at the project's workspace for internal usage.

C Test plan design: Enabler's security evaluation (R1)

Use Cases cluster 1 - Identity Management

T_UC1.3_1 Unauthorised activities related to satellite devices or network

Description: Detailed description of threat and its importance	Network Operators (e.g. SatNO) and M2M communications (e.g. updated satellite device SW) require fine-grained access to network resources (e.g. satellite device, eNB…). Also, satellite devices shall be authenticated to access satellite services (e.g. broadband access, direct-to-home services…). These network components and devices are distributed in a wide-area large enough that other wired or wireless network connectivity is not feasible. In this scenario, main threats are related to Unauthorised activities: Unauthorised access Unauthorised administration of devices and systems Falsifications of configurations
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	Information integrity. Information destruction. Service availability.
Possible Mitigation Hints (optional, if foreseen): How can we protect against the threat?	Fine-grained access control focusing on the application level. In case of resource constrain devices (e.g. satellite devices), the fine-grained access control can be based on tokens evaluated directly in the device.
Entry Points (optional, if known): What possible means does an adversary have?	Non updated network components or satellite devices compromise system security/functionality. Wide-area distributed network composed of resource constrained devices (i.e. satellite devices) with high latency.
5G-ENSURE enablers (optional, if covered for given threat): What possible means does an adversary have?	Fine-grained Authorization enabler.

Test Case 5ge-130: Unauthorised user verification
Summary: An authorized user tries to make a rest petition on a non-authorized resource. The response of the test, should be that the user is not allowed to perform the operation because the conditions of the petition does not match with the rules applied in the policy (i.e. $FGA_SAT_PATH/test/UT01/input/TestPolicy_UT01a.xml).

Conditions: - The user is registered in the LDAP server. - The user is authorized to perform this action. - The user is non-authorized to perform this action on this resource.		
Preconditions: Execute 5ge-54 in order to install and configure the environment to run.		
#:	Step actions:	Expected Results:
1	This step aims to initialize the PAP policies repository with test policy (i.e. $FGA_SAT_PATH/test/UT01/input/TestPolicy_UT01a.xml): The PUT action can be done by Christopher Carroll on resource fga-sat-rcd/api/v01.00.00/satelliteModem/mgmt/startCWCarrier. curl -v -X POST -H "Content-Type: application/xml" -H "Accept: application/json" --data "@UT01/input/TestPolicy_UT01a.xml" http://5g-fga-sat-srv01.5g-ensure.eu:8080/fga-sat-srv/api/v01.00.00/pap/policy	Expected result is HTTP/1.1 status code 201 with the following response body: {"header":{"responseCode":0,"msgType":"restResponseMC"},"content":{}} Inside $FGA_SAT_PATH/server/policies there should be the uploaded policy.
2	Not authorized request for "PUT" action on resource http://5g-fga-sat-cli01.5g-ensure.eu:8080/fga-sat-rcd/api/v01.00.00/satelliteModem/mgmt/stopCWCarrier. The contents of the $FGA_SAT_PATH/test/UT03/input/RequestContent_UT03b.json file is: {"userName":"Christopher Carroll","action":"PUT","resource":"http://5g-fga-sat-cli01.5g-ensure.eu:8080/fga-sat-rcd/api/v01.00.00/satelliteModem/mgmt/stopCWCarrier","content":{"header":{},"content":{"outputPower":1234,"frequencyParameter":4567,"action":"on","blind":false}}} Request resource action from the RCD using the server host as an authorization proxy: curl -v -X POST -H "Authorization: 5G-ENSURE base64(TWlsZHJlZCBEdW5u:bWlEdV81Zw==)" -H "Content-Type: application/json" -H "Accept: application/json" --data "@UT03/input/RequestContent_UT03b.json" http://5g-fga-sat-srv01.5g-ensure.eu:8080/fga-sat-srv/api/v01.00.00/pdp/authorize	Expected result is HTTP/1.1 status code 401 with the following response body: {"header":{"responseCode":100,"errorMsg":"Request is NOT authorized to perform this access","msgType":"restResponseMC"},"content":{}}
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Relations	depends on - 5ge-54:Installing and configure environment related to - 5ge-136:Authorised user verification	
Requirements	Feature-1.2.1: Basic Authorization in Satellite systems Use Case 1.3: Satellite Identity Management for 5G Access	

T_UC1.4_1 Compromised data

Description: Detailed description of threat and its importance	In this use case, the MNO needs to collect data about a user from the mobile network (step (c) in Figure 5 of Deliverable D2.1). If the user device or any network component is compromised, this can tamper with the integrity and confidentiality of the collected data. As the metrics provided to the service provider are cryptographically computed based on the collected data, collecting fake data may compromise the metrics, hence, the provided service.
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	In order to provide this enhanced service, the MNO needs to have an assurance about the validity of the collected data. This may imply the use of attestation protocols between the collect points (in the network) and the MNO.
Possible Mitigation Hints (optional, if foreseen): How can we protect against the threat?	In order to protect against this threat, the MNO needs to perform validity checks on the collected data. The solution may include remote attestation protocols and investigation in statistics data processing.
Entry Points (optional, if known): What possible means does an adversary have?	An adversary can have one or all the following means: Communication channels, user equipment and a network component
5G-ENSURE enablers (optional, if covered for given threat): What possible means does an adversary have?	Generic collector interface enabler can be part of the solution.

Test Case 5ge-87: ProVerif security analysis of the group-based AKA protocol
Summary: Feature 1.1.1 is a group-based Authentication and Key Agreement (AKA) protocol in which group authentication parameters are stored on the device outside of the UICC. However, the symmetric long-term key K, which is stored on the UICC, is also used in the protocol. Since parameters stored outside of the UICC could easily be leaked, the fundamental security properties of the protocol must not depend on whether the group authentication parameters are compromised or not. Specifically, an adversary having access to the group authentication parameters must be unable to authenticate to the network or derive a session master key by eavesdropping on communication. If the adversary could manage to derive the session master key, the confidentiality of all the data sent between the machine-type communications (MTC) device and the network would be compromised. Also, the adversary should not be able to break authentication or confidentiality even if, additionally, members of the same group share all its authentication parameters (including the long-term secret) with the adversary. It is proven with ProVerif that the protocol meets confidentiality and mutual authentication when the adversary has access to all the authentication parameters of members in the same group in addition to all group authentication parameters of the MTC device. See the following paper for a presentation of the proof.

Giustolisi, R., Gehrmann, C., Ahlström, M. and Holmberg, S., 2017. A secure group-based AKA protocol for machine-type communications. In *International Conference on Information Security and Cryptology ICISC 2016: Information Security and Cryptology–ICISC 2016. 30 November 2016 through 2 December 2016* (pp. 3-27).

Execution type:	Manual
Estimated exec. duration (min):	
Priority:	Medium
Scenario evaluation score:	**1- Theoretical evidence**
Requirements	Feature-1.1.1: Group authentication by extending the LTE-AKA protocol Use Case 1.4: MNO Identity Management Service
Attached files	A secure group-based AKA protocol for machine-type communications : icisc_cameraready.pdf

Test Case 5ge-66: Colluding corrupted devices

Summary:

Checks that an MTC device B in the same group as an MTC device A cannot authenticate to the network by using A's group authentication parameters instead of its own and that it cannot derive a session master key.

Preconditions:

The “Group authentication by extending the LTE-AKA protocol” feature of the Internet of Things enabler is installed.

#:	Step actions:	Expected Results:
1	On the HSS virtual machine, execute “/usr/local/etc/oai/SCRIPTS/run_hss”	“Initializing s6a layer: DONE” is printed.
2	On the MME virtual machine: Execute: “/usr/local/etc/oai/SCRIPTS/run_epc -i” If the execution pauses after “eth0:1 is virtual interface” is printed, enter the sudo password.	On the MME virtual machine: “'STATE_WAITCEA' -> 'STATE_OPEN'” is printed eventually (after a lot of other printouts). On the HSS virtual machine: "'STATE_OPEN_NEW' -> 'STATE_OPEN'" is printed.
3	On the oaisim virtual machine, execute “sudo -E usr/local/etc/oai/cmake_targets/tools/run_enb_ue_virt_s1” and then wait until the terminal output on the three virtual machines has stabilized.	The terminal output has stabilized on the the three virtual machines.
4	Stop the executions of the MME, the HSS and oaisim by using Control-C three times (or some of the three executions might stop automatically).	The executions of MME, HSS and oaisim stop.
5	On the MME virtual machine, execute “ls /usr/local/etc/oai/SCRIPTS”.	There exists a file named “group_info” in

		/usr/local/etc/oai/SCRIPTS on the MME virtual machine.
6	On the HSS virtual machine, execute “usr/local/etc/oai/SCRIPTS/run_hss”.	“Initializing s6a layer: DONE” is printed.
7	On the MME virtual machine: Execute: “/usr/local/etc/oai/SCRIPTS/run_epc -i | tee ~/caseB_deviceA.log” If the execution pauses after “eth0:1 is virtual interface” is printed, enter the sudo password.	On the MME virtual machine: “'STATE_WAITCEA' -> 'STATE_OPEN'” is printed eventually (after a lot of other printouts). On the HSS virtual machine: "'STATE_OPEN_NEW' -> 'STATE_OPEN'” is printed.
8	On the oaisim virtual machine, execute “sudo -E /usr/local/etc/oai/cmake_targets/tools/run_enb_ue_virt_s1 | tee ~/caseB _deviceA.log” and then wait until the terminal output has stabilized on the three virtual machines.	The terminal output has stabilized on the three virtual machines.
9	Stop the executions of the MME, the HSS and oaisim by using Control-C three times (or some of the three executions might stop automatically).	The executions of MME, HSS and oaisim stop.
10	On the MME, execute “cat ~/caseB_deviceA.log | grep kasme”. On oaisim, execute “cat ~/caseB_deviceA.log | grep kasme”.	The same kasme is printed on both the MME and oaisim, which means that the UE and the MME has successfully agreed a master session key.
11	On the MME, execute “cat ~/caseB_deviceA.log | grep XRES”.	On the MME terminal is printed “Success to authentify the UE”.
12	On the HSS virtual machine, execute “usr/local/etc/oai/SCRIPTS/run_hss”.	“Initializing s6a layer: DONE” is printed.
13	On the MME virtual machine: Execute: “/usr/local/etc/oai/SCRIPTS/run_epc -i | tee ~/caseB_deviceB.log” If the execution pauses after “eth0:1 is virtual interface” is printed, enter the sudo password.	On the MME virtual machine: “'STATE_WAITCEA' -> 'STATE_OPEN'” is printed eventually (after a lot of other printouts). On the HSS virtual machine: "'STATE_OPEN_NEW' -> 'STATE_OPEN'” is printed.
14	On the oaisim virtual machine, execute “sudo -E /usr/local/etc/oai/cmake_targets/tools/run_enb_ue_virt_s1 --corrupted_device | tee ~caseB_deviceB.log” and then until the terminal output has stabilized on the three virtual machines.	The console output has stabilized on the three virtual machines.

15	Stop the executions of the MME, the HSS and oaisim by using Control-C three times (or some of the executions might stop automatically).	The executions of MME, HSS and oaisim stop.
16	On the MME, execute “cat ~/caseB_deviceB.log | grep kasme”. On oaisim, execute “cat ~/caseB_deviceB.log | grep kasme”.	The same kasme is not printed on both the MME and oaisim, which means that the UE and the MME has not agreed a master session key.
17	On the MME, execute “cat ~/caseB_deviceB.log | grep XRES”.	On the MME terminal is not printed “Success to authentify the UE”.
Execution type:	Manual	
Estimated exec. duration (min):	30.00	
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Feature-1.1.1: Group authentication by extending the LTE-AKA protocol Use Case 1.4: MNO Identity Management Service	
Attached files	T_UC1.4_1 threat coverage : Wp2-Wp4_threat_coverage_evaluation_T_UC1.4_1_Compromised data_IoT_enabler.docx Wp2-Wp4_threat_coverage_evaluation_T_UC1.4_1_Compromised data_IoT_enabler.docx	

Use Cases cluster 2 - Enhanced Identity Protection and Authentication

T_UC2.2_1 Tracking of device's (user's) location

Covered Threats	**T_UC2.2_1 and T_UC2.2_2**
Description: Detailed description of threat and its importance and The EO interpretation of the threat if needed	In some procedures (e.g., initial network attach, paging requests, etc.) in all current mobile networks the IMSI is sent to the network in clear text. This opens the door to subscriber's identity interception/disclosure and unauthorized user tracking attacks. **EO interpretation of the threat**: Sending the subscribers' identifiers (IMSIs) in clear text over the air interface may allow the corresponding user information interception, therefore such identifiers must be concealed (e.g., through encryption).
Potential effect: The effect of the threat on major 5G system/domains	If the 5G network is not able to protect end-user's privacy will be considered less trustworthy by the end-users. The threat mainly affects the AN (Access Network) domain.
Threat Mitigation How can we protect against the threat?	Use of encrypted identifiers when possible. However, devices need to be aware that the communication is targeted for them, so encrypted identifier will become a pseudo-identifier that can be mapped to the device. Frequent changing of temporary identifiers (preferably by using one time temporary identifiers).
Entry Points (optional, if known): Attack Scenarios: what possible means does an adversary have? Also specify attack pre-conditions if any and choose the most feasible/probable attack scenarios. Try to identify a basic attack if possible the one that will be tested by the Test Suite proposed for	Basic attack: passive sniffing of signaling traffic (in the specific test proposed for the feature developed in Release 1 this means the sniffing of Identity Responses on the WiFi interface). Adversaries must link subscription identifiers to the users' identity. This can be achieved by triggering the mobile network into initiating the generation of paging messages to the victim (and thus to victim's terminal). For instance, adversaries may connect users with using social media application to initiate unobtrusive communications. Location tracking can be done at the granularity of base station's coverage or in more detail if the adversary has capabilities to analyse signal directions. Also, detailed location tracking is possible by eavesdropping plaintext signal measurement reports.

evaluation.	
5G-ENSURE enablers and features that cover the threat	Enabler: Privacy Enhanced Identity Protection (PEIP) Feature: Encryption of Lont Term Identifiers

Test Case 5ge-62: Subscriber's Identity Protection, through Encryption, over the air interface		
Summary: The goal for this test is to ensure that at any time, the user long term identifiers (IMSI) are protected over the air interface against eavesdropping (passive or active sniffing). The test requires to capture all the traffic over the air interface during successive attaches of the mobile device to the wireless network.		
Preconditions: Encryption of Long Term Identifiers feauture is enabled The 5ge-53 - "Test5 Check the KPABE encryption properties with Wireshark" test scenario is active with all the confguration settings on		
#:	Step actions:	Expected Results:
1	Prepare the environment for the test (the configurations are described in detailed in the test case 5ge-53: Test 5): Ensure mobile device is not attached to the network Start the access point Start a network capture on the AP or on the mobile device	The mobile device is not attached to the network The access point is ready The capture is started
3	Start the attach procedure on the mobile device Check that a web page, ping .... can be launched from the mobile device	The device is attached to the network and the access to the service requested works
4	Stop the test : Dettach the device Stop the network capture	The device is properly dettached from the network The capture is ready for analysis
5	Open the capture and check that in none of the signaling messages the IMSI is sent in clear text over the network	The IMSI is protected (cyphered) in all the messages
6	Repeate steps 3 - 5 to verify that the encrypted IMSI value is different from the one observed in the previous attach procedure.	The encrypted IMSI value is different from the one captured during the previous attach. This guarantees that the attacker (eavesdropper that passively sniff the traffic) cannot track the mobile device.

7	Repeate steps 3 - 5 to verify that the encrypted IMSI value is different from the one observed in the previous attach procedure.	The encrypted IMSI value is different from the one captured during the previous attach. This guarantees that the attacker (eavesdropper that passively sniff the traffic) cannot track the mobile device.
Execution type:	Manual	
Estimated exec. duration (min):	25.00	
Priority:	High	
Scenario evaluation score:	**4 - Testbed evaluation (real flows)**	
Relations	related to - 5ge-2:Subscriber's Identity Protection (Encryption) over the air interface	
Requirements	Feature-2.1.1: Encryption of Long Term Identifiers Use Case 2.2: Subscriber Identity Privacy	
Attached files	IMSIEncryption_WP2-WP4_threat_coverage.docx	

Test Case 5ge-63: Encryption Tests (optional if 5ge-51 already done)

Summary:

Executing these tests is optional, since they were performed in the previous test validation session. However, they demostrate the threat T_UC2.2_1 threat coverage and therefore we mention it again here.

libkpabe_test_enc3: This test is about matching cryptographic material: the encryption attribute is correct, but the wrong public key is passed. The test checks that the IMSI is not encrypted if the attribute does not match the public key, and the function exits with the appropriate error message without crashing. Therefore, in a scenario where the public key was corrupted (on the encrypted device) the IMSI cannot be encrypted.

libkpabe_test_enc9: In this test an attribute that was not included in the initial Universe is passed as an input to the encryption function. The test checks that the IMSI is not encrypted if the attribute does not match the public key. Therefore, if an attacker provides a fake attribute the client device will not encrypt the IMSI.

libkpabe_test_enc11: This test checks that a given IMSI has different encrypted outputs as a result of different successive applications of the encryption functions with the same attribute (for the same authorized network entity). This is a proof for the non-tracking requirement.

libkpabe_test_enc12: This checks that the output of the encryption of a given IMSI with a given attribute_1 is different from a second encryption output of the same IMSI with attribute_2. Therefore the user cannot be tracked if he/she changes the network attachment point.

Note: this test case contains all the 4 tests of the test suite 5ge-51: Test 3 Check the KPABE encryption function.

Preconditions:

Install packages: libgmp_6.1.1_amd64.deb, libpbc_0.5.14_amd64.deb, libcelia_1.0.0_amd64.deb, libkpabe_1.0.1_amd64.deb

#:	Step actions:	Expected Results:

2		
3	Run all 4 tests from the "5ge-51: Test 3 Check the KPABE encryption function".	The expected results are illustrated in the steps of "5ge-51: Test 3 Check the KPABE encryption function".
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Feature-2.1.1: Encryption of Long Term Identifiers Use Case 2.2: Subscriber Identity Privacy	

Test Case 5ge-64: Decryption Tests (optional if 5ge-52 already done)

Summary:

Executing these tests is optional, since they were performed in the previous test validation session. However, they demostrate the threat T_UC2.2_1 threat coverage and therefore we mention it again here.

libkpabe_test_dec5: In this test a private key that does not match with the encryption attribute (as a matter of fact the key does not match with any attribute in the Universe) is passed to the decryption function. Therefore a successful test guarantees that an encrypted IMSI cannot be decrypted with an unauthorized private key.

libkpabe_test_dec14: In this test a private key that does not match with the encryption attribute, nevertheless it is a key that corresponds to an attribute present in the Universe, is passed to the decryption function. Therefore a successful test guarantees that an encrypted IMSI cannot be decrypted with any private key that is included in the crypto system but was not specifically chosen by the client device for encryption.

Note: these tests are the ones described in the Test Suite 5ge-52 Test 4 "Check the KPABE decryption function".

#:	Step actions:	Expected Results:
1	Run the 2 tests (steps) of the test suite 5ge-52.	The results are described in the 2 steps of test suite 5ge-52
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Feature-2.1.1: Encryption of Long Term Identifiers Use Case 2.2: Subscriber Identity Privacy	

T_UC2.2_2 Mobile user interception and information interception

Covered Threats	**T_UC2.2_1 and T_UC2.2_2**
Description: Detailed description of threat and its importance	In some procedures (e.g., initial network attach, paging requests, etc.) in all current mobile networks the IMSI is sent to the network in clear text. This opens the door to subscriber's identity interception/disclosure and unauthorized user tracking attacks. **EO interpretation of the threat**: Sending the subscribers' identifiers (IMSIs) in clear text over the air interface may allow the corresponding user information interception, therefore such identifiers must be concealed (e.g., through encryption).
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	User privacy violation through IMSI (International Mobile Subscriber Identity) interception and tracking.
Possible Mitigation Hints (optional, if foreseen): How can we protect against the threat?	If the 5G network is not able to protect end-user's privacy will be considered less trustworthy by the end-users. The threat mainly affects the AN (Access Network) domain.
Entry Points (optional, if known): What possible means does an adversary have?	Basic attack: passive sniffing of signaling traffic (in the specific test proposed for the feature developed in Release 1 this means the sniffing of EAP-AKA Identity Responses on the WiFi interface). Adversaries must link subscription identifiers to the users' identity. This can be achieved by triggering the mobile network into initiating the generation of paging messages to the victim (and thus to victim's terminal). For instance, adversaries may connect users with using social media application to initiate unobtrusive communications. Location tracking can be done at the granularity of base station's coverage or in more detail if the adversary has capabilities to analyse signal directions. Also, detailed location tracking is possible by eavesdropping plaintext signal measurement reports.
5G-ENSURE enablers (optional, if covered for given threat): What possible means does an adversary have?	Enabler: Privacy Enhanced Identity Protection (PEIP) Feature: Encryption of Long Term Identifiers

Please note that the tests that prove the coverage of threat T_UC2.2_2 also prove the coverage threat T_UC2.2_1 because the randomized encryption of IMSIs prevent both the mobile user interception and information interception and the user tracking.

Test Case 5ge-86: ProVerif privacy analysis of the group-based AKA protocol
Summary: Feature 1.1.1 is a group-based Authentication and Key Agreement (AKA) protocol. A machine-type communications (MTC) device using the protocol identifies itself by the combination of a group identifier, called GID, and a value that identifies the device within the group, called PATH. Since the long-term key K (stored in the UICC) is needed for a device to authenticate using the protocol, the device identifier (GID, PATH) is associated with an International Mobile Subscriber Identity (IMSI). However, in order to achieve MTC identity privacy, it is important that an adversary cannot identify the IMSI by observing a run of the group-based AKA protocol, even though the group-based AKA device identifier is sent in the clear. The following paper presents a ProVerif verification proving that the protocol meets this MTC

identity privacy property.

Giustolisi, R., Gehrmann, C., Ahlström, M. and Holmberg, S., 2017. A secure group-based AKA protocol for machine-type communications. In *International Conference on Information Security and Cryptology ICISC 2016: Information Security and Cryptology–ICISC 2016. 30 November 2016 through 2 December 2016* (pp. 3-27).

Execution type:	Manual
Estimated exec. duration (min):	
Priority:	Medium
Scenario evaluation score:	**1- Theoretical evidence**
Requirements	Feature-1.1.1: Group authentication by extending the LTE-AKA protocol Use Case 2.2: Subscriber Identity Privacy
Attached files	A secure group-based AKA protocol for machine-type communications : icisc_cameraready.pdf icisc_cameraready.pdf

Test Case 5ge-2: Subscriber's Identity Protection (Encryption) over the air interface

Summary:

The goal for this test is to ensure that at any time, the user long term identifier (IMSI) is protected when trasmitted over the air interface against eavesdropping (passive sniffing attacks). The test requires to capture all the traffic over the air interface during the attach of the mobile device to the wireless network, exmine all the messages that contain the IMSI and check that the IMSI is encryption. The encrypted IMSI should also be different from one attach to another in order to defend against user tracking.

Preconditions:

The Encryption of Long Term Identifiers feauture of the Privacy Enhanced Identity Protection enabler is installed

The 5ge-53 - "Test5 Check the KPABE encryption properties with Wireshark" test scenario is active with all the confguration settings on

In the basic attack scenario of test 5ge-53, verify that the feature provides different encryption outputs for the same input IMSI value in different attach procedures using EAP-AKA full authentication over an WiFi connection. The passive sniffer can be a PC connected to the same AP as the client (victim) and authentication server, that can intercept the signaling traffic on the wifi interface and can visualize the packets flow of the attach procedure with Wireshark. note that the encrypted IMSI is transferred inside the Identity Responses messages of EAP-AKA full authentication.

#:	Step actions:	Expected Results:
1	Prepare the environment for the test (this is described in details in the test case 5ge-53: Supplementary Test 5 Check the KPABE encryption properties with Wireshark): Ensure mobile device is not attached to the network Start the access point Start a network capture on the AP or on the mobile device	The mobile device is not attached to the network The access point is ready The capture is started
2	Start the attach procedure on the mobile device Check that a web page, ping .... can be launched from the mobile	The device is attached to the network and the access to the service requested works

	device	
3	Stop the test : Dettach the device Stop the network capture	The device is properly dettached from the network The capture is ready for analysis
4	Open the capture and check that in none of the signaling messages the IMSI is sent in clear text over the network	The IMSI is protected (cyphered) in all the messages
Execution type:	Manual	
Estimated exec. duration (min):	15.00	
Priority:	High	
Scenario evaluation score:	**4 - Testbed evaluation (real flows)**	
Relations	related to - 5ge-53:Supplementary Test 5 Check the KPABE encryption properties with Wireshark – Different ci depends on - 5ge-53:Supplementary Test 5 Check the KPABE encryption properties with Wireshark – Different ci related to - 5ge-62:Subscriber's Identity Protection, through Encryption, over the air interface	
Requirements	Feature-2.1.1: Encryption of Long Term Identifiers Use Case 2.2: Subscriber Identity Privacy	
Attached files	IMSIEncryption_WP2-WP4_threat_coverage.docx	

T_UC2.1_2 Tracking of device's (user's) location

Test Case 5ge-75: Device Identity Privacy Evaluation
Summary: This evaluation test should demonstrate that the DIP enabler DNA privacy enhancement features (Dummy address injection and Random ordering) provide for improvement of path location privacy.
Preconditions: The DIP integration and evaluation packages have been installed: dip-integrate-scripts_1.0_amd64.deb, dip-eval-scripts_1.0_amd64.deb, dhcpcd-dip_1.1_amd64.deb The DIP integration configuration script needs have been run once. All commands need to run as root (e.g. start with sudo -s) Copy the eval config files to the system the run the following commands (only needed once): # cd /opt/dip/dip-eval/ # cp dhcpcd-conf/* /etc/ The instructions for use are in the /opt/dip/dip-eval/README New dependencies: python-scapy [installed]

To start the 4 simulated Access Points - four instances of hostapd are started on interface wlan1-wlan4, after which the ISC DHCP server is restarted to listen on the APs. # ./dip-eval-integrate-init.sh		
#:	Step actions:	Expected Results:
1	The following eval script will instantiate an LXC unshare container shell attached only to wlan0 and start an associated wpa_supplicant daemon. Then the script will run the evaluation tests: # ./dip-eval-test.sh	The result of the test should indicate that the enabler DNA privacy enhancement features (Dummy address injection and Random ordering) provide privacy improvement - the simulation ends with Analysis phase indicating the results (the improvement ratios should be greater than zero). e.g.: ========== Analysis phase: ========== - Processing and filtering packet captures Results: [Privacy ratio (0.0 : No privacy, 1.0 : Anonymous) as compared to standard DNA]: => DNA Random privacy improvement ratio: 0.333333333333 => DNA Dummy privacy improvement ratio: 0.142857142857 Evaluation Completed.
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Use Case 2.1: Device Identity Privacy Feature-2.2.1: Enhanced privacy for network attachment protocols	

Use Cases cluster 3 - IoT Device Authentication and Key Management

T_UC3.1_1 Authentication traffic spikes

Description: Detailed description of threat and its importance	Simultaneous or periodic authentication events may cause excessive amount of traffic for network. Adversaries – aiming to perform a denial-of-service attack - may try to initiate traffic spikes or emphasize the effects of natural traffic spikes with IoT application specific means. As a consequence, the network will experience more signalling and authentication functions needs to perform more processing. Potentially, the authentication of devices may fail and devices may lose connectivity.

Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	The 5G network must be over-resourced in order to handle large short-term traffic amounts.
Possible Mitigation Hints (if known): How can we protect against the threat?	Different means may be utilized to mitigate traffic spikes. Methods include relying gateway or one group member to perform authentication on the behalf of individual devices. For instance, using group authentication schemes such as [3]. Monitoring and filtering approaches can be used to mitigate effects.
Entry Points (if known): What possible means does an adversary have?	The traffic spikes may emerge naturally in the IoT network as devices may be programmed e.g. to join the network at the same time. However, an adversary may try to guide this behaviour with different means, for instance, by tampering network time or causing power outages to get large amount devices to authenticate at the same time.

Test Case 5ge-82: Monitor and control number of active UEs

Summary:

Authentication traffic spikes in a network system may arise from initialization of massive IoT systems or they can indicate a hostile denial of service attack.

The objective of this test is to show that the Trust Metric Enabler can detect and control authentication traffic spikes. The enabler monitors number of active UEs which an eNodeB controls and if they reach a pre-defined limit, the enabler indicates that the monitored network section is not trusted.

Limitations: Since the Release 1 version of Trust Metric Enabler gets its input from disk files, it cannot detect traffic spikes in real time mode. This will change in Release 2 version which can monitor NFVs' counters and KPIs.

NOTE: This test is similar to Unit Test 2 (5ge-35). It is also identical to threat coverage test 5ge-83. Executing one of these tests demonstrates that they all work.

Preconditions:

Trust Metric Enabler and its two input files (number_of_devices.csv and trust_requirements.csv) are implemented. Unit Test 1 is successfully done.

#:	Step actions:	Expected Results:
1	Do not edit the contents of the input file "trust_requirements.csv", keep it as it is after Unit Test 1. Edit the input file "number_of_devices.csv" by adding more rows to it. The file "number_of_devices.csv"should look as follows: 1 7 3 8 2 88 7 12 19 77	

	1 2	
2	Execute Python script "Trust_Metric_Enabler.py"	The output file "Trust_Metric_Log.csv" should contain character string "False". This indicates that the network is not trusted
Execution type:	Manual	
Estimated exec. duration (min):	0.10	
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Relations	related to - 5ge-83:Measure number of active UEs to evaluate trust related to - 5ge-35:Unit Test 2: No trust, too many devices	
Requirements	Feature-3.2.1: Trust metric based network domain security policy management	

Test Case 5ge-84: Bandwidth consumption of the group-based AKA protocol

Summary:

Checks that when 100 MTC devices attach to the network almost simultaneously, the bandwidth consumption counted as the sum of the bandwidth consumption between the MTC devices and the MME (NAS) and the bandwidth consumption between the MME and the HSS (S6A) is less for the group-based AKA protocol than for EPS-AKA. The number of bytes is counted by using a packet analyzer, on the test bed for the group-based AKA and outside of the test bed for EPS-AKA. Only the bytes of the parameters relevant to the group-based AKA protocol and their counterparts in EPS-AKA are counted.

Preconditions:

The group-based AKA enabler feature is installed.

#:	Step actions:	Expected Results:
1	By using tcpdump, capture the data sent between the MME and the UE and between the MME and the HSS during a Case A run of the enabler feature. To run the enabler do the following steps: On the HSS virtual machine: Execute: usr/local/etc/oai/SCRIPTS/run_hss On the MME virtual machine: Execute: “/usr/local/etc/oai/SCRIPTS/run_mme -i” On the oaisim virtual machine: cd /usr/local/etc/oai/cmake_targets/tools	A file with the captured data, let us call it caseA.pcap.

	sudo -E ./run_enb_ue_virt_s1 On the epc virtual machine: cd /usr/local/etc/oai/SCRIPTS/ execute: ./run_spgw Refer to tests 5ge-{102,103,104,106} for preconditions.	
2	Open caseA.pcap in Wireshark. For example by applying the filter S1AP, find the message Attach Request in the “Packet List” pane and select it. In the “Packet Details” pane, expand nodes in the following order: “S1 Application Protocol”, “S1AP-PDU: initiating message”, “initiatingMessage”, “value”, “InitialUEMessage”, “protocolIEs: 5 items”, “Item 1: id-NAS-PDU”, “value”, “Non-Access-Stratum (NAS)PDU”. Select the node “EPS mobile identity” and count the number of bytes that are highlighted in the “Packet Bytes” pane. Select the node “Extraneous Data” and count the number of bytes this consists of. “EPS mobile identity” contains the GID and “Extraneous Data” contains PATH and NONCE. Add the two numbers together.	The sum is 28.
3	For example by applying the filter “diameter” or the filter “sctp.port == 3868”, find the message Authentication Information Request and select it. In the “Packet Details” pane, expand “Diameter Protocol” and then select “AVP: Unknown(5001) l=14 f=VM- vnd=TGPP val=0504”. In the “Packet Bytes” pane, count the number of bytes that are highlighted. These are the bytes used for the attribute-value pair that contains PATH. Select the User-Name AVP in the “Packet Details” pane. Count the number of bytes highlighted. Select the AVP Visited-PLMN-Id. Count the number of bytes highlighted. Add the three numbers together.	The number for PATH is 16, the number for User-Name (i.e., the group identifier) is 24, the number for Visited-PLMN-Id is 16. So the sum is 56.
4	Select the message Authentication Information Answer. In the “Packet Details” pane, expand “Diameter Protocol”. Count the bytes of the AVPs with the following codes (they are not grouped in any other AVP): 5001 (PATH), 5003 (GKij), 5002 (CHij), 5004 (TREE HEIGHT), 5007 (NODE DEPTH). Count the bytes of the User-Name AVP that is not grouped inside another AVP (this is the IMSI). Expand the nodes “AVP: Authentication-Info”, “Authentication-Info”, “AVP: E-UTRAN-Vector”, and “E-UTRAN-Vector”. Select the node “E-UTRAN-Vector”. Count the number of bytes that are highlighted. Add together all the numbers obtained in this step.	PATH uses 16 bytes. GK uses 28 bytes, CH uses 28 bytes, TREE HEIGHT uses 16 bytes, NODE DEPTH uses 16 bytes, IMSI uses 24 bytes and the E-UTRAN-Vector AVPs uses 144 bytes. So the sum is 262 bytes.
5	For example by applying the filter “S1AP”, find the message Authentication Request and select it.	The number of bytes is 33.

	Expand “S1 Application Protocol”, “S1AP-PDU: initiatingMessage (0)”, “initiatingMessage”, “value”, “DownlinkNASTransport”, “protocolIEs: 3 items”, “Item 2: id-NAS-PDU”, “ProtocolIE-Field”, “value” and “Non-Access-Stratum (NAS)PDU”. Count the bytes used for the nodes “Authentication Parameter RAND - EPS challenge” and “Authentication Parameter AUTN (UMTS and EPS authentication challenge) - EPS challenge”.	
6	Find the message Authentication Response and select it. Count the number of bytes used for the “Authentication response parameter” (including the length field).	The number is 9.
7	By using tcpdump, capture the data sent between the MME and the UE during a Case B run of the enabler feature.	A file with the captured data, let us call it caseB.pcap.
8	Open caseB.pcap in Wireshark. Apply the filter “S1AP”. Select the packet sent after the attach request and before the authentication response. This should be the message “Authentication Request Derivable” introduced by the group-based AKA protocol although it cannot be identified as such by Wireshark. In the “Packet Details” pane, expand nodes in the following order: “S1 Application Protocol”, “S1AP-PDU: initiating message”, “initiatingMessage”, “value”, “DownlinkNASTransport”, “protocolIEs: 3 items”, “Item 2: id-NAS-PDU”, “ProtocolIE-Field”, “Non-Access-Stratum (NAS)PDU”. Select “Non-Access-Stratum (NAS)PDU” and count the number of bytes highlighted in the “Packet Bytes” pane. Subtract three from this number as the first three bytes are used for information elements that are included also in the Authentication Request message in EPS-AKA. The result is equal to the number of bytes that are used for CHij and AUTd.	The number of bytes is 31.
9	Select the message “Authentication response”. Count the number of bytes used for the “Authentication response parameter”.	The number is 17.
10	Add the numbers for Case A together.	The sum is 388.
11	A Case B run starts with exactly the same message being sent as in Case B, e.i., the Attach Request, for which we obtained the number 28. Add the numbers for Case B together.	The sum is 76.
12	When multiple members of the same group attach with the same MME, a Case A run is run for the first device and Case B runs are run for the remaining devices (assuming the HSS sends nodes that are high up in the trees). Calculate the number of bytes used when 100 devices in the same group attach with the group-based AKA protocol with the same MME.	The number is 7912. (388 + 99 * 76 = 7912).
13	The enabler owner has counted the corresponding bytes of an EPS-AKA protocol run using the same method.	The number is 211.

	The numbers are 9 for Attach Request, 40 for Authentication Information Request, 120 for Authentication Information Answer, 33 for Authentication Request and 9 for Authentication Information Answer. Add the numbers together.	
14	Calculate the number of bytes used when 100 devices in the same group attach with EPS-AKA.	The number is 21,100.
15	Compare the number of bytes consumed if the group-based AKA protocol is used or if EPS-AKA is used.	Less bandwidth is consumed with the group-based AKA protocol.
Execution type:	Manual	
Estimated exec. duration (min):	90.00	
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Feature-1.1.1: Group authentication by extending the LTE-AKA protocol Use Case 3.1: Authentication of IoT Devices in 5G	

Test Case 5ge-85: ProVerif security and privacy analysis of the group-based AKA protocol	
Summary: An authentication scheme for IoT devices that aims to mitigate the authentication traffic spikes threat must still provide adequate security and privacy, otherwise the effect could be that an adversary can break authentication, derive a session master key or compromise the privacy. In the paper referenced below a ProVerif analysis of the group-based AKA protocol (feature 1.1.1) is presented. It is proven that the protocol meets mutual authentication, key confidentiality and device identity privacy. Giustolisi, R., Gehrmann, C., Ahlström, M. and Holmberg, S., 2017. A secure group-based AKA protocol for machine-type communications. In *International Conference on Information Security and Cryptology ICISC 2016: Information Security and Cryptology–ICISC 2016. 30 November 2016 through 2 December 2016* (pp. 3-27).	
Execution type:	Manual
Estimated exec. duration (min):	
Priority:	Medium
Scenario evaluation score:	**1- Theoretical evidence**
Requirements	Feature-1.1.1: Group authentication by extending the LTE-AKA protocol Use Case 3.1: Authentication of IoT Devices in 5G
Attached files	A secure group-based AKA protocol for machine-type communications : icisc_cameraready.pdf

T_UC3.2_1 Leaking keys

Description: Detailed description of threat and its importance	End-to-end keys may be stolen or leak from the centralized key servers. The key server may also become tampered. As a consequence, the end-to-end secured communication is vulnerable for different attacks and adversaries gain an access to the end-points. The may e.g. provide false information to application services or send malicious commands to IoT devices.
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	The leaking keys will compromise the security (confidentiality and integrity) of those applications that are end-to-end secured.
Possible Mitigation Hints (if known): How can we protect against the threat?	The key server could be used only for authentication purposes and not for delivering the sessions keys. This would make attacks more difficult as the attacker would be required to compromise the server to provide wrong (asymmetric) authentication keys and then mount an interception attack on the end-to-end communication. However, all IoT devices may not be computationally capable to asymmetric key operations. The key server should be hardened to withstand attacks. The server cannot be isolated from the open internet as it needs to be available for the clients. However, some isolation techniques – e.g. micro-segmentation – may be utilized to control which applications may access the server.
Entry Points (if known): What possible means does an adversary have?	Attacker may compromise the key server in various ways. For instance, the attacker may utilize vulnerabilities in server interfaces to gain an access to the service. Lawful interception mechanisms may be vulnerable and leak keys for third-party attackers or authorities that are misusing their privileges.

Test Case 5ge-94: No key in plain-text

Summary:

The private key required for accessing the controller should never be available in clear text on the system. This prevents the key from being leaked to an adversary.

Preconditions:

The verification manager software is installed on VM1. The remote host software is installed on VM2. The application is also installed on host VM2.

The following files should contain on each VM the IP of the other VM host. For instance for VM1:

/opt/bootstrappingtrust/Certs/rh_host (should have IP address of VM2)
and
/opt/bootstrappingtrust/Certs/container_host (should have IP address of VM2)

#:	Step actions:	Expected Results:
1	Launch the remote host software on VM2. cd /opt/bootstrappingtrust/RemoteHost sudo ./app	It is launched and awaits connections.

2	Launch the benign application on VM2. cd /opt/bootstrappingtrust/Application sudo ./app	The application starts, and awaits connections from the verification manager.
3	Launch the Verification manager on VM1, which wil then immediately try to connect and try to attest the integrity of the application. cd /opt/bootstrappingtrust/VerificationManager ./app	The verification manager will provision the enclave with a key. No lines in the output of the verification manager will have the text "ERROR". Switching to the output from VM2 and the application, the following output should be visible near the end, since this indicates that a key was provisioned (although we can't connect since the SDN controller is not running) . Seeding the random number generator... ok . Loading the CA root certificate ...OK ok (0 skipped) . Connecting to tcp/localhost/8081... failed ! mbedtls_net_connect returned -68
4	Search for a private key on the application host (VM2). All generated private keys will have a header containing the words BEGIN PRIVATE KEY, so this string can be searched for. sudo grep -H -r -I "BEGIN PRIVATE KEY" /opt/bootstrappingtrust	Except for source code matches in mbedtls-2.2.1 directories, and two keys related to the test-bed, ./Certs/ca/ca_key ./Certs/server_app.key , no matches should be found. If the statement above holds, then this test is successful.
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Use Case 3.2: Network-Based Key Management for End-to-End Security Feature-5.3.1: Integrity Attestation of Virtual Network Components	

Use Cases cluster 5 - Software-Defined Networks, Virtualization and Monitor

T_UC5.1_1 Misbehaving control plane

Description: Detailed description of threat and its importance	Malicious or compromised control plane may jeopardize the network and the data plane. For instance, a compromised SDN controller or virtualization orchestrator may prevent data flows or direct them to a man-in-the-middle switch for eavesdropping or tampering. Centralized network controllers are an alluring targets for attacks as adversaries are not required to compromise switches or network functions it is enough that they steer data flows to their own malicious components.
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	The network and applications become vulnerable to eavesdropping and tampering as well as denial-of-service attacks.
Possible Mitigation Hints (if known): How can we protect against the threat?	Strong protection should be provided for control plane components. They should authenticate and authorize commands and support up-to-date trusted interfaces.
Entry Points (if known): What possible means does an adversary have?	To compromise control plane: Adversaries may send malicious commands / policies to the controller, if controller does not strongly authenticate and authorize the source of the policies. As a consequence, a legitimate controller will behave maliciously according to adversaries' policies. Alternatively, adversaries may compromise legitimate control plane component, for instance, by utilizing weaknesses in the controller and its interfaces. Adversaries may also get credentials to provide the controller policies using e.g. social engineering attacks against the operator. A data plane may be misconfigured so that it accepts control commands also from other slices or external parties. If data plane does not authenticate commands from the controllers, an adversary may masquerade as legitimate control plane component and send malicious southbound control messages.

Test Case 5ge-76: Monitoring of controllers' interfaces

Summary:

Misbehaving control plane (e.g. a compromised SDN controller) can be detected e.g. by monitoring interfaces to SDN controller in order to detect control channels. Incoming or outgoing communication between remote adversary - which is known to be untrusted or which is unknow - reveals potential attacks.

The objective of this test case is to show that the ***security monitor for 5G micro-segment***s enabler is able to monitor and collect network statistics from interfaces that connect SDN control plane to remote hosts. The monitoring enabler is also able to distribute this information to centralized operating device and show it to human administrator.

Limitations: release 1 of the enabler has not been integrated to the SDN network, hence, we are only monitoring coummunication of a host that could act as an SDN controller. Also, Release 1 does not yet provide any automated threat inferencing or analysis over collected statistics. Here we assume that the administrator is able to manually recognize connections to malicious remote targets.

Note, the implementation of the test is identical to enabler's unit test 2 (5ge-33) and threat coverage tests (5ge-77, 5ge-78). It is not necessary to execute it twice to demonstrate it works.

Preconditions:

The microsegment monitoring enabler has been deployed.

#:	Step actions:	Expected Results:

1	Starting the monitoring framework (the kafka broker and Zookeeper):
	
	$ cd /usr/local/src/kafka_2.11-0.10.1.0/
	$ bin/zookeeper-server-start.sh config/zookeeper.properties
	$ bin/kafka-server-start.sh config/server.properties	

3	The monitoring probe is deployed to the SDN controller. Here we assume that it is 'the localhost' i.e. all components are running in the same host. Execute libpcap-based pmaccd daemon (pmacctd) is using the default configuration distributed with the enabler: $ sudo pmacctd -f /usr/share/doc/microsegmentmonitor/tokafka.conf &	The pmacctd will start publishing network statistics information through pmacct.acct topic in the Kafka broker. The statistics are in JSON format and look something like this: {"port_src": 48922, "ip_dst": "109.105.109.212", "ip_src": "10.0.2.15", "port_dst": 443, "ip_proto": "tcp", "stamp_updated": "2016-06-16 08:37:31", "stamp_inserted": "2016-06-16 08:35:00", "packets": 5, "bytes": 323}. Where: port_src - source port ip_src - source IP address ip_dst - target's IP address ip_proto - protocol stamp_inserted, stamp_updated - timestamps packets - amount of packets transmitted during the monitoring period bytes - amount of bytes transmitted during the monitoring period

4	Subscribe and output network statistics with the monitoring application
	
	$ cd /usr/local/src/spark-1.6.2-bin-hadoop2.6/
	$ bin/spark-submit --packages org.apache.spark:spark-streaming-kafka_2.10:1.6.1 examples/src/main/python/streaming/direct_kafka_wordcount.py localhost:9092 pmacct.acct | grep 192.237.223.114 -B 3 -A 1
		The network statistics should be found from the output of the application. Note that the output may not be easily found as the example application outputs also lots of other information. Hence, we used grep to highlight some relevant parts that will be shown only after the final step.
		
		In this test, we assume that the IP address of malicious server is known. These addresses are learnt e.g. by analyzing to whom the maliciously behaving node or infected sw communicates with. Release 1 does not provide any functionality to infer this. Hence here we grepped 192.237.223.114 (www.5gensure.com)	

5	Create simulated adversarial control data. Here we assume that the attacker uses ping protocol to communicate: $ ping www.5gensure.eu	The connection to remote site is visible from the monitoring applications output: (u' "192.237.223.115",', 1)	
Execution type:	Manual		
Estimated exec. duration (min):			
Priority:	Medium		
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**		
Relations	related to - 5ge-33:Unit test 2 - Pmacct, Kafka, Spark related to - 5ge-77:Monitoring use of control interfaces and effects of misuse related to - 5ge-78:Monitoring adversarial data flows in network slices		
Requirements	Feature-4.4.1: Complex Event Processing Framework for Security Monitoring and Inferencing		

	Use Case 5.5: Control and Monitoring of Slice by Service Provider

Test Case 5ge-81: Prevention of non-policy compliant data plane reconfigurations

Summary:

Network administrators consider miscongurations to be the main source for network failures. In turn, a misconguration usually happens because the complexity of modern networks must still be mastered to a large extent with error-prone low-level manual interventions. Software-defined networking (SDN) provides the means to simplify and automate network operations. The SDN architecture comprises a logically centralized controller that provides a programmable logic to operate the network, abstractions, and services such as topology discovery or end-to-end connectivity. The services are accessed through so-called north-bound interfaces (NBIs), which can be used manually by network administrators or by network applications that run on top of the controller and automate network operation tasks.

Consider the following scenario, with a simple SDN nework consisting one switch and three hosts h1, h2, and h3 connected to it. We assume the following policy: connectivity should be provided between h1 and h2; h3 can obtain connectivity to h1 only after the network administrator grants it. Moreover, assume that the SDN controller ONOS controls the network with two applications: (1) a forwarding application, which establishes connectivity between h1 and h2, and (2) a command-line interface (CLI) to submit connectivity intents.

For our network, the network administrator may use the CLI to provide connectivity to h3. However, since the CLI can be used to provide any type of connectivity intent, the CLI can also be used to redirect the traffic between h1 and h2 to h3, which should not be permitted. Hence, the network administrator can intentionally or unintentionally easily misconfigure the data plane. Furthermore, note that such kind of misconfigurations can also originate from a bug in a network application or in the SDN controller.

See also the following demo for further details.

D. Gkounis, F. Klaedtke, R. Bifulco, G.O. Karame.

Cases for Including a Reference Monitor to SDN.

SIGCOMM 2016.

Preconditions:

- The enabler "Access Control Mechanisms" has been deployed (which includes the ONOS controller)
- Network of physical switches and hosts are in place.
- The physical network is configured with the proper topology (according to attached image).
- The policy file is put in ***/etc/onos/refmon*** and properly setup with the correct MAC addresses of the machines included in the topology. An example policy file is provided in the attachment *policy.txt*.

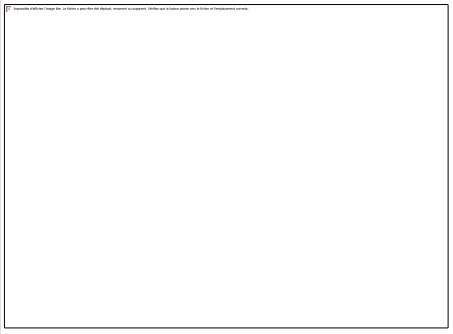
Note that we assume that the controller (ONOS) at the network's control plane is managing the switches at the network's data plane via the OpenFlow protocol. Furthermore, the switches will contact the contoller when receving network packets that do not match any of their currently installed flow rules.

Note on the evaluation score:

* We have not chosen 3, since we are not using Mininet for simulating a network. However, we are using a software switch and not hardware switches. because of the SIGCOMM demonstration,

* We have not chosen 5, although the enabler has been demonstrated at SIGCOMM 2016 with an accompanying short paper. However, this short paper is not a full fledged analysis of the enabler.

#:	Step actions:	Expected Results:
1	Start ONOS:	ONOS loads correctly into its

	1. Start ONOS daemon with: *sudo service onos start* 2. The status of the ONOS daemon can be checked with: *sudo service onos status* 3. Wait for ONOS to startup. This takes about 30 seconds. To monitor ONOS startup process, it is possible to check its log with: *tail -f /opt/onos/log/karaf.log*. The startup process should be complete when no new entries are being added to the log. 4. Start ONOS console with: */opt/onos/bin/onos*	console.
2	- Open a Terminal in Host 1 - Ping Host 2 by executing ***ping -c5 <Host2_IP_Address>***	The ping should work.
3	- Open a Terminal in Host 1 - Ping Host 3 by executing ***ping -c5 <Host3_IP_Address>***	The ping should not work.
5	Inside ONOS console, execute ***print-refmon-log***	- Several "POLICY VIOLATION" should have been logged by ONOS. - These are caused by Host 1 trying to ping Host 3, which violates the policies listed in the policy file.
Execution type:	Manual	
Estimated exec. duration (min):	30.00	
Priority:	Medium	
Scenario evaluation score:	**4 - Testbed evaluation (real flows)**	
Requirements	Feature-5.1.1: Southbound Reference Monitor Use Case 5.2: Adding a 5G Node to a Virtualized Core Network	
Attached files	policy : policy.txt topology : net_topology.png  SIGCOMM : sigcomm16posters-final62.pdf	

Test Case 5ge-138: Reactive adding of flow rules in SDN networks
Summary:

In this scenario, the compliance checker is used to check a simple policy about the interactions between the SDN controller and SDN switches. Namely, whenever a switch receives a network packet with no matching flow rule, the controller must reconfigure the switch accordingly, within a time bound. In other words, the compliance checker checks that the controller timely reacts to packet-in OpenFlow messages by corresponding flow-mod OpenFlow messages.

For the moment, we restrict ourselves to this simple policy. Other,more complex, policies about the interactions via OpenFlow messages between the control plane and the data plane can be checked accordingly. An example is that barrier requests are handled appropriately. However, the setup will be more involved and we want to keep things simple here.

In the following, we describe how to configure, setup, and run the different involved components, namely, runverif, OVS, ONOS, and Mininet.

Preconditions:

RUNVERIF

The Debian package runverif-20170912_1-amd64.deb must be installed.

Installing this package places the command-line tool runverif in the /bin directory. The tool's version should be 1.0.16 or higher (for a check, use the command-line argument --version).

OVS

OVS needs to be instrumented and configured to report OpenFlow messages to runverif. For this, the provided Debian package openvswitch-switch_2.5.0-1_amd64.deb, which contains a modified version of the OVS daemon ovs-switchd, needs to be installed in addition to the standard Debian packages for OVS. Note that that the standard Debian packages for OVS should be installed prior to installing the modified daemon. You may need to use the command-line argument --force-overwrite for installing the provided Debian package.

Note that the version 2.5.0 of the OVS daemon was modified. You can check the version of the OVS daemon with ovs-vswitchd --version. The output should contain information about runverif.

ONOS

ONOS needs to be instrumented and configured to report OpenFlow messages to runverif. For this, the provided Debian package onos-5gebuild_1.0-1.deb needs to be installed. ONOS is installed into the /opt directory. Furthermore, Java 8 needs to be installed.

MININET

The standard Debian package for Mininet needs to be installed.

#:	Step actions:	Expected Results:
1	RUNVERIF Open a new terminal. In the following, we refer with RV (RunVerif) to this terminal. Start runverif from the RV terminal:	

	RV> runverif --prefix=flowmod We assume here that the current directory contains the three configuration files flowmod.comp, flowmod.msgs, and flowmod.spec. runverif opens the UDP port 50010 and listens on it. Incoming messages are processed by runverif. With RV> runverif --prefix=flowmod --loglevel=3 runverif outputs additional information: [I] 2017/09/13 13:35:00.732019 1 out of 1 CPU core is used. [I] 2017/09/13 13:35:00.736009 Garbage collection target percentage is set to 100. [I] 2017/09/13 13:35:00.750538 2 components are monitored: ovs-vswitchd_s1, onos-out [I] 2017/09/13 13:35:00.804589 Verdicts are with respect to the specification: (FREEZE inport[inport], src[src], dst[dst]. ((NOT SwitchPacketIn(inport, src, dst)) OR (TRUE UNTIL(0s,100ms] ControllerFlowAdd(inport, src, dst)))) [I] 2017/09/13 13:35:00.804651 Monitoring algorithm: mtldata [I] 2017/09/13 13:35:00.831038 UDP socket (port: 50010) is open. When setting the loglevel to 7, runverif also outputs the received messages. Note that runverif's output is also logged in the file /tmp/runverif.log. Note that without the command-line argument --violations, verdicts TRUE are suppressed. A propositional version of the policy is also available: RV> runverif --prefix=flowmod-prop --monitor=mtl The corresponding configuration files are flowmod-prop.comp, flowmod-prop.msgs, and flowmod-prop.spec. Note that here the monitoring algorithm 'mtl' is used, which has a higher throughput. However, the policy is less precise.	
2	OVS The modified OVS daemon is set up as follows by editing the ovs-ctl script in the directory /usr/share/openvswitch/scripts/. 1. Open a new terminal. In the following, we refer with OVS to this terminal.	

	2. Stop the OVS daemon: OVS> sudo /usr/share/openvswitch/scripts/ovs-ctl stop 3. Modify the script ovs-ctl. Go to the function start_forwarding () and change the line set "$@" --runverif into set "$@" --runverif --runverif-ofp=3 The command-line arguments have the following meaning. --runverif enable the sending of runverif messages --runferiv-ofp=9 report OFPT_PACKET_IN messages (--runverif-globalcounter check if this argument is necessary) For debugging, use the additional command-line argument: --runverif-log It also may be necessary to change some of the following default values. --runverif-host=127.0.0.1 --runverif-port=50010 --runverif-prefix=ovs-switchd 4. Start the OVS daemon: OVS> sudo /usr/share/openvswitch/scripts/ovs-ctl start The log file /var/log/openvswitch/ovs-vswitchd.log for the OVS daemon should contain the information that the connection to runverif is established. 2017-09-13T13:08:01.569Z|00004|runverif|INFO|Opening the monitoring socket (host: 127.0.0.1, port: 50010). 2017-09-13T13:08:01.569Z|00005|runverif|INFO|Connection to monitor established.	
3	ONOS Set up the modified version of ONOS as follows. 1. Open a new terminal. In the following, we refer with ONOS to this terminal.	

	2. To configure ONOS to report OpenFlow messages to runverif, edit the file /opt/onos/apache-karaf-3.0.8/bin/setenv. Set the environment variable ONOS_RUNVERIF_IN to false, ONOS_RUNVERIF_OUT to true, and ONOS_RUNVERIF_MSGS to FLOW_MOD. Further environment variables are ONOS_RUNVERIF_HOST and ONOS_RUNVERIF_PORT. Their default values are 127.0.0.1 and 50010, respectively. 3. Start ONOS: ONOS> /opt/onos/bin/run-onos.sh A new terminal should open. For simplicity, we also refer with ONOS to this terminal. You first need to enter the password for obtaining superuser privileges. ONOS will then start. This may take some while. 4. Make sure that the apps org.onosproject.fwd and org.onosproject.openflow are running: ONOS> apps -s -a should show * 20 org.onosproject.optical-model 1.11.0.SNAPSHOT Optical information model * 45 org.onosproject.drivers 1.11.0.SNAPSHOT Default device drivers * 68 org.onosproject.hostprovider 1.11.0.SNAPSHOT Host Location Provider * 69 org.onosproject.openflow-base 1.11.0.SNAPSHOT OpenFlow Provider * 74 org.onosproject.lldpprovider 1.11.0.SNAPSHOT LLDP Link Provider * 75 org.onosproject.openflow 1.11.0.SNAPSHOT OpenFlow Meta App * 85 org.onosproject.fwd 1.11.0.SNAPSHOT Reactive Forwarding App In case the apps do not appear in the list, start them manually: ONOS> app activate org.onosproject.fwd ONOS> app activate org.onosproject.openflow	

4	MININET Open a new terminal. In the following, we refer with MN (MiniNet) to this terminal. Start mininet in the MN terminal: MN> sudo mn --mac --topo single,3 --switch ovs,protocols=OpenFlow10 --controller=remote,ip=127.0.0.1,port=6633 The network comprises the following components: * A controller at the IP address 127.0.0.1:6633. * A single switch that uses the OpenFlow protocol version 1.0. * Three hosts h1, h2, and h3 that are conntected to the switch.	
5	TESTS Now, all components (runverif, OVS, ONOS, and mininet) are running. In particular, when running runverif without the flag --violations, verdicts may appear from time to time on the RV terminal. The reasons are as follows. First, ONOS sends network packets for the network discovery. The switch reacts to these packets by packet-in OpenFlow messages. Runverif is notified by these messages. Second, ONOS adds default flow rules to the switch's flow table. ONOS notifies runverif about the corrsponding flow-mod OpenFlow messages. None of these OpenFlow messages should result in verdicts FALSE. When pinging in Mininet a host then this should not result in any policy violation. For example, with MN> h1 ping -c10 h2 the host h1 pings the host h2 for 10 times. Some of the ICMP (and also ARP) network packets, which are received by the switch, will result in packet-in OpenFlow messages that are sent from the switch to the controller. First, for each of the resulting packet-in OpenFlow messages, ONOS should request the installation of corresponding flow rules by sending flow-mod OpenFlow messages to the switch. Second, this should happen within the specified time bound of 100ms. Hence, in the RV terminal, one should only see verdicts TRUE. Note that a buggy reactive forwarding application may request the installation of a flow rule that does not correspond to a packet-in	Output in RV terminal: [V] @1505461416.973250000: true [V] @1505461417.047073000: true [V] @1505461417.047516000: true [V] @1505461417.047618000: true [V] @1505461417.048719000: true [V] @1505461440.296777000: true [V] @1505461440.320839000: true [V] @1505461462.549872000: true [V] @1505461462.575493000: true [V] @1505461529.682289000: true [V] @1505461529.683793000: true [V] @1505461529.683209000: true [V] @1505461529.683557000: true [V] @1505461529.684102000: true Note that the timestamps (in Unix time) will differ. They should be the current time of the test. Furthermore, the number might vary. Finally, when running the test with the changed policy (i.e. a time window of 10ms), there should be FALSE verdicts, when pinging the host in Mininet.

	OpenFlow message. In this case, runverif would detect the policy violation and output the verdict FALSE. Furthermore, note that when changing the policy (flowmod.spec) so that ONOS needs to react within a much shorter time window, say within 10ms, runverif will most likely output verdicts FALSE, since ONOS' reaction time is not anymore within the specified time window. Rerun the test, by restarting all components with the changed policy in flowmod.spec. We remark that for simplicity, we require that ONOS only needs to react to packet-in OpenFlow messages resulting from ICMP network packets, as they are sent by the ping command. Furthermore, these network packets are either of type 0 (HELLO) or type 8 (REPLY). For other network traffic, e.g., HTTPS, one needs to modify the runverif configuration file flowmod.msgs. Recall that this is a very basic scenario with a simple policy to illustrate the concept. Other policies can be checked by modifying the setup and runverif's configuration files.	
Execution type:	Manual	
Estimated exec. duration (min):	45.00	
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Feature-5.2.1: Basic OpenFlow Compliance Checker Use Case 5.3: Reactive Traffic Routing in a Virtualized Core Network Use Case 5.4: Verification of the Virtualized Node and the Virtualization Platform	
Attached files	flowmod-prop.spec README-flowmod flowmod-prop.msgs flowmod-prop.comp flowmod.spec flowmod.msgs flowmod.comp	

T_UC5.2_1 Add malicious nodes into core network

Description: Detailed description of threat and its importance	Malicious nodes may e.g. eavesdrop, tamper, and prevent data flows.
Category: ITU-T X.805 security dimension(s)	Access control; Authentication; Non-repudiation; Data confidentiality; Communication security; Data integrity; Availability; Privacy
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	Confidentiality, integrity and availability of e2e communication are compromised.
Possible Mitigation Hints (if known): How can we protect against the threat?	Applying security verification procedures – technical and organisational - for assuring that the added nodes are trustworthy. Only authenticated and authorized entities should be allowed to add nodes. Security monitoring of behaviour of added nodes as well as communication over the network.
Entry Points (if known): What possible means does an adversary have?	Software, image used for deploying new nodes may be compromised. Forwarding logic may be misconfigured so that illegitimate node, switch is able to get access to data flows. In this case, the malicious node is unintentionally added to the core network.

Test Case 5ge-93: Malicious enclave don't get key
Summary: A malicious or compromised enclave should not be added to the network. This means that if the actual measurement of the application is not in the list of expected hashes, the application should not be provisioned with a key and the network can thus not connect to the SDN controller.
Preconditions: The verification manager software is installed on VM1. The remote host software is installed on VM2. The (malicious) application (ApplicationEvil) is also installed on host VM2. The following files should contain on each VM the IP of the other VM host. For instance for VM1: /opt/bootstrappingtrust/Certs/rh_host (should have IP address of VM2) and /opt/bootstrappingtrust/Certs/container_host (should have IP address of VM2)

#:	Step actions:	Expected Results:
1	Launch the remote host software on VM2. cd /opt/bootstrappingtrust/RemoteHost sudo ./app	It is launched and awaits connections.
2	Launch the malicious application in a new window on VM2. cd /opt/bootstrappingtrust/ApplicationEvil sudo ./app	It is launched and awaits connections.
3	Launch the verification manager on VM1, which will then immediately try to connect and try to attest the integrity of the (malicious) application. cd /opt/bootstrappingtrust/VerificationManager ./app	The verification manager will not provision the malicious enclave with a key, as it detects that the measurement of the remote application is not an expected value. This is given by the message "ERROR: Actual MRENCLAVE hash is not in the list of allowed hashes!" from the final lines of the output of the verification manager. If the previous statement is true, then this test is successful.
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Use Case 5.2: Adding a 5G Node to a Virtualized Core Network Feature-5.3.1: Integrity Attestation of Virtual Network Components	

Test Case 5ge-25: Authentication to a micro-segment
Summary: The objective of this test is to check how the micro-segmentation enabler is able to respond to the threat T_UC5.2_1 Add malicious nodes into core network. In this threat malicious nodes may e.g. eavesdrop, tamper, and prevent data flows. The enabler applies security verification procedures, namely IEEE 802.1X based authentication for assuring that the added nodes are trustworthy. This test presumes that the single node version of the enabler has been installed.
Preconditions: Microsegmentation enabler has been deployed on the testbed. The configuration is been done according to the testbed chosen setup. The single node version of the enabler has been installed. sudo login to the test computer required. Unit Tests 1-4 of the microsegmentation enabler have been successful.

Modify the file wpasupplicant-mno01.conf in $HOME/OpenVirteX/scripts/ensure. Change the default password to something else.		
#:	Step actions:	Expected Results:
1	Start the enabler with the start_screen.sh command in directory $HOME/OpenVirteX/scripts/ensure. cd $HOME/OpenVirteX/scripts/ensure ./start_screen.sh	The enabler is started and no error is reported. The start_screen.sh command should generate several windows.
2	Change to screen window 1 (CTRL+A 1) and give your password for sudo command. Next same for screen window 2 (CTRL+A 2). After waiting for a while (5-10s), navigate to screen window 5 (OVX_creation) (by pressing CTRL+A 5) and then press ENTER/RETURN	No error should be reported. Virtual ports and virtual links should be created. If error messages come, the test should be started from the beginning by exiting all screen windows.
3	To test authentication of a possibly malicious node to the first micro-segment, navigate to window 1 (CTRL+A 1, mininet) and execute the following commands: remote ./ensure_test.sh br-ensure port; wpa_supplicant -i tap-ensure -Dwired -c wpasupplicant-mno01.conf	Host will not be authenticated to the micro-segment.
4	In window 1 (mininet), test the connection by the following command: remote ping 192.168.33.1	The ping should not work as the node has not been authenticated to the micro-segment.
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Use Case 5.2: Adding a 5G Node to a Virtualized Core Network Feature-5.4.1: Dynamic Arrangement of Micro-Segments	

Test Case 5ge-65: Validate VNFs element before deployment of a new node
Summary: The objective of this test is to check how an orchestrator must interact with the certificate repository before the deployement of a VNF in order to chec if this VNF has the good TWAttrributers expected by the service owner

Before the deployment of a VNF, the Orchestrator must verify if the VNF is inline with its trustworthy requirements imposed by the slice requirements, the infra, etc... For that, the orchestrator must contact the Certificate Repository in order to find the certificate of the VNFs he wants to deploy, check if these VNFs match the TWAttribute expected and deploy them if yes otherwise, the orchestrator must choose another VNF

Preconditions:

NFV-MANO functional blocks: VNF Manager (VNFM) (e.g. Openstack/Tacker), virtualized infrastructure manager (VIM) and NFV orchestrator (NFVO) must be installed on the testbed together with a VNF catalog.

Some VNFs must populate the VNF catalog (NB: For the moment, only VNF based on Ubuntu virtual machine (14.04 and higher) defined using the TOSCA format are supported).

Each VNF must have an instance running somewhere in order to perform the certification required before the deployment

The necessity to modify the VNF orchestrator to interact with the certification repository in order to check the DTwC of the certificate before the installation (The orchestrator must be modified)

#:	Step actions:	Expected Results:
1	Before the deployment of all the VNFs, some VNFs present into the VNF catalog, must be certified. For that, select 2 VNF with the same functionnality and perform the different step describe in the Unit test 5ge-8 to perfrom the certification. For the certification, the following TWAttributes must be specified Maintenability Security PerformanceEfficiency Maintainability	The VNF certification process must be realised without error and a certificate must be present into the Certificate repository for each certified VNF.
2	Configure the orchestrator with the TWAttributes required. for instance, specify the minimal values for the following attributes: Maintenability Security PerformanceEfficiency Maintainability The values set must match one of the two VNF previoulsy certified The method used to specify this is dependant of the orchestrator implementation (see the preconditions)	The TWAttributes are specified into the orchestrator
3	At deployment time, the VNF orchestrator receives an order to deploy VNFs for a specific slices or micro-segments. He selects one the two VNF previously certify (the one which not match the TWAttributes specified into the orchestrator)	The orchestrator must not deploy the VNF which doesn't match the TWAttributes specified into the orchestrator. the other VNF with the same functionality and which match the TWAttribute must be deploy
4	For another slice, perform the same demand and select directly the VNF which match the TWAttributes	The orchestrator must deploy the VNF selected
Execution type:	Manual	

Estimated exec. duration (min):	
Priority:	Medium
Scenario evaluation score:	**2 - Implementation delivered**
Requirements	Feature-3.1.1: VNF Trustworthiness Evaluation Use Case 5.2: Adding a 5G Node to a Virtualized Core Network Use Case 5.4: Verification of the Virtualized Node and the Virtualization Platform

T_UC5.2_2 Forwarding logic leakage

Description: Detailed description of threat and its importance	A network application running on the controller is able to see the forwarding logic of another application (i.e.: the OpenFlow rules installed in the switches). The applications can belong to different virtual network operators who do not want to leaking sensitive information about how their virtual nodes are located or migrated. The leakage can happen in two directions. Controller-to-switch contains rules that have been installed in the switches. A malicious application can not only intercept the OpenFlow messages as they are sent, it can also request information from the switch about installed rules and related statistics belonging to other applications. Eavesdropping on switch-to-controller (e.g.: OFPT_PACKET_IN) messages can also leak information not only about the forwarding logic, but about application data that might be confidential.
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	Information about forwarding logic is leaked: positioning of network elements like DNS or other services provided through VNFs and how they are migrated which can be used to infer user population, reliability information etc.
Possible Mitigation Hints (optional, if foreseen): How can we protect against the threat?	Insert a reference monitor at the southbound interface.
Entry Points (optional, if known): What possible means does an adversary have?	Deploy an application on the controller in a multi-tenant virtualized network.
5G-ENSURE enablers (optional, if covered for given threat): What possible means does an adversary have?	Enabler 6.2 “Access Control Mechanisms”

Test Case 5ge-95: TLS connection to controller

Summary:

Ensures that a TLS connection is setup between the Application and the Controller, after a successful provisioning of the Application. This makes the communication between the application and the controller both integrity and confidentiality protected.

Preconditions:

The verification manager software is installed on VM1. The remote host software is installed on VM2. The (malicious) application (ApplicationEvil) is also installed on host VM2.

The following files should contain on each VM the IP of the other VM host. For instance for VM1:

/opt/bootstrappingtrust/Certs/rh_host (should have IP address of VM2)
and
/opt/bootstrappingtrust/Certs/container_host (should have IP address of VM2)

#:	Step actions:	Expected Results:
1	Launch tcpdump and start capturing traffic on the application host (VM2). sudo tshark -i lo -f "port 8081" -o http.ssl.port:8081	Tshark starts the capture.
2	Launch the remote host software on VM2. cd /opt/bootstrappingtrust/RemoteHost sudo ./app	It is launched and awaits connections.
3	Launch the benign application on VM2. cd /opt/bootstrappingtrust/Application sudo ./app	The application starts, and awaits connections from the verification manager.
4	Launch the Floodlight SDN controller on VM2, separately from the application. cd /opt/Floodlight sudo java -jar target/floodlight.jar	Floodlight starts.
5	Launch the verification manager on VM1, which will then immedately connect and attest the application. cd /opt/bootstrappingtrust/VerificationManager ./app	The verification manager will provision the enclave with a key. No lines in the output of the vericiation manager will start with "ERROR". Switching to the output from VM2 and the application the following output should be visible near the end, since this indicates that communcation was successful with the floodlight controller. HTTP/1.1 200 OK Content-Type: application/json Date: Tue, 13 Jun 2017 13:37:00 GMT Accept-Ranges: byte Server: Restlet-Framework/2.3.1 Vary: Accept-Charset, Accept-Encoding, Accept-Language, Accept Connection: close 37 bytes read

		{"name":"floodlight","version":"1.2"} EOF
6	Stop the tshark capture (ctrl-c) and investigate the output.	A TLS session, with communcation, can be seen between the application and the controller on port 8081. The output should contain lines similar to: TLSv1.2 134 Application Data which shows that a TLS session has been established and that encrypted data is sent. If this is the case, then this test is successful.
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Use Case 5.2: Adding a 5G Node to a Virtualized Core Network Feature-5.3.1: Integrity Attestation of Virtual Network Components	

T_UC5.3_1 Fingerprinting attack

Description: Detailed description of threat and its importance	Unlike T_UC5.2_2, the attacker is external to the controller. The attacker can measure the time of reconfiguring the physical network. This way, the attacker can gain information about which and when a network packet triggers a reconfiguration of network components.
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	The attacker can exploit the obtained information to mount DoS attacks by overloading the controller with packets that will most likely trigger a reconfiguration of the network. Furthermore, installing flow rules in current SDN switches is a costly operation. This means that even the performance of the physical network can be impacted.
5G-ENSURE enablers (optional, if covered for given threat): What possible means does an adversary have?	Enabler 6.1 “Anti-fingerprinting”

Test Case 5ge-71: Rule Scanning
Summary:

The objective of this evaluation is to check whether a remote adversary can infer (with high probability) whether a flow rule has been already installed by the controller in order to handle a specific type of traffic or route towards a given destination. For example, the adversary can craft probe packets whose headers match the traffic type and/or destination address and infer by measuring the timing of the packets whether these packets triggered the installation of a rule. This provides a strong evidence for the adversary that e.g., communication with the given destination address has recently occurred. Depending on the underlying rule, the adversary might also be able to infer the used network protocol, and the destination port address. By doing so, the adversary obtains

additional information about the occurrence of a particular communication event; for example, the adversary can infer whether the destination address has recently established an SSL session to perform an e-banking transaction. Notice that this leakage is only particular to SDN networks, and does not apply to traditional networks.

Moreover, the remote fingerprinting of rules enables the adversary to better understand the logic adopted by the controller in managing the SDN network. This includes inferring the timeouts set for the expiry of specific rules, whether the controller aims at fine-grained or coarse-grained control in the network, etc. Similar to existing port and traffic scanners, this knowledge can empower the adversary with the necessary means to compromise the SDN network. Even worse, the adversary can leverage this knowledge in order to attack other networks which implement a similar rule installation logic. For instance, in a geographically dispersed datacenter, different sub-domains typically implement the same policies. The adversary can train using one sub-domain and leverage the acquired knowledge in order to compromise another subdomain.

See in particular Section V of the following article.
Heng Cui, Ghassan O. Karame, Felix Klaedtke, and Roberto Bifulco
On the Fingerprinting of Software-Defined Networks
IEEE Trans. Information Forensics and Security 11(10):2160-2173 (2016)

Preconditions:

This evaluation will not be carried out in the 5G-ENSURE testbed. Instead, the evaluation is described in the following artcile, including the setup and the countermeasure.
Heng Cui, Ghassan O. Karame, Felix Klaedtke, and Roberto Bifulco
On the Fingerprinting of Software-Defined Networks
IEEE Trans. Information Forensics and Security 11(10):2160-2173 (2016)

Execution type:	Manual
Estimated exec. duration (min):	0.00
Priority:	Medium
Scenario evaluation score:	**1- Theoretical evidence**
Requirements	Use Case 5.3: Reactive Traffic Routing in a Virtualized Core Network
Attached files	TIFS : main.pdf

Test Case 5ge-72: Denial-of-Service Attacks

Summary:

The rule space is a scarce resource in existing hardware switches. Namely, state-of-the-art OpenFlow hardware switches can only accommodate few tens of thousands rules, and only support a limited number of flow-table updates per second. While these limitations can be circumvented by means of a careful design of the rule installation logic, an adversary that knows which packets cause an interaction with the controller can abuse this knowledge to launch Denial-of-Service (DoS) attacks.

For instance, an adversary might simply try to overload the controller with handling packet-in events. More specifically, the adversary sends packets, where each of them most likely triggers a controller-switch interaction. Too many such interactions will overload the controller.

Another kind of DoS attack is to fill up the switches' flow tables. An analogy to this is when a computer runs out of memory and starts swapping. Usually, the computer becomes unusable. Similarly, the network performance is severely harmed when the flow tables are full (or even almost full). First, installing flow rules in an almost full table is more costly than in an almost empty flow table. Second, in case the flow table is full, either new network flows cannot be established, which would already be a DoS, or some installed flow rules need to be deleted. However, in general, it is not obvious which rules should be deleted to make room for new rules; this needs to be coordinated by the controller and is a complex operation, which can quickly overload the controller and the switches. For example, the deletion of a rule of an ongoing network flow might entail the rule's immediate reinstallation. This can escalate and the controller will have to constantly delete and reinstall rules.

An adversary can make both kinds of DoS attacks more likely to succeed by first passively fingerprinting the network traffic, instead of blindly guessing which packets trigger a controller-switch interaction.

In particular, the adversary is interested to learn prior to launching the DoS attack which packets trigger an interaction between the data plane and the control plane of the SDN network. The objective of this evaluation is whether an attacker can learn which packets cause such an interaction. See the following article for more details.

Heng Cui, Ghassan O. Karame, Felix Klaedtke, and Roberto Bifulco
On the Fingerprinting of Software-Defined Networks
IEEE Trans. Information Forensics and Security 11(10):2160-2173 (2016)

Preconditions:

This evaluation will not be carried in the project's testbed. The evaluation is described in the following article, including the testbed and the countermeasure.

Heng Cui, Ghassan O. Karame, Felix Klaedtke, and Roberto Bifulco
On the Fingerprinting of Software-Defined Networks
IEEE Trans. Information Forensics and Security 11(10):2160-2173 (2016)

Execution type:	Manual
Estimated exec. duration (min):	0.00
Priority:	Medium
Scenario evaluation score:	**1- Theoretical evidence**
Requirements	Use Case 5.3: Reactive Traffic Routing in a Virtualized Core Network
Attached files	TIFS : main.pdf

T_UC5.5_1 Misuse of open control and monitoring interfaces

Description: Detailed description of threat and its importance	Third-party service providers may misuse the access to control and monitoring interfaces and cause service disruptions for the operator or attack against data flows. For instance, monitoring information on flowing data may be captured in order to profile end-users. While interfaces are opened for service providers they may also become available for other adversaries.
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	Resources and user data become available for larger amount of parties. More trusted parties means that there may be parties that do not provide good enough security and follow good security practises.

Possible Mitigation Hints (if known): How can we protect against the threat?	Service providers should be required to protect the monitoring data they acquire. Service providers should protect their own resources sufficiently, so that adversary cannot access slices through service providers' systems. Strong isolation is needed to prevent service providers from accessing resource outside a slice. Service providers should be allowed to access only those control interfaces that are required to minimize service providers potential to escape
Entry Points (if known): What possible means does an adversary have?	Control interfaces can be enable access to operator's functions either directly (if not sufficient fine-grained protection is available) or the interfaces may contain vulnerabilities that may be utilized to gain additional privileges. A service provider itself may be untrustworthy. Alternatively, an adversary may compromise service providers systems in order to gain access to the slice.

Test Case 5ge-74: Monitoring data access control		
Summary: Check that data collected for a service are available only for this service.		
Preconditions: The security policy file is well configured and secured. the temporary repository are well secured. We assume that we have: Two services (S1) and (S2) A client engine collecting data for a service (S1). A server engine that manages collected data for two services (S1) and (S2).		
#:	Step actions:	Expected Results:
1	Run wireshark in the machine that is hosting the server engine (i.e, where "monitoringserver.py" will run)	
3	Run the server engine: execute the script "monitoringServer.py"	On the terminal, we get the message "The server is waiting the reports from monitoring clients..."
4	Run the two services: for every service, execute the script "monitoringService.py"	On terminal, you will see this message: "The service is waiting for the reports from the monitoring server".
5	Run a client engine: execute the script "monitoringClient.py"	Exchanges on wireshark On terminals, messages annoucing the sending and reception will be displayed.
6	Open the wireshark capture and check that the collected data has been sent to the right service (namely service S1)	
Execution type:	Manual	
Estimated exec. duration (min):	15.00	
Priority:	Medium	

Scenario evaluation score:	**3 - Testbed evaluation (simulation)**
Relations	related to - 5ge-4:Test report related to - 5ge-5:test case related to - 5ge-3:Check services
Requirements	Feature-4.3.1: Log and Event collector Use Case 5.5: Control and Monitoring of Slice by Service Provider

Test Case 5ge-77: Monitoring use of control interfaces and effects of misuse

Summary:

By misusing the interfaces, an attacker can cause disruptions in SDN. Such attacks can be detected by monitoring the abnormal behaviour of the SDN data plane and by monitoring who is using the control interfaces. The ***security monitor for 5G micro-segments*** enables A) collection of network statistics that reveal some anomalous behaviour and B) monitoring of who is using controlling nodes in software defined network.

The objective of this case is to show that the enabler is able to monitor and collect network statistics from interfaces that connect SDN control plane to remote hosts. The monitoring enabler is also able to distribute this information to centralized operating device and show it to human administrators.

Limitations: release 1 of the enabler has not been integrated to the SDN network, hence, we are only monitoring coummunication of a host that could act as an SDN controller or SDN swithc. Also, Release 1 does not yet provide any automated threat inferencing or analysis over collected statistics. Here, we assume that the administrator is able to manually recognize malicious attacks.

Note, the implementation of the test is identical to enabler's unit test 2 (5ge-33) and threat coverage tests (5ge-76, 5ge-78). It is not necessary to execute it twice to demonstrate that it works.

Preconditions:

The microsegment monitoring enabler has been deployed.

#:	Step actions:	Expected Results:

1	Starting the monitoring framework (the kafka broker and Zookeeper):
	
	$ cd /usr/local/src/kafka_2.11-0.10.1.0/
	$ bin/zookeeper-server-start.sh config/zookeeper.properties
	$ bin/kafka-server-start.sh config/server.properties	

2	The monitoring probe is deployed to the SDN controller. Here we assume that it is 'the localhost' i.e. all components are running in the same host. Execute libpcap-based pmaccd daemon (pmacctd) by using the command: $ sudo pmacctd -f /usr/share/doc/microsegmentmonitor/tokafka.conf &	The pmacctd will start publishing network statistics information through pmacct.acct topic in the Kafka broker. The statistics are in JSON format and look something like this:{"port_src": 48922,

		"ip_dst": "109.105.109.212", "ip_src": "10.0.2.15", "port_dst": 443, "ip_proto": "tcp", "stamp_updated": "2016-06-16 08:37:31", "stamp_inserted": "2016-06-16 08:35:00", "packets": 5, "bytes": 323}.)	
3	$ cd /usr/local/src/spark-1.6.2-bin-hadoop2.6/ $ bin/spark-submit --packages org.apache.spark:spark-streaming-kafka_2.10:1.6.1 examples/src/main/python/streaming/direct_kafka_wordcount.py localhost:9092 pmacct.acct | grep 192.237.223.114 -B 3 -A 1	The network statistics should be found from the output of the monitoring application. Note that the output may not be easily found as the example application outputs also lots of other information. So we grep the results where the target address (of suspected adversary's server) emerges. If grepping is not used, the output entries that contain network statistics look something like this: Time: 2016-11-17 15:58:32 --- (u'115}', 1) (u'256}', 1) (u'"10.102.254.107",', 2) (u'"icmp",', 1) (u'"ip_dst":', 9) (u'15:55:00",', 9) (u'"port_dst":', 9) (u'2,', 2) (u'39068,', 2) (u'"10.102.8.62",', 9) ...	
4	Create simulated adversarial data. Here we assume that the attacker's purpose is to allow ping protocol messages to traverse through SDN: $ ping www.5gensure.eu	Traffic flows caused by the attacker should be visible from the monitoring application. In this case we can see connections to www.5gensure.eu (192.237.223.115).	
Execution type:	Manual		
Estimated exec. duration (min):			

Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Relations	related to - 5ge-76:Monitoring of controllers' interfaces related to - 5ge-33:Unit test 2 - Pmacct, Kafka, Spark related to - 5ge-78:Monitoring adversarial data flows in network slices	
Requirements	Feature-4.4.1: Complex Event Processing Framework for Security Monitoring and Inferencing Use Case 5.5: Control and Monitoring of Slice by Service Provider	

T_UC5.5_2 Unauthorized access to a network slice

Description: Detailed description of threat and its importance	Isolation of the slice may fail allowing a service provider to gain an access to resources belonging to the operator or other slices. This may jeopardize availability and security of the operators and other services providers' network services.
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	Availability and security of operators' resources and service provider's resources jeopardized. This may prevent opportunities that are gained by opening operator's network to third-party service providers.
Possible Mitigation Hints (if known): How can we protect against the threat?	Strong isolation between slices is needed. Authentication and authorization over the access to control and data plane. Security monitoring is needed to detect ongoing incidents.
Entry Points (if known): What possible means does an adversary have?	Failing or misconfigured authentication and authorization both in the control or data plane may enable access to slices.

Test Case 5ge-78: Monitoring adversarial data flows in network slices
Summary: Unauthorized adversary may be able to circumvent access controls to network slice. Successfull intrusions may be later on detected by monitoring communication flows and searching for anomalies (either distrupting behaviour or just suspicious unexpected behaviour). The objective of this test case is to show that the ***security monitor for 5G micro-segments*** enables collection of network statistics (that may be used to detect anomalous behaviour and to reveal some attacks). Limitations: release 1 of the enabler has not been integrated to the network slice (implemented by SDN network/microsegment enabler), hence, we are only monitoring coummunication of a host that could act as an SDN switch. Also, Release 1 does not yet provide any automated threat inferencing or analysis over collected statistics. Here, we assume that the administrator is able to manually recognize malicious attacks.

Note, the implementation of the test is identical to enabler's unit test 2 (5ge-33) and threat coverage tests (5ge-76, 5ge-77). It is not necessary to execute it twice to demonstrate that it works.		
Preconditions: The microsegment monitoring enabler has been deployed.		
#:	Step actions:	Expected Results:

1	Starting the monitoring framework (the kafka broker and Zookeeper):
	
	$ cd /usr/local/src/kafka_2.11-0.10.1.0/
	$ bin/zookeeper-server-start.sh config/zookeeper.properties
	$ bin/kafka-server-start.sh config/server.properties	

2	The monitoring probe is deployed to the SDN controller. Here we assume that it is 'the localhost' i.e. all components are running in the same host. Execute libpcap-based pmaccd daemon (pmacctd) by using the command: $ sudo pmacctd -f /usr/share/doc/microsegmentmonitor/tokafka.conf &	The probe (pmacctd) will start publishing network statistics information through pmacct.acct topic in the Kafka broker. The statistics are in JSON format and look something like this:{"port_src": 48922, "ip_dst": "109.105.109.212", "ip_src": "10.0.2.15", "port_dst": 443, "ip_proto": "tcp", "stamp_updated": "2016-06-16 08:37:31", "stamp_inserted": "2016-06-16 08:35:00", "packets": 5, "bytes": 323}.)

3	Subscribe and output network statistics with the monitoring application
	
	$ cd /usr/local/src/spark-1.6.2-bin-hadoop2.6/
	$ bin/spark-submit --packages org.apache.spark:spark-streaming-kafka_2.10:1.6.1 examples/src/main/python/streaming/direct_kafka_wordcount.py localhost:9092 pmacct.acct | grep 192.237.223.114 -B 3 -A 1
		The network statistics should be found from the output of the monitoring application. Note that the output may not be easily found as the example application outputs also lots of other information.
		We use grepping so capture traffic flows that we will generate in the next step.
		If grepping is not used, the output entries that contain network statistics look something like this:
		
		Time: 2016-11-17 15:58:32
		--- 	
		(u'115}', 1)
		(u'256}', 1)
		(u'"10.102.254.107",', 2)
		(u'"icmp",', 1)
		(u'"ip_dst":', 9)

		(u'15:55:00",', 9) (u'"port_dst":', 9) (u'2,', 2) (u'39068,', 2) (u'"10.102.8.62",', 9) ...	
4	Create simulated adversarial data. Here we assume that the attacker's purpose is to allow ping protocol messages to traverse through SDN. We also assume that the network slice is not supposed to have either ping protocol or connections to such remote target. $ ping www.5gensure.eu	Traffic flows caused by the attacker should be visible from the monitoring application. In this case we can see connections to www.5gensure.eu (192.237.223.115). The idea here is that microsegments may be dedicated for homogenous traffic types. If we see traffic types (e.g. to particular target address) that are not supposed to be this microsegment, we can be more sure that there is an ongoing attack. Here we assumed that we know all the devices that are allowed to communicate using the microsegment. 5gensure.com was not among the trusted ones, so communication towards it is considered a security problem.	
Execution type:	Manual		
Estimated exec. duration (min):			
Priority:	Medium		
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**		
Relations	related to - 5ge-33:Unit test 2 - Pmacct, Kafka, Spark related to - 5ge-76:Monitoring of controllers' interfaces related to - 5ge-77:Monitoring use of control interfaces and effects of misuse		
Requirements	Feature-4.4.1: Complex Event Processing Framework for Security Monitoring and Inferencing Use Case 5.5: Control and Monitoring of Slice by Service Provider		

T_UC5.5_4 No control of Cyber-attacks by the Service providers

Description: Detailed description of threat and its importance	The use case features a Service Provider (SP) offering its Massively Multiplayer Online Game service to gamers. The Service Provider buys its network service to Virtual Mobile Network Operator (VMNO) which itself relies on an Infrastructure Provider. The VMNO supplies a sub-slice to the SP with the required QoS. The service of the SP is subject to cyber-attacks. The SP wants to manage the cyber-security of its service. It signs a contract with a third party Security Service Operator (SSO) to monitor and remediate to cyber-security attacks. Thanks to the terms of the contract between the SP and the VMNO, the SSO can benefit from network topology information and routing tables from the slice controller. Nevertheless, since it has not the information about the configuration of the NVF and their vulnerabilities, it cannot build a classical attack graph to monitor the cyber-attacks.
Potential effect: What effect it will have on 5G system (network, hosts, applications…)	The Service Provider has no control over the cyber-attacks on its slice.
Possible Mitigation Hints (if known): How can we protect against the threat?	A possible mitigation hint would be to enable the SSO to get access to the information from the infrastructure domain, especially the type of software used for NVF in order to establish the vulnerabilities of it. Another way to mitigate this is to separate the responsibilities by contract between the infrastructure domain and the VMNO. The SP will have to rely on the VMNO interface and will only control its cyber-threats at application level.
Entry Points (if known): What possible means does an adversary have?	An adversary could attack the VNFs, hypervisor or orchestrator of the Infrastructure Provider to compromise the Service Provider's service.

Test Case 5ge-83: Measure number of active UEs to evaluate trust

Summary:

A rather common cyber-attack is a denial-of-service (DoS) attack in which number of active network users quickly overflows. The attacker may activate huge numbers of real or virtual UEs or IoT systems to prevent legitimate users from using a system.

The objective of this test is to show that the Trust Metric Enabler can detect DoS. The enabler monitors number of active UEs which an eNodeB controls and if they reach a pre-defined limit, the enabler indicates that the monitored network section is not trusted.

Limitations: Since the Release 1 version of Trust Metric Enabler gets its input from disk files, it cannot detect traffic spikes in real time mode. This will change in Release 2 version which can monitor NFVs' counters and KPIs.

NOTE: This test is equal to Unit Test 2 (5ge-35), and it is also identical to threat coverage test 5ge-82. Executing one of these tests demonstrates that they all work.

Preconditions:

Trust Metric Enabler and its two input files (number_of_devices.csv and trust_requirements.csv) are implemented. Unit Test 1 is successfully done.

#:	Step actions:	Expected Results:
1	Do not edit the contents of the input file "trust_requirements.csv",	

	keep it as it is after Unit Test 1. Edit the input file "number_of_devices.csv" by adding more rows to it. The file "number_of_devices.csv"should look as follows: 1 7 3 8 2 88 7 12 19 77 1 2	
2	Execute Python script "Trust_Metric_Enabler.py"	The output file "Trust_Metric_Log.csv" should contain character string "False". This indicates that the network is not trusted
Execution type:	Manual	
Estimated exec. duration (min):	0.10	
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Relations	related to - 5ge-35:Unit Test 2: No trust, too many devices related to - 5ge-82:Monitor and control number of active UEs	
Requirements	Feature-3.2.1: Trust metric based network domain security policy management	

Use Cases cluster 8 - Ultra-Reliable and Standalone Operations

T_UC8.1_1 Service failure over satellite capable eNB

Description: Detailed description of threat and its importance	Main threats that may cause a service failure are related to the following activities: Failures or malfunctions: Failure or disruption of communication links Failure or disruption of main supply Failure or disruption of service providers Malfunction of equipment Outages: Network connectivity Loss of physical resources Support services (Internet provider or Electricity provider) Disasters: Natural disasters Environmental disaster Physical attacks: Sabotage Vandalism Terrorists attack A Service Provider (i.e. telecommunications company) has a contract with the Satellite Network Operator (SatNO) to supply a suitable system capacity with some QoS guarantees to be used by its customers. Therefore, the Service Provider has to ensure that the SatNO is providing what is required by the contract (SLA). This threat is particularly acute in ultra-reliable services (i.e. e-health, lifeline communications, military scenarios…).
EO interpretation of the threat:	Accidental or deliberate link failures or traffic congestion may comprise the service availability and should be mitigated reconfiguring the transport network topology.
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	Service availability or traffic congestion
Possible Mitigation Hints (optional, if foreseen): How can we protect against the threat?	Allowing the Service Provider to have some degree of control over their micro-slice or sub network enabling dynamic allocations and network reconfigurations on the fly. Evolving the Transport Network Architecture (TNA) by combining both satellite and terrestrial transport architectures. Once a link failure has been detected, new topology is forwarded to base stations with satellite links and smart antennas, enabling topology reconfiguration according to traffic failures and traffic demands.
Entry Points (optional, if known):	4G backhaul networks are fixed topologies, therefore the network barely manages accidental/deliberate link failures or traffic congestion.

What possible means does an adversary have?	An exhaustive radio planning is needed before base station deployment and new backhaul nodes cannot be easily added.
5G-ENSURE enablers (optional, if covered for given threat): What possible means does an adversary have?	Once a link failure/congestion is detected, Satellite Network Monitoring provides a Topology algorithm to reconfigure the network components.

Test Case 5ge-132: Reconfigure the network topology

Summary:

Checks that the user can configure the security/performance indicators to be collected.
Checks that the updated topology may be forwarded.

The initial topology is configured in step #8 and can be checked in steps #9, #10 and #11.

http://10.102.0.51/lib/attachments/attachmentdownload.php?id=111

The indicators to be collected are configured in step #12. Node 5g-enodeb3 is configured with $MON_SAT_PATH/test/UT01/input/indicators_UT01.5g-enodeb3.json:

ifOperStatus from terrestrial terminal 1.

ifOperStatus from terrestrial terminal 2.

ifOperStatus from satellite terminal 1.

Each node sents the operational state of the interface (ifOperStatus) to the satellite-network-monitoring-server every 10 seconds (snmp_retry_timeout_msg property in $MON_SAT_PATH/client/SatelliteNetworkMonitoringClient.properties). If the operational state of the interface is set to down ("error_value": 2) the node sends an alarm message.

Link failure is emulated in step #13.

The incident/failure is detected in step #14. The satellite-network-monitoring-server is continuosly collecting messages from the message broker (i.e. ActiveMQ). When the SatelliteNetworkMonitoringServer detecs an alarm message (messageType field in the header set to "alarm") it launches the Topology Manager (see "apply" trace in $MON_SAT_PATH/logs/monitoring.log).

The Topology Manager calculates the best topology that fixes the issue based on two KPIs:

Similarity (the final topology should be similar as the original one).

TotalPowerConsumed (the lower the better).

Later, this topology is forwarded to all the nodes.

The final topology can be checked in steps #15, #16 and #17.

http://10.102.0.51/lib/attachments/attachmentdownload.php?id=112

#:	Step actions:	Expected Results:
1	ssh admin5g@<5ge-satellite-network-monitoring-server> pkill -f SatelliteNetworkMonitoring	The server app has been initialized

	rm -f $MON_SAT_PATH/logs/* psql -f $MON_SAT_PATH/server/clean.sql snm admin5g	
2	ssh admin5g@<5ge-satellite-network-monitoring-server> $MON_SAT_PATH/apache-activemq/bin/activemq restart $MON_SAT_PATH/server/monitoring.sh	The server environment has been started up
3	ssh root5g@<5ge-satellite-network-monitoring-client> $MON_SAT_PATH/client/updateIP.sh docker-compose -f $MON_SAT_PATH/client/docker-compose/docker-compose.yml down -v pkill -f SatelliteNetworkMonitoring rm -f $MON_SAT_PATH/logs/* rm -f $MON_SAT_PATH/client/indicators/* rm -f $MON_SAT_PATH/client/topologies/*	The client app has been initialized
4	ssh root5g@<5ge-satellite-network-monitoring-client> docker-compose -f $MON_SAT_PATH/client/docker-compose/docker-compose.yml up -d	The server environment has been started up
5	ssh root5g@<5ge-satellite-network-monitoring-client> docker exec epc_tt1 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh docker exec epc_st1 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh docker exec epc_tt1 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh docker exec 5g-enodeb1_tt1 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh docker exec 5g-enodeb1_tt2 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh docker exec 5g-enodeb1_tt3 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh docker exec 5g-enodeb2_tt1 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh docker exec 5g-enodeb2_tt2 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh docker exec 5g-enodeb3_tt1 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh docker exec 5g-enodeb3_tt2 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh docker exec 5g-enodeb3_st1 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh	The SNMP simulators have been started up

6	ssh root5g@<5ge-satellite-network-monitoring-client> docker exec epc /mnt/SatelliteNetworkMonitoring/5g-enodeb/deploy.sh docker exec 5g-enodeb1 /mnt/SatelliteNetworkMonitoring/5g-enodeb/deploy.sh docker exec 5g-enodeb2 /mnt/SatelliteNetworkMonitoring/5g-enodeb/deploy.sh docker exec 5g-enodeb3 /mnt/SatelliteNetworkMonitoring/5g-enodeb/deploy.sh	The client SW has been deployed
7	ssh root5g@<5ge-satellite-network-monitoring-client> docker exec epc /root/SatelliteNetworkMonitoring/client/startSnmpManager.sh docker exec 5g-enodeb1 /root/SatelliteNetworkMonitoring/client/startSnmpManager.sh docker exec 5g-enodeb2 /root/SatelliteNetworkMonitoring/client/startSnmpManager.sh docker exec 5g-enodeb3 /root/SatelliteNetworkMonitoring/client/startSnmpManager.sh	The SNMP clients have been started up
8	ssh root5g@<5ge-satellite-network-monitoring-server> cd $MON_SAT_PATH/test curl -v -X POST -H "Content-Type: application/json" -H "Accept: application/json" --data "@UT02/input/topology_UT02.epc.json" http://epc:8080/mon-sat-cli/api/v01.00.00/sna/resource/topology curl -v -X POST -H "Content-Type: application/json" -H "Accept: application/json" --data "@UT02/input/topology_UT02.5g-enodeb1.json" http://5g-enodeb1:8080/mon-sat-cli/api/v01.00.00/sna/resource/topology curl -v -X POST -H "Content-Type: application/json" -H "Accept: application/json" --data "@UT02/input/topology_UT02.5g-enodeb2.json" http://5g-enodeb2:8080/mon-sat-cli/api/v01.00.00/sna/resource/topology curl -v -X POST -H "Content-Type: application/json" -H "Accept: application/json" --data "@UT02/input/topology_UT02.5g-enodeb3.json" http://5g-enodeb3:8080/mon-sat-cli/api/v01.00.00/sna/resource/topology	The initial topoloy has been deployed in all the nodes {"header":{"responseCode":0,"msgType":"restResponseMC"},"content":{}}

9	curl -X GET -H "Content-Type: application/json" -H "Accept: application/json" http://5g-enodeb1:8080/mon-sat-cli/api/v01.00.00/sna/resource/topology	In 5g-enodeb1, all the terrestrial terminals are power on
		{"header":{"responseCode":0,"msgType":"restResponseMC"},"content":[
		{
		"node": "tt_1",
		"ip": "172.18.1.1",
		"enabled": "true",
		"status": "MANDATORY_ON"
		},
		{
		"node": "tt_2",
		"ip": "172.18.1.2",
		"enabled": "true",
		"status": "MANDATORY_ON"
		},
		{
		"node": "tt_3",
		"ip": "172.18.1.3",
		"enabled": "true",
		"status": "MANDATORY_ON"
		}
		]}

10	curl -X GET -H "Content-Type: application/json" -H "Accept: application/json" http://5g-enodeb2:8080/mon-sat-cli/api/v01.00.00/sna/resource/topology	In 5g-enodeb2, the terrestrial terminal #1 is power on and the terrestrial terminal #2 is power off
		{"header":{"responseCode":0,"msgType":"restResponseMC"},"content":[
		{
		"node": "tt_1",
		"ip": "172.18.2.1",
		"enabled": "true",
		"status": "MANDATORY_ON"
		},
		{
		"node": "tt_2",
		"ip": "172.18.2.2",
		"enabled": "true",
		"status": "OFF"
		}
		]}

11	curl -X GET -H "Content-Type: application/json" -H "Accept: application/json" http://5g-enodeb3:8080/mon-sat-cli/api/v01.00.00/sna/resource/topology	In 5g-enodeb3, the terrestrial terminal #1 is power on, the terrestrial terminal #2 is power off and the satellite terminal is power off {"header":{"responseCode":0,"msgType":"restResponseMC"},"content":[{ "node": "tt_1", "ip": "172.18.3.1", "enabled": "true", "status": "MANDATORY_ON" }, {

		"node": "tt_2",
		"ip": "172.18.3.2",
		"enabled": "true",
		"status": "OFF"
		},
		{
		"node": "st_1",
		"ip": "172.18.3.11",
		"enabled": "true",
		"status": "OFF"
		}
		]}

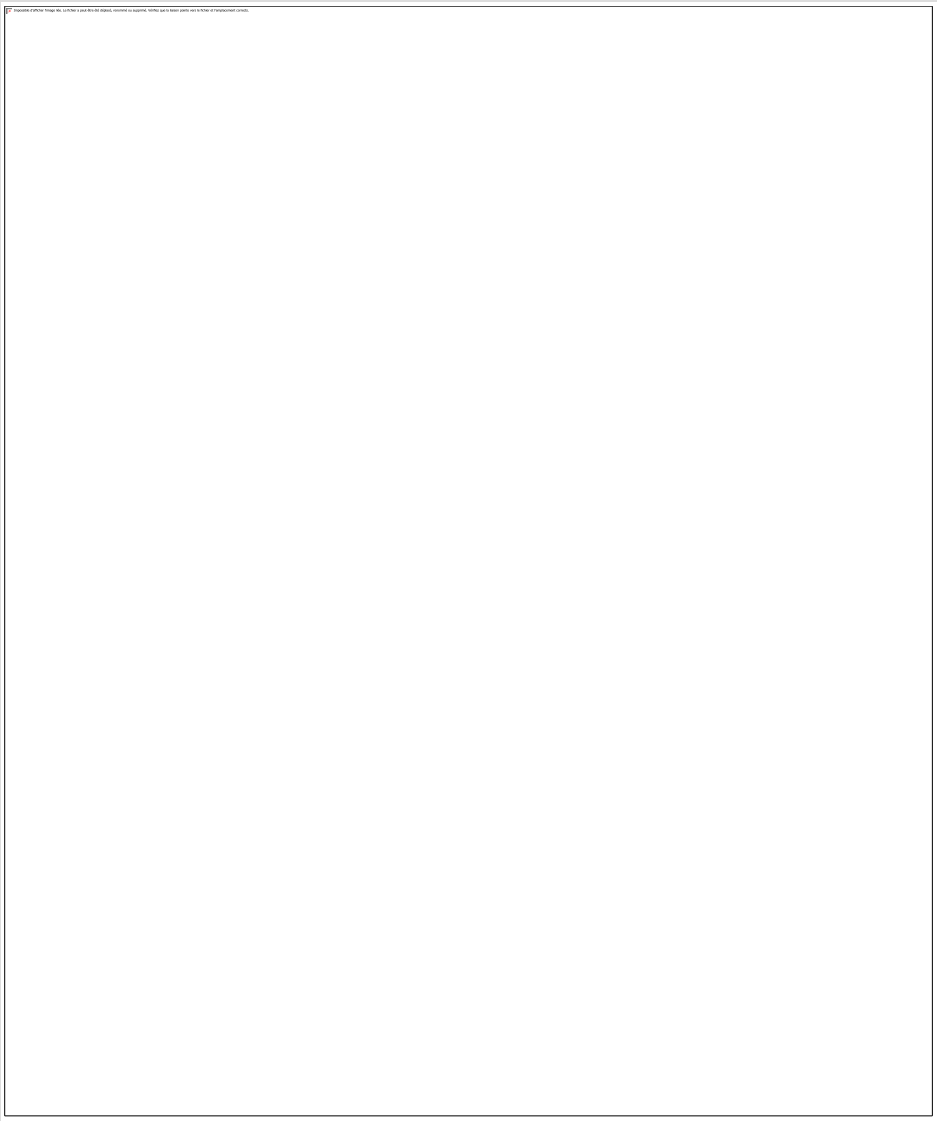
12	curl -v -X POST -H "Content-Type: application/json" -H "Accept: application/json" --data "@UT01/input/indicators_UT01.5g-enodeb3.json" http://5g-enodeb3.5g-ensure.eu:8080/mon-sat-cli/api/v01.00.00/sna/resource/indicators	In 5g-enodeb3, the security/performance indicators has been configured and are sent to the server {"header":{"responseCode":0,"msgType":"restResponseMC"},"content":{}}
13	ssh root5g@<5ge-satellite-network-monitoring-client> docker exec 5g-enodeb3 snmpset -v 2c -c terminal 172.18.3.1 .1.3.6.1.2.1.2.2.1.8.1 i 2	In 5g-enodeb3, emulate a link failure in the terrestrial terminal #1 updating the OID .1.3.6.1.2.1.2.2.1.8.1 in 172.18.3.1 Therefore, the link between 5g-enodeb1 and 5g-enodeb3 is down iso.3.6.1.2.1.2.2.1.8.1 = INTEGER: 2
14	5g-enodeb3 sends an alarm message to the SatelliteNetworkMonitoringServer The SatelliteNetworkMonitoringServer detecs an alarm message in the ActiveMQ (messageType field in the header set to "alarm") and launches the Topology Manager After a few seconds the topology has been reconfigured and the link between 5g-enodeb2 and 5g-enodeb3 is power on	
15	curl -X GET -H "Content-Type: application/json" -H "Accept: application/json" http://5g-enodeb1:8080/mon-sat-cli/api/v01.00.00/sna/resource/topology	In 5g-enodeb1, the terrestrial terminal #3 is power off due to the link failure {"header":{"responseCode":0,"msgType":"restResponseMC"},"content":[{ "node": "tt_1", "ip": "172.18.1.1", "enabled": "true", "status": "MANDATORY_ON", "communities": [] }, { "node": "tt_2", "ip": "172.18.1.2", "enabled": "true", "status": "MANDATORY_ON", "communities": [] }, {

		"node": "tt_3", "ip": "172.18.1.3", "enabled": "false", "status": "OFF", "communities": [] }]}

16	curl -X GET -H "Content-Type: application/json" -H "Accept: application/json" http://5g-enodeb2:8080/mon-sat-cli/api/v01.00.00/sna/resource/topology	In 5g-enodeb2, the terrestrial terminal #2 is power on in order to fix the link failure (to power on the link between 5g-enodeb2 and 5g-enodeb3)
		{"header":{"responseCode":0,"msgType":"restResponseMC"},"content":[
		{
		"node": "tt_1",
		"ip": "172.18.2.1",
		"enabled": "true",
		"status": "MANDATORY_ON",
		"communities": []
		},
		{
		"node": "tt_2",
		"ip": "172.18.2.2",
		"enabled": "true",
		"status": "ON",
		"communities": []
		}
		]}a

17	curl -X GET -H "Content-Type: application/json" -H "Accept: application/json" http://5g-enodeb3:8080/mon-sat-cli/api/v01.00.00/sna/resource/topology	In 5g-enodeb3, the terrestrial terminal #1 is power off due to the link failure and the terrestrial terminal #2 is power on in order to fix the link failure (to power on the link between 5g-enodeb2 and 5g-enodeb3) {"header":{"responseCode":0,"msgType":"restResponseMC"},"content":[{ "node": "tt_1", "ip": "172.18.3.1", "enabled": "false", "status": "OFF", "communities": [] }, { "node": "tt_2", "ip": "172.18.3.2", "enabled": "true", "status": "ON", "communities": [] }, { "node": "st_1", "ip": "172.18.3.11",

		"enabled": "true", "status": "OFF", "communities": [] }]}
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Feature-4.2.1: Pseudo real-time monitoring Feature-4.2.2: Threat detection Use Case 8.1: Satellite-Capable eNB	
Attached files	TopologyMatrix : TopologyMatrix.png	



Use Cases cluster 9 - Trusted Core Network and Interconnect

T_UC9.3_1 Hardening or patching of systems is not done

Description: Detailed description of threat and its importance	If the systems are not hardened correctly or if the patching processes do not keep the systems up-to-date, the systems could be compromised through the vulnerabilities existing in the systems.
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	Systems can be compromised through the vulnerabilities and elevated privileges gained. Thus, total control of a node can be achieved.
Possible Mitigation Hints (optional, if foreseen): How can we protect against the threat?	Monitoring of systems can help in detecting breaches. This can potentially be cooperative actions between different operators, so that indicators of compromise are reported to the operator of the source traffic. Proper segmentation of systems can isolate the breach to only one system. Thus, other systems should be considered potentially hostile.
Entry Points (optional, if known): What possible means does an adversary have?	Abuse of software vulnerabilities in the software
5G-ENSURE enablers (optional, if covered for given threat): What possible means does an adversary have?	Proactive security analysis and remediation Microsegmentation

Test Case 5ge-73: T_UC9.3_1 – "Hardening or patching of systems is not done" R1		
Summary: 5G networks allow more dynamism through virtualisation and new functions can be introduced to the network on the fly. As these environments are more virtualised, there is always a danger that someone manages to introduce a malicious function into the network. Similarly, unauthorized physical elements could be attached to the network, if their authenticity is only based on the location in the network.		
Preconditions: 1) The HN and the VN have a roaming agreement 2) The VN does not have up-to-date patch management 3) There is an exploitable vulnerability in the VN infrastructure 4) Poor physical security of the VN has resulted in the installation of unauthorised device		
#:	Step actions:	Expected Results:
1	After starting the VM the Trust Builder can be accessed in a browser on this URL:localhost:7070/system-modeller	Login page presented to the user (see Figure 3 in the attached file).
3	After clicking on "View Models" we can login.	Model design canvas and the list of previously designed models (if there were any) presented to the user (see

	login: jp@it-innovation.soton.ac.uk Password: de8b3b661ecfc68f9100ab468569a491	Figures 4 and 5)
4	Click on "Create New Model".	This opens a popup window where we can name the model. In the dropdown menu select "Simple Network Model" (see Figure 6 and 7).
5	Click on the Edit control of the new model.	See Figures 7 and 8.
6	Drag items from the assets panel to the canvas.	The mode design canvas opens up (see Figure 8).
7	Give meaningful names to the items.	On the left side various assets are available that can be used for model construction (see Figure 9, 10).
8	Connect the assets by arrows.	By clicking on the asset a green cross appears in the left corner, this indicates that the asset can be connected to other assets. The target assets are marked by a blue tick, showing that a connection can be made between the assets. By clicking on the blue tick icons we can establish connections between assets (see Figure 11, 12, 13).
9	Validate model.	Once the model is constructed it can be validated. This operation is activated by clicking on the red “play” button (see Figure 14).
10	Sort out issues with inferred assets (if there are any).	There is a “red/green” boundary for the “uses” connection. This indicates that the Incoming Relations need to be fixed (see the right panel).
11	First we sort out the “uses” connection between UE and MME. Click on the question mark (?) under Incoming relations.	Under the “Incoming Relations” the user can see a list of incomplete relations that need to be specified by clicking on the “?” (see Figure 15).
12	For the MME select the "specifies service pool" option.	A Change Relation window comes up (see Figure 16).
13	For the UE select the "selectsFrom service pool" option. Save changes.	Change Relation message window comes up (see Figure 17, 18).
14	For the connection between MME and HSS follow similar steps as described for UE and MME connection (see steps 11-13) .	You will notice that the background colour of Incoming Relations should turn blue and the “uses” relation label should change from red indicating that the asset is fully specified and the schema is ready validation (see Figure 19, 20, 21).
15	Now you can click on the “Red button” at the bottom on the canvas.	This validates the schema, might take a minute or so (see Figure 22).
16	To see the the theats associated with MME-H click on the MME-H asset.	Under the Threads on the right side the list of threats related to MME-H is presented (see Figure 23, 24).
17	Click on the Control Set tab (blue) and select a few options, for example “SoftwarePatching”.	As a result of this actions some threats will be resolved, the color will change to green. This final step completes the test (see Figure 25, 26, 27).

Execution type:	Manual
Estimated exec. duration (min):	
Priority:	Medium
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**
Requirements	Use Case 9.3: Authentication of New Network Elements Feature-3.3.1: 5G Asset Model Feature-3.3.2: 5G Threat knowledge base v1
Attached files	Modelling T_UC9.3_1 – “Hardening or patching of systems is not done” TrustBuilder R1 : Moldelling_T_UC9.3_1_ver4.7z

D Test plan design: Enabler's security evaluation (R2)

Use Cases cluster 1 - Identity Management

T_UC1.3_1 Unauthorised activities related to satellite devices or network

Description: Detailed description of threat and its importance	Network Operators (e.g. SatNO) and M2M communications (e.g. updated satellite device SW) require fine-grained access to network resources (e.g. satellite device, eNB…). Also, satellite devices shall be authenticated to access satellite services (e.g. broadband access, direct-to-home services…). These network components and devices are distributed in a wide-area large enough that other wired or wireless network connectivity is not feasible. In this scenario, main threats are related to Unauthorised activities: Unauthorised access Unauthorised administration of devices and systems Falsifications of configurations
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	Information integrity. Information destruction. Service availability.
Possible Mitigation Hints (optional, if foreseen): How can we protect against the threat?	Fine-grained access control focusing on the application level. In case of resource constrain devices (e.g. satellite devices), the fine-grained access control can be based on tokens evaluated directly in the device.
Entry Points (optional, if known): What possible means does an adversary have?	Non updated network components or satellite devices compromise system security/functionality. Wide-area distributed network composed of resource constrained devices (i.e. satellite devices) with high latency.
5G-ENSURE enablers (optional, if covered for given threat): What possible means does an adversary have?	Fine-grained Authorization enabler.

Test Case 5ge-130: Unauthorised user verification
Summary:

An authorized user tries to make a rest petition on a non-authorized resource.

The response of the test, should be that the user is not allowed to perform the operation because the conditions of the petition does not match with the rules applied in the policy (i.e. $FGA_SAT_PATH/test/UT01/input/TestPolicy_UT01a.xml).

Conditions:

- The user is registered in the LDAP server.

- The user is authorized to perform this action.

- The user is non-authorized to perform this action on this resource.

Preconditions:

Execute 5ge-54 in order to install and configure the environment to run.

#:	Step actions:	Expected Results:
1	This step aims to initialize the PAP policies repository with test policy (i.e. $FGA_SAT_PATH/test/UT01/input/TestPolicy_UT01a.xml): The PUT action can be done by Christopher Carroll on resource fga-sat-rcd/api/v01.00.00/satelliteModem/mgmt/startCWCarrier. curl -v -X POST -H "Content-Type: application/xml" -H "Accept: application/json" --data "@UT01/input/TestPolicy_UT01a.xml" http://5g-fga-sat-srv01.5g-ensure.eu:8080/fga-sat-srv/api/v01.00.00/pap/policy	Expected result is HTTP/1.1 status code 201 with the following response body: {"header":{"responseCode":0,"msgType":"restResponseMC"},"content":{}} Inside $FGA_SAT_PATH/server/policies there should be the uploaded policy.
2	Not authorized request for “PUT” action on resource http://5g-fga-sat-cli01.5g-ensure.eu:8080/fga-sat-rcd/api/v01.00.00/satelliteModem/mgmt/stopCWCarrier. The contents of the $FGA_SAT_PATH/test/UT03/input/RequestContent_UT03b.json file is: {"userName":"Christopher Carroll","action":"PUT","resource":"http://5g-fga-sat-cli01.5g-ensure.eu:8080/fga-sat-rcd/api/v01.00.00/satelliteModem/mgmt/stopCWCarrier","content":{"header":{},"content":{"outputPower":1234,"frequencyParameter":4567,"action":"on","blind":false}}} Request resource action from the RCD using the server host as an authorization proxy: curl -v -X POST -H "Authorization: 5G-ENSURE base64(TWlsZHJlZCBEdW5u:bWlEdV81Zw==)" -H "Content-Type: application/json" -H "Accept: application/json" --data "@UT03/input/RequestContent_UT03b.json" http://5g-fga-sat-srv01.5g-ensure.eu:8080/fga-sat-srv/api/v01.00.00/pdp/authorize	Expected result is HTTP/1.1 status code 401 with the following response body: {"header":{"responseCode":100,"errorMsg":"Request is NOT authorized to perform this access","msgType":"restResponseMC"},"content":{}}
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Relations	depends on - 5ge-54:Installing and configure environment related to - 5ge-136:Authorised user verification	
Requirements	Feature-1.2.1: Basic Authorization in Satellite systems Use Case 1.3: Satellite Identity Management for 5G Access	

Test Case 5ge-136: Authorised user verification		
Summary: An authorized user tries to make a rest petition using an user declared inside the policy. The response of the test, should be that the user is allowed to perform the operation because the conditions of the petition does not match with the rules applied in the policy. Conditions: - The user is registered in the LDAP server. - The time when the user is trying to make the petition is in the range 08:00-18:00. - The location from where the user is trying the connection is in Spain. To simulate the above conditions, the policy file in the server can be modified, just for environment verification.		
Preconditions: Execute 5ge-54 in order to install and configure the environment to run.		
#:	Step actions:	Expected Results:
1	curl -v -X POST -H "Content-Type: application/xml" -H "Accept: application/json" --data "@UT01/input/TestPolicy_UT01b.xml" http://5g-fga-sat-srv01.5g-ensure.eu:8080/fga-sat-srv/api/v01.00.00/pap/policy	{"header":{"responseCode":0,"msgType":"restResponseMC"},"content":{}}
2	curl -v -X POST -H "Authorization: 5G-ENSURE base64(TWlsZHJlZCBEdW5u:bWlEdV81Zw==)" -H "Content-Type: application/json" -H "Accept: application/json" --data "@UT03/input/RequestContent_UT03b.json" http://5g-fga-sat-srv01.5g-ensure.eu:8080/fga-sat-srv/api/v01.00.00/pdp/authorize	{"header":{"responseCode":100,"errorMsg":"Request is NOT authorized to perform this access","msgType":"restResponseMC"},"content":{}}
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Relations	related to - 5ge-130:Unauthorised user verification depends on - 5ge-54:Installing and configure environment	
Requirements	Feature-1.2.1: Basic Authorization in Satellite systems Use Case 1.3: Satellite Identity Management for 5G Access	

T_UC1.3_2 Fake roaming from terrestrial network into satellite network

Description: Detailed description of threat and its importance	Due to the fact that 5G is of multi-operator nature, 5G devices shall be connected to different networks. These 5G devices could be identified in either the satellite network or the terrestrial network with a set of credentials that allows access to both networks. Then due to coverage issues the 5G device performs a roaming to the other network. Non-repudiation of SLAs between integrated satellite and terrestrial networks and different operators should be considered. In this scenario, main threats are related to Legal and business category: Breach of SLAs Abuse of personal data from not honestly operators Identity theft: a customer of MNO A (authenticated by A), present an identity of MNO B inside MNO B network thank to the roaming agreement (SIP fraud over VoIP interconnect) Thread agents could be dishonest external operators.
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	Service availability. Information confidentiality.
Possible Mitigation Hints (optional, if foreseen): How can we protect against the threat?	Integrating the envisaged 5G AAA system mechanisms with satellite authentication function using standard interfaces.
Entry Points (optional, if known): What possible means does an adversary have?	Heterogeneous security levels between network operators may allow fraudulent behaviours and permits customers to gain unauthorised access to content, services and resources.
5G-ENSURE enablers (optional, if covered for given threat): What possible means does an adversary have?	Fine-grained Authorization enabler R2.

Test Case 5ge-131: Registered user from unknown location
Summary: An authorised user registered in LDAP server tries to make a REST petition. This is done from an unknown or not registered location (country) in the policy. The only one authorized country is Spain, so to make the right petition should be done from an user registered and from an specified country. The response of the test, should be that the user is not allowed to perform the operation because the conditions of the petition does not match with the rules applied in the policy.

Conditions:

- The user is registered in the LDAP server, and it is using the same user role that the declared in the policy.

- The time when the user is trying to make the petition is in the range 08:00-18:00

- The location from where the user is trying the connection is outside Spain.

To simulate the above conditions, the policy file in the server can be modified, just for environment verification.

Preconditions:

Execute 5ge-54 in order to install and configure the environment.

Depends on the country from where it is executed the code, it should be needed to change in the accepted policy the right one. If this is tested from outside Spain, the policy will dismiss all the requests.

#:	Step actions:	Expected Results:
1	First, initialize the PAP policy repository with the required policiy. $ curl -v -X POST -H "Content-Type: application/xml" -H "Accept: application/json" --data "@UT01/input/TestPolicy_UT01b.xml" http://5g-fga-sat-srv01.5g-ensure.eu:8080/fga-sat-srv/api/v01.00.00/pap/policy	Expected result is HTTP/1.1 status code 201 with the following response body: {"header":{"responseCode":0,"msgType":"restResponseMC"},"content":{}} The required policy has been deployed in the PAP policy repository (i.e. $FGA_SAT_PATH/server/policies).
2	Later, verify the XACML policies are correctly deployed in the PAP policies repository located at the server folder. $ curl -v -X GET -H "Accept: application/json" http://5g-fga-sat-srv01.5g-ensure.eu:8080/fga-sat-srv/api/v01.00.00/pap/policies/TestPolicy_UT01b	Expected result is HTTP/1.1 status code 200 and the response body with the XACML policy previously uploaded: {"header":"responseCode":0,"msgType":"restResponseMC"},"content":"< XACML policy>"} NOTE: The policy is located in the PAP policy repository (i.e. $FGA_SAT_PATH/server/policies)
3	Finally, verify that the authentication/authorization API is functional. $ curl -v -X POST -H "Authorization: 5G-ENSURE base64(Q2hyaXN0b3BoZXIgQ2Fycm9sbA==:Y2hDQV81Zw==)" -H "Content-Type: application/json" -H "Accept: application/json" --data "@UT03/input/RequestContent_UT03b.json" http://5g-fga-sat-srv01.5g-ensure.eu:8080/fga-sat-srv/api/v01.00.00/pdp/authorize	If the request is denied, expected result is HTTP/1.1 status code 401 with the following response body: {"header":{"responseCode":100,"errorMsg":"Request is NOT authorized to perform this access","msgType":"restResponseMC"},"content":{}} If the reques is authorized, expected result is HTTP/1.1 status code 200 with the following response body: {"header":{"responseCode":0,"msgType":"restResponseMC"},"content":{}} The response of the request is OK when the user name Christopher Carrol meets the following conditions, applied in the policies during Unit Test 1: Location: Madrid Time: between 08:00 and 20:00

		Role: SNO NOTE: Depends on the country/time from where it is executed, it should be needed to change the uploaded policy (repeat the test creating a new file named for example TestPolicy_UT01c.xml and changin "Spain" with your country name).
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	High	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Relations	related to - 5ge-135:Registered user from known location depends on - 5ge-54:Installing and configure environment	
Requirements	Use Case 1.3: Satellite Identity Management for 5G Access Feature-1.2.3: AAA integration with satellite systems	

Test Case 5ge-135: Registered user from known location
Summary: An authorised user registered in LDAP server tries to make a REST petition. This is done from an registered country in the policy. The only one authorized country is Spain, so to make the right petition should be done from this specified country or modify the policy to make it match. The response of the test, should be that the user is allowed to perform the operation because the conditions of the petition does match with the rules applied in the policy. Conditions: - The user is registered in the LDAP server, and it is using the same user role that the declared in the policy. - The time when the user is trying to make the petition is in the range 08:00-18:00. - The location from where the user is trying the connection is in Spain. To simulate the above conditions, the policy file in the server can be modified, just for environment verification.
Preconditions: Execute 5ge-54 in order to install and configure the environment.

Depends on the country from where it is executed the code, it should be needed to change in the accepted policy the right one. If this is tested from outside Spain, the policy will dismiss all the requests.

#:	Step actions:	Expected Results:
1	$ curl -v -X POST -H "Content-Type: application/xml" -H "Accept: application/json" --data "@UT01/input/TestPolicy_UT01b.xml" http://5g-fga-sat-srv01.5g-ensure.eu:8080/fga-sat-srv/api/v01.00.00/pap/policy	{"header":{"responseCode":0,"msgType":"restResponseMC"},"content":{}}
2	$ curl -v -X GET -H "Accept: application/json" http://5g-fga-sat-srv01.5g-ensure.eu:8080/fga-sat-srv/api/v01.00.00/pap/policies/TestPolicy_UT01b	{"header":"responseCode":0,"msgType":"restResponseMC"},"content":"< XACML policy>"}
3	$ curl -v -X POST -H "Authorization: 5G-ENSURE base64(Q2hyaXN0b3BoZXIgQ2Fycm9sbA==:Y2hDQV81Zw==)" -H "Content-Type: application/json" -H "Accept: application/json" --data "@UT03/input/RequestContent_UT03b.json" http://5g-fga-sat-srv01.5g-ensure.eu:8080/fga-sat-srv/api/v01.00.00/pdp/authorize	{"header":{"responseCode":0,"msgType":"restResponseMC"},"content":{}}
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	High	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Relations	related to - 5ge-131:Registered user from unknown location depends on - 5ge-54:Installing and configure environment	
Requirements	Use Case 1.3: Satellite Identity Management for 5G Access Feature-1.2.3: AAA integration with satellite systems	

T_UC1.4_1 Compromised data

Description: Detailed description of threat and its importance	In this use case, the MNO needs to collect data about a user from the mobile network (step (c) in Figure 5 of Deliverable D2.1). If the user device or any network component is compromised, this can tamper with the integrity and confidentiality of the collected data. As the metrics provided to the service provider are cryptographically computed based on the collected data, collecting fake data may compromise the metrics, hence, the provided service.
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	In order to provide this enhanced service, the MNO needs to have an assurance about the validity of the collected data. This may imply the use of attestation protocols between the collect points (in the network) and the MNO.
Possible Mitigation Hints (optional, if foreseen): How can we protect against the	In order to protect against this threat, the MNO needs to perform validity checks on the collected data. The solution may include remote attestation protocols and investigation in statistics data processing.

threat?	
Entry Points (optional, if known): What possible means does an adversary have?	An adversary can have one or all the following means: Communication channels, user equipment and a network component
5G-ENSURE enablers (optional, if covered for given threat): What possible means does an adversary have?	Generic collector interface enabler can be part of the solution.

Test Case 5ge-87: ProVerif security analysis of the group-based AKA protocol	
Summary: Feature 1.1.1 is a group-based Authentication and Key Agreement (AKA) protocol in which group authentication parameters are stored on the device outside of the UICC. However, the symmetric long-term key K, which is stored on the UICC, is also used in the protocol. Since parameters stored outside of the UICC could easily be leaked, the fundamental security properties of the protocol must not depend on whether the group authentication parameters are compromised or not. Specifically, an adversary having access to the group authentication parameters must be unable to authenticate to the network or derive a session master key by eavesdropping on communication. If the adversary could manage to derive the session master key, the confidentiality of all the data sent between the machine-type communications (MTC) device and the network would be compromised. Also, the adversary should not be able to break authentication or confidentiality even if, additionally, members of the same group share all its authentication parameters (including the long-term secret) with the adversary. It is proven with ProVerif that the protocol meets confidentiality and mutual authentication when the adversary has access to all the authentication parameters of members in the same group in addition to all group authentication parameters of the MTC device. See the following paper for a presentation of the proof. Giustolisi, R., Gehrmann, C., Ahlström, M. and Holmberg, S., 2017. A secure group-based AKA protocol for machine-type communications. In *International Conference on Information Security and Cryptology ICISC 2016: Information Security and Cryptology–ICISC 2016. 30 November 2016 through 2 December 2016* (pp. 3-27).	
Execution type:	Manual
Estimated exec. duration (min):	
Priority:	Medium
Scenario evaluation score:	**1- Theoretical evidence**
Requirements	Feature-1.1.1: Group authentication by extending the LTE-AKA protocol Use Case 1.4: MNO Identity Management Service
Attached files	A secure group-based AKA protocol for machine-type communications : icisc_cameraready.pdf icisc_cameraready.pdf

Test Case 5ge-146: STRIDE analysis of the ACE framework	
Summary: For Feature 1.2.4. We have analyzed the ACE framework with Microsoft's Threat Modeling Tool to be able to evaluate the security of the ACE-framework. The attached document contains the analysis.	
Execution type:	Manual
Estimated exec. duration (min):	
Priority:	Medium
Scenario evaluation score:	**1- Theoretical evidence**
Requirements	Use Case 1.4: MNO Identity Management Service Feature-1.2.4: Authorization and authentication for RCD based on ongoing IETF standard
Attached files	ACE_threat_report : ACE_threat_report.pdf

Use Cases cluster 2 - Enhanced Identity Protection and Authentication

T_UC2.2_1 Tracking of device's (user's) location

Covered Threats	**T_UC2.2_1 and T_UC2.2_2**
Description: Detailed description of threat and its importance and The EO interpretation of the threat if needed	In some procedures (e.g., initial network attach, paging requests, etc.) in all current mobile networks the IMSI is sent to the network in clear text. This opens the door to subscriber's identity interception/disclosure and unauthorized user tracking attacks. **EO interpretation of the threat**: Sending the subscribers' identifiers (IMSIs) in clear text over the air interface may allow the corresponding user information interception, therefore such identifiers must be concealed (e.g., through encryption).
Potential effect: The effect of the threat on major 5G system/domains	If the 5G network is not able to protect end-user's privacy will be considered less trustworthy by the end-users. The threat mainly affects the AN (Access Network) domain.
Threat Mitigation How can we protect against the threat?	Use of encrypted identifiers when possible. However, devices need to be aware that the communication is targeted for them, so encrypted identifier will become a pseudo-identifier that can be mapped to the device. Frequent changing of temporary identifiers (preferably by using one time temporary identifiers).
Entry Points (optional, if known): Attack Scenarios: what possible means does an adversary have? Also specify attack pre-conditions if any and choose the most feasible/probable attack scenarios. Try to identify a basic attack if possible the one that will be tested by the	Basic attack: passive sniffing of signaling traffic (in the specific test proposed for the feature developed in Release 1 this means the sniffing of Identity Responses on the WiFi interface). Adversaries must link subscription identifiers to the users' identity. This can be achieved by triggering the mobile network into initiating the generation of paging messages to the victim (and thus to victim's terminal). For instance, adversaries may connect users with using social media application to initiate unobtrusive communications. Location tracking can be done at the granularity of base station's coverage or in more detail if the adversary has capabilities to analyse signal directions. Also, detailed location tracking is possible by eavesdropping plaintext signal measurement reports.

Test Suite proposed for evaluation.	
5G-ENSURE enablers and features that cover the threat	Enabler: Privacy Enhanced Identity Protection (PEIP) Feature: Encryption of Lont Term Identifiers

Test Case 5ge-149: IMSI Pseudonymization test - check RTMSI pseudonyms

Summary:

Description: Verify that the feature IMSI Pseudonymization provides different pseudonyms for the same input IMSI value in different attach procedures using EAP-AKA full authentication

Strategy: Configure the public key on the client (wpa_supplicant) and the private key on the server (hostapd). Insert a SIM card (with a known IMSI value) in the smart card reader and connect to the SSID1 WiFi network with EAP-AKA full authentication method. Check that the EAP-AKA authentication is successsful and observe the IMSI value that is transiting in Identity Response messages. Detach (stop the wpa_supplicant process) and connect again to the SSID1 WiFi network with EAP-AKA full authentication. Observe the IMSI value that is transiting in Identity Response messages. Repeat the procedure a desired number of times to test that different identities are used each time.

Preconditions:

1. On the server system install the package librtmsi and hostapd and make sure the opessl library is installed. On the client system installa the wpa_supplicant package and make sure wireshark is also installed.

2. Get USIMs with known secrets (Ki), for example a programmable USIM, and a USB card reader. **Use the SIM card and reader sent by TIIT on the client system.**

On the server system (hostapd):

3. Copy the milenage file (5g_ensure_IMSI_encryption_milenage.db) provided by TIIT in the /etc/hostapd/ folder.

4. Copy the pub_key and priv_key files in the /etc/hostapd/ folder.

5. Review the /etc/hostapd/hostapd.conf configuration file to make sure that the interface value matches your wireless adapter.

6. Start hostapd: sudo hostapd -d /etc/hostapd/hostapd.conf

7. Start the hlr authentication server: sudo hlr_auc_gw -s /tmp/hlr_auc_gw.sock -g /etc/hostapd/hostapd.sim_db -m /etc/hostapd/5g_ensure_IMSI_encryption_milenage.db -p /etc/hostapd/pub_key -d /etc/hostapd/priv_key

On the client system (wpa_supplicant):

8. Copy the pub_key file in the /etc/wpa_supplicant/ folder.

9. Open the wpa_supplicant configuration file (/etc/wpa_supplicant/wpa_supplicant.conf) and check that the *update_config* option is set to 1 to allow wpa_supplicant to overwrite the configuration file in order to store the current anonymous identity.

10. Start a wireshark capture (it may be either on the client or on the server system) on the wifi interface. In the following steps we suppose the wireshark was started on the client*(*)*. *(*) Note that it might be necessary to stop the wpa_supplicant system service "sudo service wpa_supplicant stop"*		
#:	Step actions:	Expected Results:
1	sudo wpa_supplicant -iwlan0 -d /etc/wpa_supplicant/wpa_supplicant.conf	In the wireshark capture check that the authentication is successful and observe the IMSI value that is transiting in Identity Response messages.
2	Ctrl+C	wpa_supplicant process dies
3	sudo wpa_supplicant -iwlan0 -d /etc/wpa_supplicant/wpa_supplicant.conf	In the wireshark capture check that the authentication is successful and observe the IMSI value that is transiting in Identity Response messages and check if it is different from the prior IMSI.
Execution type:	Manual	
Estimated exec. duration (min):	15.00	
Priority:	Medium	
Scenario evaluation score:	**4 - Testbed evaluation (real flows)**	
Relations	related to - 5ge-125:Supplementary Test: Check the RTMSI pseudonyms with Wireshark related to - 5ge-151:IMSI Pseudonymization test - check RTMSI pseudonyms	
Requirements	Use Case 2.2: Subscriber Identity Privacy Feature-2.1.3: IMSI Pseudonymization	
Attached files	IMSI_Pseudonymization_test_description.txt	

T_UC2.2_2 Mobile user interception and information interception

Covered Threats	**T_UC2.2_1 and T_UC2.2_2**
Description: Detailed description of threat and its importance	In some procedures (e.g., initial network attach, paging requests, etc.) in all current mobile networks the IMSI is sent to the network in clear text. This opens the door to subscriber's identity interception/disclosure and unauthorized user tracking attacks. **EO interpretation of the threat**: Sending the subscribers' identifiers (IMSIs) in clear text over the air interface may allow the corresponding user information interception, therefore such identifiers must be concealed (e.g., through encryption).
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	User privacy violation through IMSI (International Mobile Subscriber Identity) interception and tracking.
Possible Mitigation Hints (optional, if foreseen): How can we protect against the threat?	If the 5G network is not able to protect end-user's privacy will be considered less trustworthy by the end-users. The threat mainly affects the AN (Access Network) domain.
Entry Points (optional, if known): What possible means does an adversary have?	Basic attack: passive sniffing of signaling traffic (in the specific test proposed for the feature developed in Release 1 this means the sniffing of EAP-AKA Identity Responses on the WiFi interface). Adversaries must link subscription identifiers to the users' identity. This can be achieved by

	triggering the mobile network into initiating the generation of paging messages to the victim (and thus to victim's terminal). For instance, adversaries may connect users with using social media application to initiate unobtrusive communications. Location tracking can be done at the granularity of base station's coverage or in more detail if the adversary has capabilities to analyse signal directions. Also, detailed location tracking is possible by eavesdropping plaintext signal measurement reports.
5G-ENSURE enablers (optional, if covered for given threat): What possible means does an adversary have?	Enabler: Privacy Enhanced Identity Protection (PEIP) Feature: Encryption of Long Term Identifiers

Test Case 5ge-86: ProVerif privacy analysis of the group-based AKA protocol	
Summary: Feature 1.1.1 is a group-based Authentication and Key Agreement (AKA) protocol. A machine-type communications (MTC) device using the protocol identifies itself by the combination of a group identifier, called GID, and a value that identifies the device within the group, called PATH. Since the long-term key K (stored in the UICC) is needed for a device to authenticate using the protocol, the device identifier (GID, PATH) is associated with an International Mobile Subscriber Identity (IMSI). However, in order to achieve MTC identity privacy, it is important that an adversary cannot identify the IMSI by observing a run of the group-based AKA protocol, even though the group-based AKA device identifier is sent in the clear. The following paper presents a ProVerif verification proving that the protocol meets this MTC identity privacy property. Giustolisi, R., Gehrmann, C., Ahlström, M. and Holmberg, S., 2017. A secure group-based AKA protocol for machine-type communications. In *International Conference on Information Security and Cryptology ICISC 2016: Information Security and Cryptology–ICISC 2016. 30 November 2016 through 2 December 2016* (pp. 3-27).	
Execution type:	Manual
Estimated exec. duration (min):	
Priority:	Medium
Scenario evaluation score:	**1- Theoretical evidence**
Requirements	Feature-1.1.1: Group authentication by extending the LTE-AKA protocol Use Case 2.2: Subscriber Identity Privacy
Attached files	A secure group-based AKA protocol for machine-type communications : icisc_cameraready.pdf

Test Case 5ge-151: IMSI Pseudonymization test - check RTMSI pseudonyms	
Summary: The same test as 5ge-149 also proves the coverage of this threat.	
Execution type:	Manual
Estimated exec. duration (min):	0.00

Priority:	Medium
Scenario evaluation score:	**4 - Testbed evaluation (real flows)**
Relations	related to - 5ge-149:IMSI Pseudonymization test - check RTMSI pseudonyms
Requirements	Use Case 2.2: Subscriber Identity Privacy Feature-2.1.3: IMSI Pseudonymization

T_UC2.1_2 Tracking of device's (user's) location

Test Case 5ge-144: Device Identity Privacy Evaluation R2		
Summary: This evaluation test should demonstrate that the DIP enabler R2 DNA privacy enhancement features for both retest for R1 features (Dummy address injection and Random ordering) and R2 features (Dummy address injection automatic mode and Geolocation prefiltering) provide for improvement of path location privacy.		
Preconditions: The DIP integration and evaluation packages have been installed: dip-test-scripts_2.0_amd64.deb, dhcpcd-dip_2.1_amd64.deb Dependencies: python-scapy, libcurl3 The DIP integration configuration script needs have been run once as root (/opt/dip/dip-tests-R2/dip-configure.sh) All commands need to run as root (e.g. start with sudo -s) To start the 4 simulated Access Points - four instances of hostapd are started on interface wlan1-wlan4, after which the ISC DHCP server is restarted to listen on the APs. # ./dip-init.sh		
#:	Step actions:	Expected Results:
1	The following eval script will instantiate an LXC unshare container shell attached only to wlan0 and start an associated wpa_supplicant daemon. Then the script will run the evaluation tests - which will restest R1 features and show test results for the R2 features: # ./dip-eval-test.sh	The result of the test should initially indicate that the R1 enabler DNA privacy enhancement features (Dummy address injection and Random ordering) provide privacy improvement and that R2 features DNA_Dummy automatic mode which automates the choice of the number of injected dummy addresses dependent upon the number of remaining leases, and DNA_Geolocation pre-filtering which pre-filters the DNA MAC address pairs for geolocatability - removing them if they can be geolocated (this is simulated by launching a test geolocation server which contains a specified list of co-located geolocatble MAC addresses - emulating Google's geolocation API). After the simultation phase there follows the analysis and results phase - which should look similar to the below: firstly showing privacy improvement ratio above zero and secondly showing that the analysis of R2 features was successfull: Analysis and results phase ---------- - Processing and filtering packet captures and leases files [Path 'Privacy improvement ratio' (using Ratcliff-Obershelp algorithm)

		as compared to standard DNA: (0.0(No privacy) - 1.0(Anonymous))] DNA_Random feature (R1): => Privacy improvement ratio: 0.333333333333 DNA_Dummy feature (R1): => privacy improvement ratio: 0.142857142857 DNA_Dummy auto mode feature (R2): * Number of Remaining leases: 4 * Number of Captured MACs: 6 => Expected number (2) of DNA_Dummy MACs seen. DNA_Geolocation pre-filtering feature (R2): * Geolocatable MAC list in file: geolocatables * Checking against remaining (4) leases. => Successfully removed all geolocatable MACs pairs Evaluation Completed.
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Use Case 2.1: Device Identity Privacy Feature-2.2.1: Enhanced privacy for network attachment protocols Feature-2.2.2: Anonymous and optimised address selection for network attachment protocols	

Use Cases cluster 3 - IoT Device Authentication and Key Management

T_UC3.1_1 Authentication traffic spikes

Description: Detailed description of threat and its importance	Simultaneous or periodic authentication events may cause excessive amount of traffic for network. Adversaries – aiming to perform a denial-of-service attack - may try to initiate traffic spikes or emphasize the effects of natural traffic spikes with IoT application specific means. As a consequence, the network will experience more signalling and authentication functions needs to perform more processing. Potentially, the authentication of devices may fail and devices may lose connectivity.
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	The 5G network must be over-resourced in order to handle large short-term traffic amounts.
Possible Mitigation Hints (if known): How can we protect against the threat?	Different means may be utilized to mitigate traffic spikes. Methods include relying gateway or one group member to perform authentication on the behalf of individual devices. For instance, using group authentication schemes such as [3]. Monitoring and filtering approaches can be used to mitigate effects.
Entry Points (if known): What possible means does an adversary have?	The traffic spikes may emerge naturally in the IoT network as devices may be programmed e.g. to join the network at the same time. However, an adversary may try to guide this behaviour with different means, for instance, by tampering network time or causing power outages to get large amount devices to authenticate at the same time.

Test Case 5ge-85: ProVerif security and privacy analysis of the group-based AKA protocol	
Summary: An authentication scheme for IoT devices that aims to mitigate the authentication traffic spikes threat must still provide adequate security and privacy, otherwise the effect could be that an adversary can break authentication, derive a session master key or compromise the privacy. In the paper referenced below a ProVerif analysis of the group-based AKA protocol (feature 1.1.1) is presented. It is proven that the protocol meets mutual authentication, key confidentiality and device identity privacy. Giustolisi, R., Gehrmann, C., Ahlström, M. and Holmberg, S., 2017. A secure group-based AKA protocol for machine-type communications. In *International Conference on Information Security and Cryptology ICISC 2016: Information Security and Cryptology–ICISC 2016. 30 November 2016 through 2 December 2016* (pp. 3-27).	
Execution type:	Manual
Estimated exec. duration (min):	
Priority:	Medium
Scenario evaluation score:	**1- Theoretical evidence**
Requirements	Feature-1.1.1: Group authentication by extending the LTE-AKA protocol

	Use Case 3.1: Authentication of IoT Devices in 5G
Attached files	A secure group-based AKA protocol for machine-type communications : icisc_cameraready.pdf

T_UC3.1_2 Compromised authentication gateway

Description: Detailed description of threat and its importance	Compromised and maliciously acting node providing authentication on the behalf of a group – an IoT gateway or a mobile phone - may endanger IoT devices' security. Authenticating node may act as a man-in-the-middle – tamper or eavesdrop communication – or provide tampered security configurations. As a result, data collected from IoT devices may leak from to wrong parties and IoT devices may receive commands from malicious party.
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	5G network will have more potentially misbehaving end-points. Application services cannot rely on strong authentication of individual nodes.

Test Case 5ge-147: STRIDE analysis of the ACE framework	
Summary: For Feature 1.2.4. We have analyzed the ACE framework with Microsoft's Threat Modeling Tool to be able to evaluate the security of the ACE-framework. The attached document contains the analysis.	
Execution type:	Manual
Estimated exec. duration (min):	
Priority:	Medium
Scenario evaluation score:	1- Theoretical evidence
Requirements	Use Case 3.1: Authentication of IoT Devices in 5G Feature-1.2.4: Authorization and authentication for RCD based on ongoing IETF standard
Attached files	WP2 Evaluation score : 5ge-147-WP2Evaluationscore.txt ACE_threat_report : ACE_threat_report.pdf

T_UC3.2_1 Leaking keys

Description: Detailed description of threat and its importance	End-to-end keys may be stolen or leak from the centralized key servers. The key server may also become tampered. As a consequence, the end-to-end secured communication is vulnerable for different attacks and adversaries gain an access to the end-points. The may e.g. provide false information to application services or send malicious commands to IoT devices.

Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	The leaking keys will compromise the security (confidentiality and integrity) of those applications that are end-to-end secured.
Possible Mitigation Hints (if known): How can we protect against the threat?	The key server could be used only for authentication purposes and not for delivering the sessions keys. This would make attacks more difficult as the attacker would be required to compromise the server to provide wrong (asymmetric) authentication keys and then mount an interception attack on the end-to-end communication. However, all IoT devices may not be computationally capable to asymmetric key operations. The key server should be hardened to withstand attacks. The server cannot be isolated from the open internet as it needs to be available for the clients. However, some isolation techniques – e.g. micro-segmentation – may be utilized to control which applications may access the server.
Entry Points (if known): What possible means does an adversary have?	Attacker may compromise the key server in various ways. For instance, the attacker may utilize vulnerabilities in server interfaces to gain an access to the service. Lawful interception mechanisms may be vulnerable and leak keys for third-party attackers or authorities that are misusing their privileges.

Test Case 5ge-94: No key in plain-text

Summary:

The private key required for accessing the controller should never be available in clear text on the system. This prevents the key from being leaked to an adversary.

Preconditions:

The verification manager software is installed on VM1. The remote host software is installed on VM2. The application is also installed on host VM2.

The following files should contain on each VM the IP of the other VM host. For instance for VM1:

/opt/bootstrappingtrust/Certs/rh_host (should have IP address of VM2)
and
/opt/bootstrappingtrust/Certs/container_host (should have IP address of VM2)

#:	Step actions:	Expected Results:
1	Launch the remote host software on VM2. cd /opt/bootstrappingtrust/RemoteHost sudo ./app	It is launched and awaits connections.
2	Launch the benign application on VM2. cd /opt/bootstrappingtrust/Application sudo ./app	The application starts, and awaits connections from the verification manager.
3	Launch the Verification manager on VM1, which wil then immediately try to connect and try to attest the integrity of the application. cd /opt/bootstrappingtrust/VerificationManager	The verification manager will provision the enclave with a key. No lines in the output of the verification manager will have the text "ERROR". Switching to the output from VM2 and the application, the

	./app	following output should be visible near the end, since this indicates that a key was provisioned (although we can't connect since the SDN controller is not running) . Seeding the random number generator... ok . Loading the CA root certificate ...OK ok (0 skipped) . Connecting to tcp/localhost/8081... failed ! mbedtls_net_connect returned -68
4	Search for a private key on the application host (VM2). All generated private keys will have a header containing the words BEGIN PRIVATE KEY, so this string can be searched for. sudo grep -H -r -I "BEGIN PRIVATE KEY" /opt/bootstrappingtrust	Except for source code matches in mbedtls-2.2.1 directories, and two keys related to the test-bed, ./Certs/ca/ca_key ./Certs/server_app.key , no matches should be found. If the statement above holds, then this test is successful.
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Use Case 3.2: Network-Based Key Management for End-to-End Security Feature-5.3.1: Integrity Attestation of Virtual Network Components	

Use Cases cluster 5 - Software-Defined Networks, Virtualization and Monitor

T_UC5.1_1 Misbehaving control plane

Description: Detailed description of threat and its importance	Malicious or compromised control plane may jeopardize the network and the data plane. For instance, a compromised SDN controller or virtualization orchestrator may prevent data flows or direct them to a man-in-the-middle switch for eavesdropping or tampering. Centralized network controllers are an alluring targets for attacks as adversaries are not required to compromise switches or network functions it is enough that they steer data flows to their own malicious components.
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	The network and applications become vulnerable to eavesdropping and tampering as well as denial-of-service attacks.
Possible Mitigation Hints (if known): How can we protect against the threat?	Strong protection should be provided for control plane components. They should authenticate and authorize commands and support up-to-date trusted interfaces.
Entry Points (if known): What possible means does an adversary have?	To compromise control plane: Adversaries may send malicious commands / policies to the controller, if controller does not strongly authenticate and authorize the source of the policies. As a consequence, a legitimate controller will behave maliciously according to adversaries' policies. Alternatively, adversaries may compromise legitimate control plane component, for instance, by utilizing weaknesses in the controller and its interfaces. Adversaries may also get credentials to provide the controller policies using e.g. social engineering attacks against the operator. A data plane may be misconfigured so that it accepts control commands also from other slices or external parties. If data plane does not authenticate commands from the controllers, an adversary may masquerade as legitimate control plane component and send malicious southbound control messages.

Test Case 5ge-99: Detection and mitigation of malicius traffic directed to critical network function
Summary: This test aims at detecting and mitigating malicious traffic pattern targeting vital VNFs deployed in vEPC. The Flow Control enabler is deployed as a gateway for the VNF to protect, providing filtering and shaping for incoming traffic. In the test case, a DoS attack is performed against the vMME, a key node of the EPC that performs Mobility management A DDoS attack against the MME (e.g., overloading through a botnet of infected devices) would prevent the network from operating. To be successful the test should be able to identify and block the malicious traffic while not blocking the legitimate traffic.
Preconditions: Flow Control container setup and running , deployed as the unique gateway for the traffic from/to the protected VNF (MME). Background traffic coming from real devices (or simulated).

#:	Step actions:	Expected Results:
1	Start the Flow Control application: From /app/execution "bash start.sh"	Three screens (tabs) are created that can be navigated through Ctrl-a n Tab named "ofdatapath" is the data-plane and perform switching functionalities Tab named ofcontrol is the OpenFlow control protocol between the ofdatapath and the controller Tab named controller is the OpenFlow controller for the switch and performs DDoS detection and mitigation
2	Craft a volumetric DDoS attack targeting the protected VNF (MME) from an outside (rogue) host: hping3 --udp 10.0.0.1 -I eth1 -q -n -d 110 -k -p 99 -a 1.2.3.4 --flood Here it is assumed that target MME is located at "10.0.0.1" and that it the attack is carried out on interface named "eth1"	Attack is detected inside the "controller" tab of Flow Control enabler. Drop rules matching the source(s) of the attack are installed on the datapath
3		Traffic coming from attacker(s) is nowblocked Can be confirmed by logging tcpdump output at the MME node Legitimate traffic is not dropped
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**4 - Testbed evaluation (real flows)**	
Relations	related to - 5ge-98:Setup check	
Requirements	Use Case 5.1: Virtualized Core Networks, and Network Slicing Feature-5.5.1: Detection of malicious behaviors Feature-5.5.2: Mitigation of detected malicious behaviors	

Test Case 5ge-110: Removal check of misbehaving node in micro-segment
Summary: This scenario comprises three enablers, namely, the compliance checker (CC), the micro-segmentation enabler (MSE), and the micro-segmentation monitoring enabler (MSME). CC checks—based on the information it receives from the two other enablers—that malicious nodes identified by the MSME are eventually deleted within a specified deadline by the MSE from the micro-segment. Other policies are possible, e.g., that the MSE only removes nodes from the micro-segment that the MSME has previously identified as malicious. In this scenario, the CC acts here as a control mechansim that checks that the MSE and the MSME interact with each other as intended. Note that this scenario was part of the EuCNC demo by VTT and others showing the use of micro-segments. See the EuCNC video. The theoretical underpinnings, the algorithms used by the CC are described in the following conference paper together with an experimental evaluation of the tool's performance. D. Basin, F. Klaedtke, and E. Zalinescu. Runtime Verification of Temporal Properties over Out-of-Order Data Streams. In Proceedings of the

29th International Conference on Computer Aided Verification (CAV). Lecture Nodes in Computer Science, volume 10426, Springer 2017.

Preconditions:

For the sake of simplicity, the MSE and MSME are not instrumented and we simulate their interaction with the CC by console comands that send the relevant messages to the CC. This has the advantage that only the CC needs to run.

The CC must be installed and the appropriate configuration files for the checked policy must be given. These files are given as attachments to this scenario.

#:	Step actions:	Expected Results:
1	Start three consoles. We name them the CC console, the MSE console, and the MSME console.	
2	**Starting the CC.** Run the tool runverif in the CC console: CC> runverif –prefix malnodedeletion –violations –loglevel 3 It will open a UDP socket over which it will receive messages from the MSE and MSME.	Some information is logged in the file /tmp/runverif.log. The information is also printed on the CC console: INFO: 2017/08/31 10:01:30 1 out of 1 CPU core is used. INFO: 2017/08/31 10:01:30 Garbage collection target percentage is set to 100. INFO: 2017/08/31 10:01:30 2 components are monitored: MSE, MSME INFO: 2017/08/31 10:01:30 Verdicts are with respect to the specification: (FREEZE id[id]. ((NOT MaliciousNode(id)) OR (TRUE UNTIL[0s,100ms] DeleteNode(id)))) INFO: 2017/08/31 10:01:30 Only violations will be reported. INFO: 2017/08/31 10:01:30 Monitoring algorithm: mtldata INFO: 2017/08/31 10:01:30 UDP socket (port: 50010) is open.
3	**Sending alive messages.** When sending alive messages, the CC will not output anything. However, it will update its internal state. For example, we can send an alive message from the MSE console: MSE> echo -n "1.00@[MSE] (1): Alive()" | nc -u -q1 localhost 50010 Similarly, we can send an alive message from the MSME console: MSME> echo -n "1.10@[MSME] (1): Alive()" | nc -u -q1 localhost 50010 Note that the command-line arguments of the nc command might differ slightly differ, depending on the Linux distribution.	

4	**Sending a malicious node message.** Assume that the MSME identified two nodes that are malicious. It will send the following messages to the CC. We do this in this scenario by the following commands from the MSME console. MSME> echo -n "2.00@[MSME] (2): MaliciousNode(1234)" | nc -u -q1 localhost 50010 MSME> echo -n "2.01@[MSME] (3): MaliciousNode(9876)" | nc -u -q1 localhost 50010 Note that the sequence numbers of the messages are 2 and 3. The message sent to the CC with the sequence number 1 was the alive message in the previous step. Again, the CC will not output anything but updates its state. Namely, the CC's state is waiting now for two delete node messages, one with the identifier 1234 and another one with the identifier 9876. Both these messages must have a timestamp within the deadline of 100ms.	
5	**Sending a delete node message.** Assume that the MSE deletes the node 9876. Therefore, it sends the following message to the CC. This message has the sequence number 2 since it is the second message from the MSE to the CC. MSE> echo -n "2.03@[MSE] (2): DeleteNode(9876)" | nc -u -q1 localhost 50010 Again, the CC's state is update but nothing is output. Because of the command-line argument -violations in step 2, only violations are reported. Note that the node 1234 can still be deleted. There is enough time left before the deadline of 100ms is over.	
6	**Sending more alive messages.** When sending the following alive message from the MSE console, there will be no output on the runverif console. MSE> echo -n "3.00@[MSE] (3): Alive()" | nc -u -q1 localhost 50010 This might be surprising at first thought, since the deadline has passed for the MSE to delete the malicious node 1234. The reason is that the CC does not know that only the MSE can delete nodes. In principle, the MSME could send a delete node message for the node 1234 that is within the deadline. With the following alive message, the CC can conclude that the node 1234 was not deleted in time. MSME> echo -n "4.00@[MSME] (4): Alive()" | nc -u -q1 localhost 50010 Note that we assume the both the MSE and the MSME continously send alive messages to the CC.	VERDICT: @2.000000000: false
7	**Additional notes.** The order in which the messages are sent is actually irrelevant. The CC correctly deals with out-of-order message delivery. The timestamps of the messages determine the order, not when they are sent or received. Timestamps are given in Unix time. The precision of the integral part is in seconds. The fractional part is not mandatory. However, with the assumption that timestamps are unique, the precision should either be milliseconds or even microseconds. Another specification would be that nodes should only be deleted when they are identified as malicious previously (e.g., 10 seconds before). That is, the MSE does not "randomly" delete nodes. There should be a reason for a node deletion. This	

	can be checked by the CC by changing the specification. FREEZE id. DeleteNode(id) IMPLIES ONCE[0,10s] MaliciousNode(id)	
Execution type:	Manual	
Estimated exec. duration (min):	30.00	
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Use Case 5.1: Virtualized Core Networks, and Network Slicing Use Case 5.4: Verification of the Virtualized Node and the Virtualization Platform Feature-5.2.2: Basic NFV Reconfiguration Compliance Checker	
Attached files	malnodedeletion.log malnodedeletion.spec malnodedeletion.msgs malnodedeletion.comp	

Test Case 5ge-120: Capture attack against VNFM

Summary:

This test aims at checking that an attack leveraging a compromised control plane (VNF Manager) is detected by CyberCAPTOR.
It uses en example topology where a VNF is present with vulnerabilities that permits to take control of its VNF manager.

Preconditions:

it is assumed that the test is performed on a machine with the following IP address : 10.102.8.68

pulsar-V1.8.1 container running with port 8080 redirecting to container's port 8080, and port 8000 redirecting to container's port 8000

cyber-data-extract-V1.8.1 should be running with its configuration set to access the pulsar API (ie http://10.102.8.68:8080)

for instance launched using the following command, in order to overwrite the embedded config file :

docker run -it -v ${PWD}/auto-fetcher-config-10.102.8.68.yaml:/root/cyber-data-extract/auto-fetcher-config.yaml cyber-data-extract:1.8.1

the auto-fetcher-config-10.102.8.68.yaml is provided as an attached file.

#:	Step actions:	Expected Results:
1	Open a browser (preferrably Chrome) and go to the web interface address : http://10.102.8.68:8000/	The "initialization" page should load
2	change the API server to the address of PulSAR server : http://10.102.8.68:8080 and click on "Save"	Nothing should happen. However if future file upload fails, repeat this step
3	Click on the "Attack Path" tab on top of the page	The "Attack Path" tab should appear
4	On the selector ("Select the path (by target)") select the attack path "execCode(mme_1,vnfmAttacker)"	The attack path should appear on the left
5	On the attack path view,hover the black circle (corresponding to the path target) and follow graph arcs from this point.	One should notice a "Take of control of the VNF Manager from VNF" rule, showing that corrupt control plane cases are handled

6	Threats covered are both: -T_UC5.1_1 : Misbehaving control plane Scenario models a situation where an attacker takes control of part of the control plane (a VNF Manager) and can access VNFs through it. - T_UC5.5_1 : Misuse of open control and monitoring interfaces Scenario models a situation where open control and monitoring interface is unsecured, letting an attacker break in the network and start an attack from any machine (attackerLocated(host2) for instance).	Attack graph here works for both cases where an attack through control interfaces enables to attack a VNF manager and take control of the VNF. Remediation is to patch the vulnerabilities of the weak control interface of a VNF and to redeploy on a non vulnerable VNF Manager (before patching the vulnerable VNF manager).
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Relations	depends on - 5ge-126:Cyber-data-extract running depends on - 5ge-37:API running depends on - 5ge-38:Attack graph generation depends on - 5ge-39:Custom attack graph generation depends on - 5ge-40:Web UI running	
Requirements	Use Case 5.1: Virtualized Core Networks, and Network Slicing Use Case 5.5: Control and Monitoring of Slice by Service Provider Feature-4.1.2: 5G specific vulnerability schema implementation	
Attached files	configuration de cyber-data-extract : auto-fetcher-config-10.102.8.68.yaml GCI report : gci-report2.xml	

Test Case 5ge-138: Reactive adding of flow rules in SDN networks

Summary:

In this scenario, the compliance checker is used to check a simple policy about the interactions between the SDN controller and SDN switches. Namely, whenever a switch receives a network packet with no matching flow rule, the controller must reconfigure the switch accordingly, within a time bound. In other words, the compliance checker checks that the controller timely reacts to packet-in OpenFlow messages by corresponding flow-mod OpenFlow messages.

For the moment, we restrict ourselves to this simple policy. Other, more complex, policies about the interactions via OpenFlow messages between the control plane and the data plane can be checked accordingly. An example is that barrier requests are handled appropriately. However, the setup will be more involved and we want to keep things simple here.

In the following, we describe how to configure, setup, and run the different involved components, namely, runverif, OVS, ONOS, and Mininet.

Preconditions:

RUNVERIF

The Debian package runverif-20170912_1-amd64.deb must be installed.

Installing this package places the command-line tool runverif in the /bin directory. The tool's version should be 1.0.16 or higher (for a check, use the command-line argument --version).

OVS

OVS needs to be instrumented and configured to report OpenFlow messages to runverif. For this, the provided Debian package openvswitch-switch_2.5.0-1_amd64.deb, which contains a modified version of the OVS daemon ovs-switchd, needs to be installed in addition to the standard Debian packages for OVS. Note that that the standard Debian packages for OVS should be installed prior to installing the modified daemon. You may need to use the command-line argument --force-overwrite for installing the provided Debian Package.

Note that the version 2.5.0 of the OVS daemon was modified. You can

check the version of the OVS daemon with ovs-vswitchd --version. The output should contain information about runverif.

ONOS

ONOS needs to be instrumented and configured to report OpenFlow messages to runverif. For this, the provided Debian package

onos-5gebuild_1.0-1.deb needs to be installed. ONOS is installed into the /opt directory. Furthermore, Java 8 needs to be installed.

MININET

The standard Debian package for Mininet needs to be installed.

#:	Step actions:	Expected Results:
1	RUNVERIF Open a new terminal. In the following, we refer with RV (RunVerif) to this terminal. Start runverif from the RV terminal: RV> runverif --prefix=flowmod We assume here that the current directory contains the three configuration files flowmod.comp, flowmod.msgs, and flowmod.spec. runverif opens the UDP port 50010 and listens on it. Incoming messages are processed by runverif. With	

	RV> runverif --prefix=flowmod --loglevel=3 runverif outputs additional information: [I] 2017/09/13 13:35:00.732019 1 out of 1 CPU core is used. [I] 2017/09/13 13:35:00.736009 Garbage collection target percentage is set to 100. [I] 2017/09/13 13:35:00.750538 2 components are monitored: ovs-vswitchd_s1, onos-out [I] 2017/09/13 13:35:00.804589 Verdicts are with respect to the specification: (FREEZE inport[inport], src[src], dst[dst]. ((NOT SwitchPacketIn(inport, src, dst)) OR (TRUE UNTIL(0s,100ms] ControllerFlowAdd(inport, src, dst)))) [I] 2017/09/13 13:35:00.804651 Monitoring algorithm: mtldata [I] 2017/09/13 13:35:00.831038 UDP socket (port: 50010) is open. When setting the loglevel to 7, runverif also outputs the received messages. Note that runverif's output is also logged in the file /tmp/runverif.log. Note that without the command-line argument --violations, verdicts TRUE are suppressed. A propositional version of the policy is also available: RV> runverif --prefix=flowmod-prop --monitor=mtl The corresponding configuration files are flowmod-prop.comp, flowmod-prop.msgs, and flowmod-prop.spec. Note that here the monitoring algorithm 'mtl' is used, which has a higher throughput. However, the policy is less precise.	
2	OVS The modified OVS daemon is set up as follows by editing the ovs-ctl script in the directory /usr/share/openvswitch/scripts/. 1. Open a new terminal. In the following, we refer with OVS to this terminal. 2. Stop the OVS daemon:	

	OVS> sudo /usr/share/openvswitch/scripts/ovs-ctl stop 3. Modify the script ovs-ctl. Go to the function start_forwarding () and change the line set "$@" --runverif into set "$@" --runverif --runverif-ofp=3 The command-line arguments have the following meaning. --runverif enable the sending of runverif messages --runferiv-ofp=9 report OFPT_PACKET_IN messages (--runverif-globalcounter check if this argument is necessary) For debugging, use the additional command-line argument: --runverif-log It also may be necessary to change some of the following default values. --runverif-host=127.0.0.1 --runverif-port=50010 --runverif-prefix=ovs-switchd 4. Start the OVS daemon: OVS> sudo /usr/share/openvswitch/scripts/ovs-ctl start The log file /var/log/openvswitch/ovs-vswitchd.log for the OVS daemon should contain the information that the connection to runverif is established. 2017-09-13T13:08:01.569Z|00004|runverif|INFO|Opening the monitoring socket (host: 127.0.0.1, port: 50010). 2017-09-13T13:08:01.569Z|00005|runverif|INFO|Connection to monitor established.	
3	ONOS	

Set up the modified version of ONOS as follows.

1. Open a new terminal. In the following, we refer with ONOS to this terminal.

2. To configure ONOS to report OpenFlow messages to runverif, edit the file /opt/onos/apache-karaf-3.0.8/bin/setenv. Set the environment variable ONOS_RUNVERIF_IN to false, ONOS_RUNVERIF_OUT to true, and ONOS_RUNVERIF_MSGS to FLOW_MOD. Further environment variables are ONOS_RUNVERIF_HOST and ONOS_RUNVERIF_PORT. Their default values are 127.0.0.1 and 50010, respectively.

3. Start ONOS:

ONOS> /opt/onos/bin/run-onos.sh

A new terminal should open. For simplicity, we also refer with ONOS to this terminal. You first need to enter the password for obtaining superuser privileges. ONOS will then start. This may take some while.

4. Make sure that the apps org.onosproject.fwd and org.onosproject.openflow are running:

ONOS> apps -s -a

should show

* 20 org.onosproject.optical-model 1.11.0.SNAPSHOT Optical information model

* 45 org.onosproject.drivers 1.11.0.SNAPSHOT Default device drivers

* 68 org.onosproject.hostprovider 1.11.0.SNAPSHOT Host Location Provider

* 69 org.onosproject.openflow-base 1.11.0.SNAPSHOT OpenFlow Provider

* 74 org.onosproject.lldpprovider 1.11.0.SNAPSHOT LLDP Link Provider

* 75 org.onosproject.openflow 1.11.0.SNAPSHOT OpenFlow Meta App

* 85 org.onosproject.fwd 1.11.0.SNAPSHOT Reactive Forwarding App

In case the apps do not appear in the list, start them manually:

	ONOS> app activate org.onosproject.fwd ONOS> app activate org.onosproject.openflow	
4	MININET Open a new terminal. In the following, we refer with MN (MiniNet) to this terminal. Start mininet in the MN terminal: MN> sudo mn --mac --topo single,3 --switch ovs,protocols=OpenFlow10 --controller=remote,ip=127.0.0.1,port=6633 The network comprises the following components: * A controller at the IP address 127.0.0.1:6633. * A single switch that uses the OpenFlow protocol version 1.0. * Three hosts h1, h2, and h3 that are conntected to the switch.	
5	TESTS Now, all components (runverif, OVS, ONOS, and mininet) are running. In particular, when running runverif without the flag --violations, verdicts may appear from time to time on the RV terminal. The reasons are as follows. First, ONOS sends network packets for the network discovery. The switch reacts to these packets by packet-in OpenFlow messages. Runverif is notified by these messages. Second, ONOS adds default flow rules to the switch's flow table. ONOS notifies runverif about the corresponding flow-mod OpenFlow messages. None of these OpenFlow messages should result in verdicts FALSE. When pinging in Mininet a host then this should not result in any policy violation. For example, with MN> h1 ping -c10 h2 the host h1 pings the host h2 for 10 times. Some of the ICMP (and also ARP) network packets, which are received by the switch, will result in packet-in OpenFlow messages that are sent from the switch to the controller. First, for each of the resulting packet-in OpenFlow messages, ONOS should request the installation of corresponding flow rules by sending flow-mod OpenFlow messages to the switch. Second, this should happen within the specified time bound of 100ms. Hence, in the RV terminal, one should only	Output in RV terminal: [V] @1505461416.973250000: true [V] @1505461417.047073000: true [V] @1505461417.047516000: true [V] @1505461417.047618000: true [V] @1505461417.048719000: true [V] @1505461440.296777000: true [V] @1505461440.320839000: true [V] @1505461462.549872000: true [V] @1505461462.575493000: true [V] @1505461529.682289000: true [V] @1505461529.683793000: true [V] @1505461529.683209000: true [V] @1505461529.683557000: true [V] @1505461529.684102000: true

	see verdicts TRUE. Note that a buggy reactive forwarding application may request the installation of a flow rule that does not correspond to a packet-in OpenFlow message. In this case, runverif would detect the policy violation and output the verdict FALSE. Furthermore, note that when changing the policy (flowmod.spec) so that ONOS needs to react within a much shorter time window, say within 10ms, runverif will most likely output verdicts FALSE, since ONOS' reaction time is not anymore within the specified time window. Rerun the test, by restarting all components with the changed policy in flowmod.spec. We remark that for simplicity, we require that ONOS only needs to react to packet-in OpenFlow messages resulting from ICMP network packets, as they are sent by the ping command. Furthermore, these network packets are either of type 0 (HELLO) or type 8 (REPLY). For other network traffic, e.g., HTTPS, one needs to modify the runverif configuration file flowmod.msgs. Recall that this is a very basic scenario with a simple policy to illustrate the concept. Other policies can be checked by modifying the setup and runverif's configuration files.	Note that the timestamps (in Unix time) will differ. They should be the current time of the test. Furthermore, the number might vary. Finally, when running the test with the changed policy (i.e. a time window of 10ms), there should be FALSE verdicts, when pinging the host in Mininet.
Execution type:	Manual	
Estimated exec. duration (min):	45.00	
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Feature-5.2.1: Basic OpenFlow Compliance Checker Use Case 5.3: Reactive Traffic Routing in a Virtualized Core Network Use Case 5.4: Verification of the Virtualized Node and the Virtualization Platform	
Attached files	flowmod-prop.spec README-flowmod flowmod-prop.msgs flowmod-prop.comp flowmod.spec flowmod.msgs flowmod.comp	

Test Case 5ge-139: Deactiviation of SDN network applicatons
Summary: In this scenario, the compliance checker checks whether deactivating a network service is allowed. We restrict ourselves here to

deactivating network applications of the controller ONOS. Concretely, we consider the policy that it is only allowed to deactivate the driver app when the OpenFlow app is not active.

In a broader setting, the network services could be NFVs that for example run in Docker containers. More complex dependencies between services can also be expressed. Furthermore, we could also check that certain network services, when deactivated, must be reactivated within

a specified time window. Another example is that certain network services should not be activated at the same time, e.g., because of conflicting use of network resources.

Preconditions:

RUNVERIF

The Debian package runverif-20170912_1-amd64.deb must be installed.

Installing this package places the command-line tool runverif in the /bin directory. The tool's version should be 1.0.16 or higher (for a check, use the command-line argument --version).

ONOS

ONOS needs to be instrumented and configured to report administrative actions to runverif. For this, the provided Debian package onos-5gebuild_1.0-1.deb needs to be installed. ONOS is installed into the /opt directory. Furthermore, Java 8 needs to be installed.

#:	Step actions:	Expected Results:
1	RUNVERIF Open a new terminal. In the following, we refer with RV (RunVerif) to this terminal. Start runverif from the RV terminal: RV> runverif --prefix=deactivatingapps -monitor=mtl --violations We assume here that the current directory contains the three configuration files deactivingapps.comp, deactivingapps.msgs, and deactivingapps.spec. runverif opens the UDP port 50010 and listens on it. Incoming messages are processed by runverif. To increase the verbosity of runverif's output, the loglevel can be set to 7 (command-line argument --loglevel=7), runverif outputs additional information and the received messages. This can be helpful for debugging. Note that with the command-line argument --violations, verdicts TRUE are suppressed and only violations are reported. When omitting this command-line argument, runverif will also report policy satisfaction. Furthermore, note that runverif's output is also logged in the file /tmp/runverif.log.	
2	ONOS	

	Set up the modified version of ONOS as follows. 1. Open a new terminal. In the following, we refer with ONOS to this terminal. 2. Disable the reporting of OpenFlow messages to runverif. For this, edit the file /opt/onos/apache-karaf-3.0.8/bin/setenv. Set both the environment variable ONOS_RUNVERIF_IN and ONOS_RUNVERIF_OUT to false. 3. Start ONOS: ONOS> /opt/onos/bin/run-onos.sh -q -p deactivatingapps.proxy We assume that the configuration file deactivatingapps.proxy is contained in the current directory. A new terminal should open. For simplicity, we also refer with ONOS to this terminal. You first need to enter the password for obtaining superuser privileges. ONOS will then start in this terminal. This might take some time. Already during the start of ONOS, ONOS will send messages to the runverif. The host IP address and the port are 127.0.0.1 and 50010 to which these messages are sent. This can be changed in the script /opt/ono/bin/run-onos.sh by editing the variables RUNVERIF_PORT and RUNVERIF_HOST. When changing the port, we note that runverif must be started so that it listens on this port. To increase the verbosity (i.e., the messages sent to runverif), omit the quiet flag -q.	
3	TESTS Make sure that the apps org.onosproject.openflow and org.onosproject.drivers are activated. To output the list of active apps use ONOS> apps -s -a If, e.g., org.onosproject.openflow is not active, activate it ONOS> app activate org.onosproject.openflow Both these apps should be automatically be activated during the start up of ONOS. We remark that sometimes ONOS behaves strangely here. In particular, activating and deactivating apps fails sometimes, with no obvious reason.	Output in RV terminal: [V] @1505459831.848781000: false Note that the timestamp (in Unix time) will differ. It should be the current time.

	Depending on the order of activating and deactivating certain apps, runverif will report policy violations. For instance, with ONOS> app deactivate org.onosproject.openflow ONOS> app deactivate org.onosproject.drivers runverif outputs in the RV terminal the verdict FALSE. Note that for the test it is not necessary to connect ONOS to data plane devices. Nevertheless, by running > sudo mn --mac --topo single,3 --switch ovs,protocols=OpenFlow10 --controller=remote,ip=127.0.0.1,port=6633 from a terminal, we can connect ONOS to a simple network with a single switch connected to three hosts.	
Execution type:	Manual	
Estimated exec. duration (min):	45.00	
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Use Case 5.4: Verification of the Virtualized Node and the Virtualization Platform Feature-5.2.2: Basic NFV Reconfiguration Compliance Checker	
Attached files	README-deactivatingapps deactivatingapps.spec deactivatingapps.proxy deactivatingapps.msgs deactivatingapps.comp	

T_UC5.2_1 Add malicious nodes into core network

Description: Detailed description of threat and its importance	Malicious nodes may e.g. eavesdrop, tamper, and prevent data flows.

Category: ITU-T X.805 security dimension(s)	Access control; Authentication; Non-repudiation; Data confidentiality; Communication security; Data integrity; Availability; Privacy
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	Confidentiality, integrity and availability of e2e communication are compromised.
Possible Mitigation Hints (if known): How can we protect against the threat?	Applying security verification procedures – technical and organisational - for assuring that the added nodes are trustworthy. Only authenticated and authorized entities should be allowed to add nodes. Security monitoring of behaviour of added nodes as well as communication over the network.
Entry Points (if known): What possible means does an adversary have?	Software, image used for deploying new nodes may be compromised. Forwarding logic may be misconfigured so that illegitimate node, switch is able to get access to data flows. In this case, the malicious node is unintentionally added to the core network.

Test Case 5ge-25: Authentication to a micro-segment
Summary: The objective of this test is to check how the micro-segmentation enabler is able to respond to the threat T_UC5.2_1 Add malicious nodes into core network. In this threat malicious nodes may e.g. eavesdrop, tamper, and prevent data flows. The enabler applies security verification procedures, namely IEEE 802.1X based authentication for assuring that the added nodes are trustworthy. This test presumes that the single node version of the enabler has been installed.
Preconditions: Microsegmentation enabler has been deployed on the testbed. The configuration is been done according to the testbed chosen setup. The single node version of the enabler has been installed. sudo login to the test computer required.

Unit Tests 1-4 of the microsegmentation enabler have been successful. Modify the file wpasupplicant-mno01.conf in $HOME/OpenVirteX/scripts/ensure. Change the default password to something else.		
#:	Step actions:	Expected Results:
1	Start the enabler with the start_screen.sh command in directory $HOME/OpenVirteX/scripts/ensure. cd $HOME/OpenVirteX/scripts/ensure ./start_screen.sh	The enabler is started and no error is reported. The start_screen.sh command should generate several windows.
2	Change to screen window 1 (CTRL+A 1) and give your password for sudo command. Next same for screen window 2 (CTRL+A 2). After waiting for a while (5-10s), navigate to screen window 5 (OVX_creation) (by pressing CTRL+A 5) and then press ENTER/RETURN	No error should be reported. Virtual ports and virtual links should be created. If error messages come, the test should be started from the beginning by exiting all screen windows.
3	To test authentication of a possibly malicious node to the first micro-segment, navigate to window 1 (CTRL+A 1, mininet) and execute the following commands: remote ./ensure_test.sh br-ensure port; wpa_supplicant -i tap-ensure -Dwired -c wpasupplicant-mno01.conf	Host will not be authenticated to the micro-segment.
4	In window 1 (mininet), test the connection by the following command: remote ping 192.168.33.1	The ping should not work as the node has not been authenticated to the micro-segment.
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Use Case 5.2: Adding a 5G Node to a Virtualized Core Network Feature-5.4.1: Dynamic Arrangement of Micro-Segments	

Test Case 5ge-93: Malicious enclave don't get key
Summary: A malicious or compromised enclave should not be added to the network. This means that if the actual measurement of the application is not in the list of expected hashes, the application should not be provisioned with a key and the network can thus not connect to the SDN controller.
Preconditions: The verification manager software is installed on VM1. The remote host software is installed on VM2. The (malicious) application (ApplicationEvil) is also installed on host VM2.

The following files should contain on each VM the IP of the other VM host. For instance for VM1: /opt/bootstrappingtrust/Certs/rh_host (should have IP address of VM2) and /opt/bootstrappingtrust/Certs/container_host (should have IP address of VM2)		
#:	Step actions:	Expected Results:
1	Launch the remote host software on VM2. cd /opt/bootstrappingtrust/RemoteHost sudo ./app	It is launched and awaits connections.
2	Launch the malicious application in a new window on VM2. cd /opt/bootstrappingtrust/ApplicationEvil sudo ./app	It is launched and awaits connections.
3	Launch the verification manager on VM1, which will then immediately try to connect and try to attest the integrity of the (malicious) application. cd /opt/bootstrappingtrust/VerificationManager ./app	The verification manager will not provision the malicious enclave with a key, as it detects that the measurement of the remote application is not an expected value. This is given by the message "ERROR: Actual MRENCLAVE hash is not in the list of allowed hashes!" from the final lines of the output of the verification manager. If the previous statement is true, then this test is successful.
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Use Case 5.2: Adding a 5G Node to a Virtualized Core Network Feature-5.3.1: Integrity Attestation of Virtual Network Components	

T_UC5.2_2 Forwarding logic leakage

Description: Detailed description of threat and its importance	A network application running on the controller is able to see the forwarding logic of another application (i.e.: the OpenFlow rules installed in the switches). The applications can belong to different virtual network operators who do not want to leaking sensitive information about how their virtual nodes are located or migrated. The leakage can happen in two directions. Controller-to-switch contains rules that have been installed in the switches. A malicious application can not only intercept the OpenFlow messages as they are sent, it can also request information from the switch about installed rules and related statistics belonging to other applications. Eavesdropping on switch-to-controller (e.g.: OFPT_PACKET_IN) messages can also leak information not only about the forwarding logic, but about application data that might be confidential.
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	Information about forwarding logic is leaked: positioning of network elements like DNS or other services provided through VNFs and how they are migrated which can be used to infer user population, reliability information etc.

Possible Mitigation Hints (optional, if foreseen): How can we protect against the threat?	Insert a reference monitor at the southbound interface.
Entry Points (optional, if known): What possible means does an adversary have?	Deploy an application on the controller in a multi-tenant virtualized network.
5G-ENSURE enablers (optional, if covered for given threat): What possible means does an adversary have?	Enabler 6.2 “Access Control Mechanisms”

Test Case 5ge-95: TLS connection to controller		
Summary: Ensures that a TLS connection is setup between the Application and the Controller, after a successful provisioning of the Application. This makes the communication between the application and the controller both integrity and confidentiality protected.		
Preconditions: The verification manager software is installed on VM1. The remote host software is installed on VM2. The (malicious) application (ApplicationEvil) is also installed on host VM2. The following files should contain on each VM the IP of the other VM host. For instance for VM1: /opt/bootstrappingtrust/Certs/rh_host (should have IP address of VM2) and /opt/bootstrappingtrust/Certs/container_host (should have IP address of VM2)		
#:	Step actions:	Expected Results:
1	Launch tcpdump and start capturing traffic on the application host (VM2). sudo tshark -i lo -f "port 8081" -o http.ssl.port:8081	Tshark starts the capture.
2	Launch the remote host software on VM2. cd /opt/bootstrappingtrust/RemoteHost sudo ./app	It is launched and awaits connections.
3	Launch the benign application on VM2. cd /opt/bootstrappingtrust/Application sudo ./app	The application starts, and awaits connections from the verification manager.
4	Launch the Floodlight SDN controller on VM2, separately from the application. cd /opt/Floodlight sudo java -jar target/floodlight.jar	Floodlight starts.

5	Launch the verification manager on VM1, which will then immedately connect and attest the application. cd /opt/bootstrappingtrust/VerificationManager ./app	The verification manager will provision the enclave with a key. No lines in the output of the vericiation manager will start with "ERROR". Switching to the output from VM2 and the application the following output should be visible near the end, since this indicates that communcation was successful with the floodlight controller. HTTP/1.1 200 OK Content-Type: application/json Date: Tue, 13 Jun 2017 13:37:00 GMT Accept-Ranges: byte Server: Restlet-Framework/2.3.1 Vary: Accept-Charset, Accept-Encoding, Accept-Language, Accept Connection: close 37 bytes read {"name":"floodlight","version":"1.2"} EOF
6	Stop the tshark capture (ctrl-c) and investigate the output.	A TLS session, with communcation, can be seen between the application and the controller on port 8081. The output should contain lines similar to: TLSv1.2 134 Application Data which shows that a TLS session has been established and that encrypted data is sent. If this is the case, then this test is successful.
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	**Medium**	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Use Case 5.2: Adding a 5G Node to a Virtualized Core Network Feature-5.3.1: Integrity Attestation of Virtual Network Components	

T_UC5.5_1 Misuse of open control and monitoring interfaces

Description: Detailed description of threat and its importance	Third-party service providers may misuse the access to control and monitoring interfaces and cause service disruptions for the operator or attack against data flows. For instance, monitoring information on flowing data may be captured in order to profile end-users. While interfaces are opened for service providers they may also become available for other adversaries.
Potential effect: What global effect it will have on	Resources and user data become available for larger amount of parties. More trusted parties means that there may be parties that do not provide good enough security and follow good security practises.

major 5G system domains (network, hosts, applications, e2e effect…)	
Possible Mitigation Hints (if known): How can we protect against the threat?	Service providers should be required to protect the monitoring data they acquire. Service providers should protect their own resources sufficiently, so that adversary cannot access slices through service providers' systems. Strong isolation is needed to prevent service providers from accessing resource outside a slice. Service providers should be allowed to access only those control interfaces that are required to minimize service providers potential to escape
Entry Points (if known): What possible means does an adversary have?	Control interfaces can be enable access to operator's functions either directly (if not sufficient fine-grained protection is available) or the interfaces may contain vulnerabilities that may be utilized to gain additional privileges. A service provider itself may be untrustworthy. Alternatively, an adversary may compromise service providers systems in order to gain access to the slice.

Test Case 5ge-128: Monitoring access control misuse in a mobile network

Summary:

The System Security Threat Repository (SSSR) makes use of a knowledgebase encoding information about the assets, trust relationships, threats and controls in the 5G architecture. This knowledgebase is used to addresses the need to enrich the system view with information about the system's assets, the threats, incidents, and analysis results in order to understand the state of the whole system. The enabler allows querying and analysis for a higher-level view of security incidents and trends.

See attached PDF for detailed description and screenshots. Sample mobile network model for trust builder provided as well

Preconditions:

The following steps are required to run SSSR:

Trust Builder from the SSSR distribution package is installed and a sample model is loaded as described in section 2.3.1 of "T34 R2 SSSR.pdf" in the attachment

SSSR is configured to use the sample model loaded in Trust Builder as described in section 2.3.2 of "T34 R2 SSSR.pdf" in the attachment

#:	Step actions:	Expected Results:
1	In Trust Builder, import sample mobile network model and select it	Mobile network model is presented to the user (see Figure 3. Sample mobile network as a Design-Time model in Trust Builder)
2	Ensure the same model is loaded in SSSR (http://localhost:3020)	Mobile network model is visible in SSSR interface (see Figure 4. Graphical representation of non-compliance analysis results in SSSR)
3	Whilst toggling Access Control switch in GCI Client interface (http://localhost:3001), ensure that GCI interface (http://localhost:3010) is updated with valid report on every change	See section 2.2 Simulating asset non-compliance
4	Repeat previous step with SSSR instead GCI. "Latest GCI report" should update on any Access Control change. Non-compliant assets are visualised in "Trust Builder Model" and "Compliance analysis" tabs	See sections 2.3 and 2.4

Execution type:	Manual
Estimated exec. duration (min):	
Priority:	Medium
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**
Requirements	Use Case 5.5: Control and Monitoring of Slice by Service Provider Feature-4.5.2: System Security State Repository service
Attached files	T34 R2 SSSR : T34_SSSR_sml.pdf Trust Builder Mobile Network Model : Model_for_SSSR_validated.nq T_UC5.5_1 - Misuse of open control and monitoring interfaces : T_UC5.5_1 Misuse of open control and monitoring interfaces sml.pdf

T_UC5.5_4 No control of Cyber-attacks by the Service providers

Description: Detailed description of threat and its importance	The use case features a Service Provider (SP) offering its Massively Multiplayer Online Game service to gamers. The Service Provider buys its network service to Virtual Mobile Network Operator (VMNO) which itself relies on an Infrastructure Provider. The VMNO supplies a sub-slice to the SP with the required QoS. The service of the SP is subject to cyber-attacks. The SP wants to manage the cyber-security of its service. It signs a contract with a third party Security Service Operator (SSO) to monitor and remediate to cyber-security attacks. Thanks to the terms of the contract between the SP and the VMNO, the SSO can benefit from network topology information and routing tables from the slice controller. Nevertheless, since it has not the information about the configuration of the NVF and their vulnerabilities, it cannot build a classical attack graph to monitor the cyber-attacks.
Potential effect: What effect it will have on 5G system (network, hosts, applications…)	The Service Provider has no control over the cyber-attacks on its slice.
Possible Mitigation Hints (if known): How can we protect against the threat?	A possible mitigation hint would be to enable the SSO to get access to the information from the infrastructure domain, especially the type of software used for NVF in order to establish the vulnerabilities of it. Another way to mitigate this is to separate the responsibilities by contract between the infrastructure domain and the VMNO. The SP will have to rely on the VMNO interface and will only control its cyber-threats at application level.
Entry Points (if known): What possible means does an adversary have?	An adversary could attack the VNFs, hypervisor or orchestrator of the Infrastructure Provider to compromise the Service Provider's service.

Test Case 5ge-108: Two types of security control for service provider

Summary:

This test scenario demonstrates how three enablers - micro-segmentation, security monitor for 5G microsegments, and trust metric enabler - provide more control over the cyber attacks for service providers that using are the 5G network. The case demonstrates how service providers can be delivered coarse or fine-grained security and trust information from the 5G network (segment) that has been dedicated for the service provider. The case also illustrates that, when the control and monitoring APIs to 5G network are opened, service provider are able to get custom security functionality to 5G networks (to microsegments).

In this test case, the service provider gets futher availability quarantees as a machine learning algorithm for anomaly detection is analysing network flows (and able to quarantine flows from suspected DoS attacks). Further, a status notifications on the real-time trust situation (based e.g. anomaly detection and availability of security services in the micro-segment) is delivered to the service provider.

In this scenario, the service provider is given two types of alternative security controls:

1) Coarse-grained: Trust Metric enabler provides real time information to the service provider about the security level of the segmented network, i.e., a micro-segment. (Coarse grained information does not disclose information that is sensitive for the operator or other clients/service providers). The service provider may use this information during orchestration, when deciding whether the network offered by the operator can be trusted or not.

2) Fine-grained - Security Monitor for 5G Micro-Segments enabler provides observation/reaction algorithms to the network. (Fine-grained information is available, if the network operator wants to pass this information forward. The operator may also agree with the service provider on the customization of the monitoring algorithms.)

The security control is enabled by the micro-segmentation enabler, which segments the network so that the service provider is able to retrieve information from it and control it (in cooperation with the operator without disturbing traffic flows of other services providers). (Microsegmentation removes also some legal / privacy obstacles from sharing of monitoring information as monitoring can focus to segmented flows originating to the service provider. Hhence information belonging to other customers of operator are not disclosed).

The purpose of the test case is to show that

A) enablers are starting and running

B) one security monitoring instance is running focusing on the micro-segment (this enabler is running monitoring and control algorithms preferred by the service providers)

C) Trust metric enabler shows to the service provider how secure / trusted the micro-segment is.

For further information on the enablers, please see open specifications and user guides.

Preconditions:

Security Monitor for 5G Micro-Segments enabler (R2) has been installed the virtual machine A (VM A)

Trust Metric enabler (R2) has been installed on the same machine than Security Monitor for 5G Micro-Segments enabler (VM A)

Micro-segmentation enabler (R2) has been installed to different virtual machine (VM B) than those two other enablers

#:	Step actions:	Expected Results:
1	Start a script that will create the micro-segment as well as start the trust metric enabler and the framework for the security monitoring. **VM B:** *cd $HOME/OpenVirteX/scripts/ensure* *./change_mininet_mode.sh badguys* *./start_screen.sh*	The scripts creates several shell windows using screen utility.

	VM A: *cd $HOME/OpenVirteX/scripts/ensure* *./start_msme.sh*	
2	Create a virtual network **VM B:** After waiting for a while (5-10s), navigate to screen window 5: OVX_creation (by pressing CTRL+A 5) and then press ENTER/RETURN. (This will execute script: *./create_microsegments.sh*)	
3	**VM B:** Authenticate a node to the microsegment using mininet (network simulator) Navigate to screen 1: mininet (by pressign CTRL+A 1) Hit enter to execute: *remote ./ensure_test.sh br-ensure port; wpa_supplicant -i tap-ensure -Dwired -c wpasupplicant-mno01.conf* Afterwards, "tap-ensure: CTRL-EVENT-CONNECTED" should be shown on the screen and you can press “CTRL-C”.	
4	**VM A:** Start feeding monitoring information from the micro-segment to security monitor. Please check that /usr/local/etc/msme_config will include the correct IP address of VM B mgmt interface in variable *mse_ip* Navigate to screen 0: mse2kafka (by pressign CTRL+A 0) Hit enter to execute: *python -m microsegmentmonitoring/adapters/ensure_websocket_kafka 2>/dev/null*	
5	**VM A:** Start security monitor (with anomaly detection). Wait several seconds after the previous task. Navigate to screen 1: MSME (by pressign CTRL+A 1) Hit enter to execute: */usr/local/src/spark-1.6.2-bin-hadoop2.6/bin/spark-submit --packages org.apache.spark:spark-streaming-kafka_2.10:1.6.1 /usr/local/lib/python2.7/dist-packages/microsegmentmonitoring/msme_spark/ms_cep.py localhost:9092 localhost:8888 localhost:44444 192.168.33.1 1 2>&1 | grep -v 'INFO'*	
6	**VM B:** Test microsegment connection:	Successful ping events should start emerging to the screen.

	Navigate to screen 1: mininet (by pressign CTRL+A 1) *remote ping 192.168.33.1* Stop the ping by pressing CTRL-C and then give the following command: *bad1 hping3 -V -c 1000 -d 120 -S -w 64 -p 21 192.168.33.1*	
7	**VM A:** Observe that the security monitor is able to collect security data from the microsegment: Navigate back to screen 1: MSME (by pressign CTRL+A 1)	The monitor is able to see the ping traffic we generated in the previous step. The monitor will output status information when it sees new events. It will output either anomaly or not anomaly. In the anomaly case, the enabler will instruct microsegmentation enabler to block the traffic from the node. Whether node is quarantined or not is visible from the screen 1 (unquarantined node continues to produce new ping lines). The expected output looks something like this: --- Time: 2017-09-27 22:37:30 --- [1740.0,10.0,3232243969.0,3232244168.0,7.0] AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA anomaly: feature vector 2.08948034204e+19 > 1.0 handling anomaly in flow 192.168.33.1 -> 192.168.33.200 src mac: 00:00:00:00:00:07 send kafka localhost:9092 msidx.anomality_level {'msidx.anomality_level': 2} QuarantineAction 00:00:00:00:00:07 send kafka localhost:9092 msidx.quarantine_node {'msidx.quarantine_node': '00:00:00:00:00:07'} send kafka localhost:9092 msidx.anomality_level {'msidx.anomality_level': 0} Note, also 'non-anomaly' is an accepted result. The test case shows that the service provider gets awareness through the selected machine learning algorithm. Whether, the monitor interprets the ping events as anomaly or not, depends on the time when ping was executed. If the ping occurs soon - while the monitor is still learning what is the normal behaviour in the network - it is not considered as an anomaly.
8	**VM B:** Check that hping3 command stops printing any valid output when the previous task 8 prints out anomaly and Quarantine Action.	
9	**VM A:** Observe trust metrics: Navigate to Screen 6: TrustClient (by pressign CTRL+A 6)	Trust metric client will display what trust policies it has requested and how trust metric enabler is answering (are policies matched by the micro-segment or not). The expected output looks something like this:

	Trust metric client (i.e. the service provider) was launched in Step 1 and is running here.	send {'services': {'msidx.securitymonitoringenabler': 1, 'msidx.policycompliancechecker': 1}, 'policyid': 2, 'maxlevels': {'msidx.authcounter.MD5': 50, 'msidx.anomality_level': 1}} send {'services': {'msidx.securitymonitoringenabler': 1}, 'policyid': 1, 'maxlevels': {'msidx.authcounter.MD5': 0}} send {'services': {'msidx.securitymonitoringenabler': 1, 'msidx.policycompliancechecker': 1}, 'policyid': 3, 'maxlevels': {'msidx.authcounter.MD5': 10, 'msidx.anomality_level': 0}} availability related trust metric: False privacy related trust metric: False high availability related trust metric: False privacy related trust metric: True privacy related trust metric: True
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**4 - Testbed evaluation (real flows)**	
Requirements	Feature-3.2.1: Trust metric based network domain security policy management Feature-4.4.1: Complex Event Processing Framework for Security Monitoring and Inferencing Use Case 5.5: Control and Monitoring of Slice by Service Provider Feature-5.4.1: Dynamic Arrangement of Micro-Segments	

T_UC5.6_1 Security threats in a satellite network

Description: Detailed description of threat and its importance	Security client-side agents are deployed over the satellite network components in order to periodically collect information related to the security dimensions. Once registered, these components deliver to the security monitoring (server-side) the compiled information. This information is supervised in the security monitor that carry out a security analysis to detect attacks and malicious behaviour. The origin of most fraudulent accesses or security breaches can be summarized as either technical identity alteration (after an illegal or illegitimate privilege augmentation) or signalling messages received outside of the normal sequences. These systems are exposed to new threats in 5G that must be mitigated. …). Some of the threats identified are: Attack on network components: RF interference, power or communications lines… Attack on the network management system: intruding the system by hijacking, blackmailing, placing or impersonating the operator, to obtain credentials or/and gain control of the system… Denial of service: flood the network with dummy indicators to make the network unusable, preventing any useful communications with the network management system.

Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	The security properties that this threat can compromise are: Service availability Outages Information confidentiality
Possible Mitigation Hints (optional, if foreseen): How can we protect against the threat?	System can be protected against these threats acting on three levels: Client-side: Generic secure interface to provide indicators from a heterogeneous network. Server-side: Data analytics and intelligence-driven security to detect threats based on security metrics. Network-side: Partitioning the satellite network into virtual private networks.
Entry Points (optional, if known): What possible means does an adversary have?	Heterogeneous networks (satellite and terrestrial) which components are geographically widespread distributed. Some of these network components (e.g. eNBs) are outside the MNO facilities and even on the customer's premises (e.g. satellite device).
5G-ENSURE enablers (optional, if covered for given threat): What possible means does an adversary have?	Satellite Network Monitoring

Test Case 5ge-133: Unauthorised user athentication

Summary:

In this test case, it is going to be tested all the policy rules setted in the policy file. For this an user registered but with other role, will try to access from a different country in a different time that the allowed.

The response of the test, should be that the user is not allowed to perform the operation because the conditions of the petition does not match with the rules applied in the policy.

Conditions:

- The user is registered in the LDAP server and the role does not match with the role declared in the policy file.

- The time when the user is trying to make the petition is out of the range 08:00-18:00

- The location from where the user is trying the connection is outside Spain.

To simulate the above conditions, the policy file in the server can be modified, just for a verification of the conditions.

Preconditions:

Execute the 5g-54 in order to preload the environment.

#:	Step actions:	Expected Results:

1	$ curl -v -X POST -H "Content-Type: application/xml" -H "Accept: application/json" --data "@UT01/input/TestPolicy_UT01b.xml" http://5g-fga-sat-srv01.5g-ensure.eu:8080/fga-sat-srv/api/v01.00.00/pap/policy	{"header":{"responseCode":0,"msgType":"restResponseMC"},"content":{}}
2	$ curl -v -X GET -H "Accept: application/json" http://5g-fga-sat-srv01.5g-ensure.eu:8080/fga-sat-srv/api/v01.00.00/pap/policies/TestPolicy_UT01b	{"header":{"responseCode":0,"msgType":"restResponseMC"},"content":"<XACML policy>”}
3	$ curl -v -X POST -H "Authorization: 5G-ENSURE base64(Q2hyaXN0b3BoZXIgQ2Fycm9sbA==:Y2hDQV81Zw==)" -H "Content-Type: application/json" -H "Accept: application/json" --data "@UT03/input/RequestContent_UT03b.json" http://5g-fga-sat-srv01.5g-ensure.eu:8080/fga-sat-srv/api/v01.00.00/pdp/authorize	{"header":{"responseCode":0,"msgType":"restResponseMC"},"content":{}}
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Relations	depends on - 5ge-54:Installing and configure environment related to - 5ge-137:Authorised user authentication	
Requirements	Use Case 5.6: Integrated Satellite and Terrestrial Systems Monitor Feature-1.2.3: AAA integration with satellite systems	

Test Case 5ge-137: Authorised user authentication

Summary:

In this test case, it is going to be tested all the policy rules setted in the policy file. For this an user registered, will try to access from a the country declared in the policy in a the time allowed.

The response of the test, should be that the user is allowed to perform the operation because the conditions of the petition does not match with the rules applied in the policy.

Conditions:

- The user is registered in the LDAP server and the role does match with the role declared in the policy file.

- The time when the user is trying to make the petition is in the range 08:00-18:00

- The location from where the user is trying the connection is in Spain.

To simulate the above conditions, the policy file in the server can be modified, just for a verification of the conditions.

Preconditions: Execute the 5g-54 in order to preload the environment.		
#:	Step actions:	Expected Results:
1	$ curl -v -X POST -H "Content-Type: application/xml" -H "Accept: application/json" --data "@UT01/input/TestPolicy_UT01b.xml" http://5g-fga-sat-srv01.5g-ensure.eu:8080/fga-sat-srv/api/v01.00.00/pap/policy	{"header":{"responseCode":0,"msgType":"restResponseMC"},"content":{}}
2	$ curl -v -X GET -H "Accept: application/json" http://5g-fga-sat-srv01.5g-ensure.eu:8080/fga-sat-srv/api/v01.00.00/pap/policies/TestPolicy_UT01b	{"header":{"responseCode":0,"msgType":"restResponseMC"},"content":"<XACML policy>”}
3	$ curl -v -X POST -H "Authorization: 5G-ENSURE base64(Q2hyaXN0b3BoZXIgQ2Fycm9sbA==:Y2hDQV81Zw==)" -H "Content-Type: application/json" -H "Accept: application/json" --data "@UT03/input/RequestContent_UT03b.json" http://5g-fga-sat-srv01.5g-ensure.eu:8080/fga-sat-srv/api/v01.00.00/pdp/authorize	{"header":{"responseCode":0,"msgType":"restResponseMC"},"content":{}}
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Relations	depends on - 5ge-54:Installing and configure environment related to - 5ge-133:Unauthorised user athentication	
Requirements	Use Case 5.6: Integrated Satellite and Terrestrial Systems Monitor Feature-1.2.3: AAA integration with satellite systems	

Use Cases cluster 8 - Ultra-Reliable and Standalone Operations

T_UC8.1_1 Service failure over satellite capable eNB

Description: Detailed description of threat and its importance	Main threats that may cause a service failure are related to the following activities: Failures or malfunctions: Failure or disruption of communication links Failure or disruption of main supply Failure or disruption of service providers Malfunction of equipment Outages: Network connectivity Loss of physical resources Support services (Internet provider or Electricity provider) Disasters: Natural disasters Environmental disaster Physical attacks: Sabotage Vandalism Terrorists attack A Service Provider (i.e. telecommunications company) has a contract with the Satellite Network Operator (SatNO) to supply a suitable system capacity with some QoS guarantees to be used by its customers. Therefore, the Service Provider has to ensure that the SatNO is providing what is required by the contract (SLA). This threat is particularly acute in ultra-reliable services (i.e. e-health, lifeline communications, military scenarios…).
EO interpretation of the threat:	Accidental or deliberate link failures or traffic congestion may comprise the service availability and should be mitigated reconfiguring the transport network topology.
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	Service availability or traffic congestion
Possible Mitigation Hints	Allowing the Service Provider to have some degree of control over their micro-slice or sub network enabling

(optional, if foreseen): How can we protect against the threat?	dynamic allocations and network reconfigurations on the fly. Evolving the Transport Network Architecture (TNA) by combining both satellite and terrestrial transport architectures. Once a link failure has been detected, new topology is forwarded to base stations with satellite links and smart antennas, enabling topology reconfiguration according to traffic failures and traffic demands.
Entry Points (optional, if known): What possible means does an adversary have?	4G backhaul networks are fixed topologies, therefore the network barely manages accidental/deliberate link failures or traffic congestion. An exhaustive radio planning is needed before base station deployment and new backhaul nodes cannot be easily added.
5G-ENSURE enablers (optional, if covered for given threat): What possible means does an adversary have?	Once a link failure/congestion is detected, Satellite Network Monitoring provides a Topology algorithm to reconfigure the network components.

Test Case 5ge-132: Reconfigure the network topology

Summary:

Checks that the user can configure the security/performance indicators to be collected.
Checks that the updated topology may be forwarded.

The initial topology is configured in step #8 and can be checked in steps #9, #10 and #11.

http://10.102.0.51/lib/attachments/attachmentdownload.php?id=111

The indicators to be collected are configured in step #12. Node 5g-enodeb3 is configured with $MON_SAT_PATH/test/UT01/input/indicators_UT01.5g-enodeb3.json:

ifOperStatus from terrestrial terminal 1.

ifOperStatus from terrestrial terminal 2.

ifOperStatus from satellite terminal 1.

Each node sents the operational state of the interface (ifOperStatus) to the satellite-network-monitoring-server every 10 seconds (snmp_retry_timeout_msg property in $MON_SAT_PATH/client/SatelliteNetworkMonitoringClient.properties). If the operational state of the interface is set to down ("error_value": 2) the node sends an alarm message.

Link failure is emulated in step #13.

The incident/failure is detected in step #14. The satellite-network-monitoring-server is continuosly collecting messages from the message broker (i.e. ActiveMQ). When the SatelliteNetworkMonitoringServer detecs an alarm message (messageType field in the header set to "alarm") it launches the Topology Manager (see "apply" trace in $MON_SAT_PATH/logs/monitoring.log).

The Topology Manager calculates the best topology that fixes the issue based on two KPIs:

Similarity (the final topology should be similar as the original one).

TotalPowerConsumed (the lower the better).

Later, this topology is forwarded to all the nodes.

The final topology can be checked in steps #15, #16 and #17.

http://10.102.0.51/lib/attachments/attachmentdownload.php?id=112

#:	Step actions:	Expected Results:
1	ssh admin5g@<5ge-satellite-network-monitoring-server> pkill -f SatelliteNetworkMonitoring rm -f $MON_SAT_PATH/logs/* psql -f $MON_SAT_PATH/server/clean.sql snm admin5g	The server app has been initialized
2	ssh admin5g@<5ge-satellite-network-monitoring-server> $MON_SAT_PATH/apache-activemq/bin/activemq restart $MON_SAT_PATH/server/monitoring.sh	The server environment has been started up
3	ssh root5g@<5ge-satellite-network-monitoring-client> $MON_SAT_PATH/client/updateIP.sh docker-compose -f $MON_SAT_PATH/client/docker-compose/docker-compose.yml down -v pkill -f SatelliteNetworkMonitoring rm -f $MON_SAT_PATH/logs/* rm -f $MON_SAT_PATH/client/indicators/* rm -f $MON_SAT_PATH/client/topologies/*	The client app has been initialized
4	ssh root5g@<5ge-satellite-network-monitoring-client> docker-compose -f $MON_SAT_PATH/client/docker-compose/docker-compose.yml up -d	The server environment has been started up
5	ssh root5g@<5ge-satellite-network-monitoring-client> docker exec epc_tt1 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh docker exec epc_st1 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh docker exec epc_tt1 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh docker exec 5g-enodeb1_tt1 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh docker exec 5g-enodeb1_tt2	The SNMP simulators have been started up

	/mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh docker exec 5g-enodeb1_tt3 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh docker exec 5g-enodeb2_tt1 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh docker exec 5g-enodeb2_tt2 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh docker exec 5g-enodeb3_tt1 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh docker exec 5g-enodeb3_tt2 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh docker exec 5g-enodeb3_st1 /mnt/SatelliteNetworkMonitoring/terminal/snmpsim.sh	
6	ssh root5g@<5ge-satellite-network-monitoring-client> docker exec epc /mnt/SatelliteNetworkMonitoring/5g-enodeb/deploy.sh docker exec 5g-enodeb1 /mnt/SatelliteNetworkMonitoring/5g-enodeb/deploy.sh docker exec 5g-enodeb2 /mnt/SatelliteNetworkMonitoring/5g-enodeb/deploy.sh docker exec 5g-enodeb3 /mnt/SatelliteNetworkMonitoring/5g-enodeb/deploy.sh	The client SW has been deployed
7	ssh root5g@<5ge-satellite-network-monitoring-client> docker exec epc /root/SatelliteNetworkMonitoring/client/startSnmpManager.sh docker exec 5g-enodeb1 /root/SatelliteNetworkMonitoring/client/startSnmpManager.sh docker exec 5g-enodeb2 /root/SatelliteNetworkMonitoring/client/startSnmpManager.sh docker exec 5g-enodeb3 /root/SatelliteNetworkMonitoring/client/startSnmpManager.sh	The SNMP clients have been started up
8	ssh root5g@<5ge-satellite-network-monitoring-server> cd $MON_SAT_PATH/test curl -v -X POST -H "Content-Type: application/json" -H "Accept: application/json" --data	The initial topoloy has been deployed in all the nodes {"header":{"responseCode":0,"msgType":"restResponseMC"},"content":{}}

	"@UT02/input/topology_UT02.epc.json" http://epc:8080/mon-sat-cli/api/v01.00.00/sna/resource/topology curl -v -X POST -H "Content-Type: application/json" -H "Accept: application/json" --data "@UT02/input/topology_UT02.5g-enodeb1.json" http://5g-enodeb1:8080/mon-sat-cli/api/v01.00.00/sna/resource/topology curl -v -X POST -H "Content-Type: application/json" -H "Accept: application/json" --data "@UT02/input/topology_UT02.5g-enodeb2.json" http://5g-enodeb2:8080/mon-sat-cli/api/v01.00.00/sna/resource/topology curl -v -X POST -H "Content-Type: application/json" -H "Accept: application/json" --data "@UT02/input/topology_UT02.5g-enodeb3.json" http://5g-enodeb3:8080/mon-sat-cli/api/v01.00.00/sna/resource/topology	

9	curl -X GET -H "Content-Type: application/json" -H "Accept: application/json" http://5g-enodeb1:8080/mon-sat-cli/api/v01.00.00/sna/resource/topology	In 5g-enodeb1, all the terrestrial terminals are power on
		
		{"header":{"responseCode":0,"msgType":"restResponseMC"},"content":[
		{
		"node": "tt_1",
		"ip": "172.18.1.1",
		"enabled": "true",
		"status": "MANDATORY_ON"
		},
		{
		"node": "tt_2",
		"ip": "172.18.1.2",
		"enabled": "true",
		"status": "MANDATORY_ON"
		},
		{
		"node": "tt_3",
		"ip": "172.18.1.3",
		"enabled": "true",
		"status": "MANDATORY_ON"
		}
		]}

10	curl -X GET -H "Content-Type: application/json" -H "Accept: application/json" http://5g-enodeb2:8080/mon-sat-cli/api/v01.00.00/sna/resource/topology	In 5g-enodeb2, the terrestrial terminal #1 is power on and the terrestrial terminal #2 is power off
		
		{"header":{"responseCode":0,"msgType":"restResponseMC"},"content":[
		{
		"node": "tt_1",
		"ip": "172.18.2.1",
		"enabled": "true",
		"status": "MANDATORY_ON"
		},
		{
		"node": "tt_2",
		"ip": "172.18.2.2",
		"enabled": "true",
		"status": "OFF"
		}
		]}

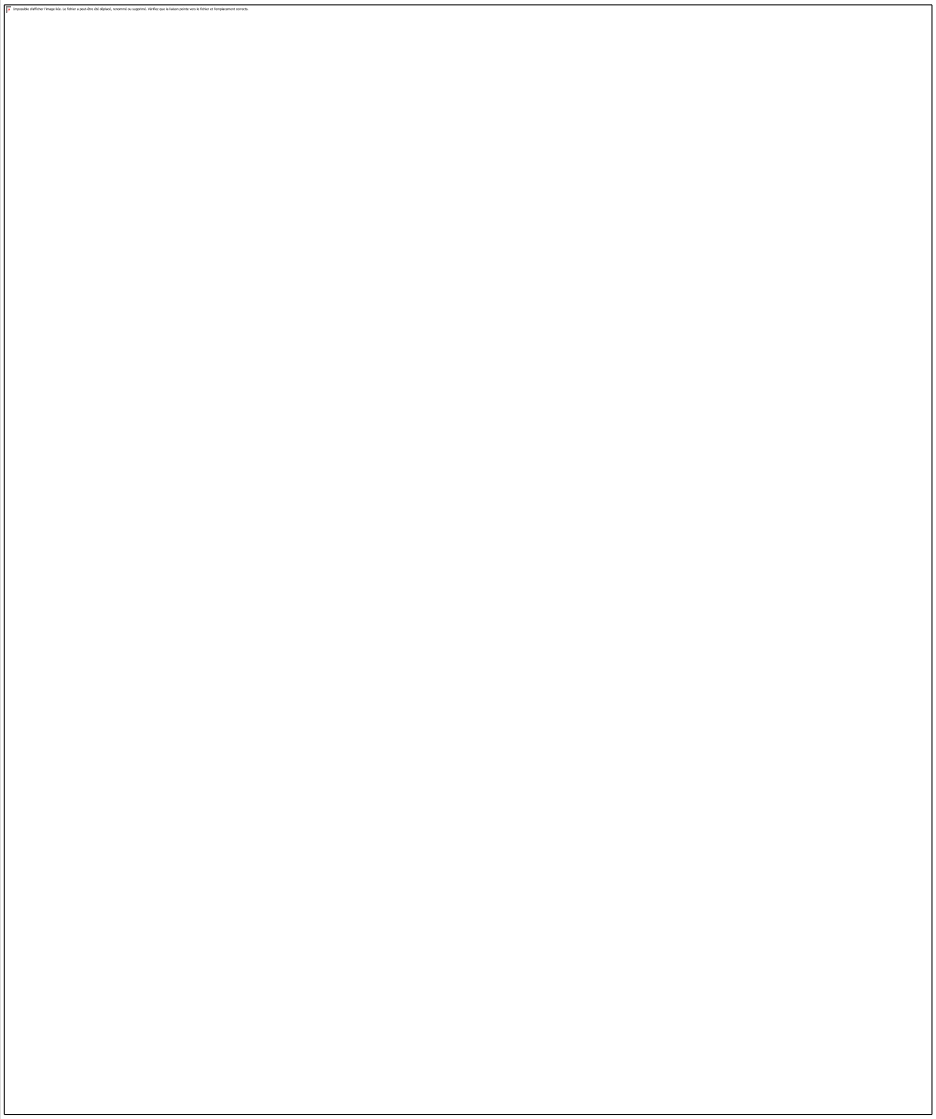
11	curl -X GET -H "Content-Type: application/json" -H "Accept: application/json" http://5g-enodeb3:8080/mon-sat-cli/api/v01.00.00/sna/resource/topology	In 5g-enodeb3, the terrestrial terminal #1 is power on, the terrestrial terminal #2 is power off and the satellite terminal is power off
		
		{"header":{"responseCode":0,"msgType":"restResponseMC"},"content":[
		{
		"node": "tt_1",
		"ip": "172.18.3.1",
		"enabled": "true",
		"status": "MANDATORY_ON"
		},
		{
		"node": "tt_2",
		"ip": "172.18.3.2",
		"enabled": "true",
		"status": "OFF"
		},
		{
		"node": "st_1",
		"ip": "172.18.3.11",
		"enabled": "true",
		"status": "OFF"
		}
		]}

12	curl -v -X POST -H "Content-Type: application/json" -H "Accept: application/json" --data "@UT01/input/indicators_UT01.5g-enodeb3.json" http://5g-enodeb3.5g-ensure.eu:8080/mon-sat-cli/api/v01.00.00/sna/resource/indicators	In 5g-enodeb3, the security/performance indicators has been configured and are sent to the server {"header":{"responseCode":0,"msgType":"restResponseMC"},"content":{}}
13	ssh root5g@<5ge-satellite-network-monitoring-client> docker exec 5g-enodeb3 snmpset -v 2c -c terminal 172.18.3.1 .1.3.6.1.2.1.2.2.1.8.1 i 2	In 5g-enodeb3, emulate a link failure in the terrestrial terminal #1 updating the OID .1.3.6.1.2.1.2.2.1.8.1 in 172.18.3.1 Therefore, the link between 5g-enodeb1 and 5g-enodeb3 is down iso.3.6.1.2.1.2.2.1.8.1 = INTEGER: 2
14	5g-enodeb3 sends an alarm message to the SatelliteNetworkMonitoringServer The SatelliteNetworkMonitoringServer detecs an alarm message in the ActiveMQ (messageType field in the header set to "alarm") and launches the Topology Manager After a few seconds the topology has been reconfigured and the link between 5g-enodeb2 and 5g-enodeb3 is power on	
15	curl -X GET -H "Content-Type: application/json" -H "Accept: application/json" http://5g-enodeb1:8080/mon-sat-cli/api/v01.00.00/sna/resource/topology	In 5g-enodeb1, the terrestrial terminal #3 is power off due to the link failure {"header":{"responseCode":0,"msgType":"restResponseMC"},"content":[{ "node": "tt_1", "ip": "172.18.1.1", "enabled": "true", "status": "MANDATORY_ON", "communities": []

		},
		{
		"node": "tt_2",
		"ip": "172.18.1.2",
		"enabled": "true",
		"status": "MANDATORY_ON",
		"communities": []
		},
		{
		"node": "tt_3",
		"ip": "172.18.1.3",
		"enabled": "false",
		"status": "OFF",
		"communities": []
		}
		]}

16	curl -X GET -H "Content-Type: application/json" -H "Accept: application/json" http://5g-enodeb2:8080/mon-sat-cli/api/v01.00.00/sna/resource/topology	In 5g-enodeb2, the terrestrial terminal #2 is power on in order to fix the link failure (to power on the link between 5g-enodeb2 and 5g-enodeb3)
		
		{"header":{"responseCode":0,"msgType":"restResponseMC"},"content":[
		{
		"node": "tt_1",
		"ip": "172.18.2.1",
		"enabled": "true",
		"status": "MANDATORY_ON",
		"communities": []
		},
		{
		"node": "tt_2",
		"ip": "172.18.2.2",
		"enabled": "true",
		"status": "ON",
		"communities": []
		}
		]}a

17	curl -X GET -H "Content-Type: application/json" -H "Accept: application/json" http://5g-enodeb3:8080/mon-sat-cli/api/v01.00.00/sna/resource/topology	In 5g-enodeb3, the terrestrial terminal #1 is power off due to the link failure and the terrestrial terminal #2 is power on in order to fix the link failure (to power on the link between 5g-enodeb2 and 5g-enodeb3) {"header":{"responseCode":0,"msgType":"restResponseMC"},"content":[{ "node": "tt_1", "ip": "172.18.3.1", "enabled": "false", "status": "OFF", "communities": [] }, { "node": "tt_2", "ip": "172.18.3.2", "enabled": "true", "status": "ON", "communities": [] }, {

		"node": "st_1", "ip": "172.18.3.11", "enabled": "true", "status": "OFF", "communities": [] }]}
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Feature-4.2.1: Pseudo real-time monitoring Feature-4.2.2: Threat detection Use Case 8.1: Satellite-Capable eNB	
Attached files	TopologyMatrix : TopologyMatrix.png 	

Use Cases cluster 9 - Trusted Core Network and Interconnect

T_UC9.3_1 Hardening or patching of systems is not done

Description: Detailed description of threat and its importance	If the systems are not hardened correctly or if the patching processes do not keep the systems up-to-date, the systems could be compromised through the vulnerabilities existing in the systems.
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, e2e effect…)	Systems can be compromised through the vulnerabilities and elevated privileges gained. Thus, total control of a node can be achieved.
Possible Mitigation Hints (optional, if foreseen): How can we protect against the threat?	Monitoring of systems can help in detecting breaches. This can potentially be cooperative actions between different operators, so that indicators of compromise are reported to the operator of the source traffic. Proper segmentation of systems can isolate the breach to only one system. Thus, other systems should be considered potentially hostile.
Entry Points (optional, if known): What possible means does an adversary have?	Abuse of software vulnerabilities in the software
5G-ENSURE enablers (optional, if covered for given threat): What possible means does an adversary have?	Proactive security analysis and remediation Microsegmentation

Test Case 5ge-127: T_UC9.3_1 - "Hardening or patching of systems is not done" R2		
Summary: 5G networks allow more dynamism through virtualisation and new functions can be introduced to the network on the fly. As these environments are more virtualised, there is always a danger that someone manages to introduce a malicious function into the network. Similarly, unauthorized physical elements could be attached to the network, if their authenticity is only based on the location in the network. This test case describes the sequence of steps that correspond to Release 2 of Tust Builder. For the detailed description of individual steps please refer to the attached document "Modelling_T_UC9.3_1_TrustBuilder_Release2.pdf".		
Preconditions: 1) The HN and the VN have a roaming agreement 2) The VN does not have up-to-date patch management 3) There is an exploitable vulnerability in the VN infrastructure 4) Poor physical security of the VN has resulted in the installation of unauthorised device		
#:	Step actions:	Expected Results:
1	After starting the VM the Trust Builder can be accessed in a browser on this URL:localhost:7070/system-modeller	The main page presented to the user (see Figure 3)
2	Click on “Login”.	The login screen is presented to the user (Figure

		4).
3	Use these login details: Username: trustbuilder Password: 5fd4661f32ef9d2be4a3f794dff64cdd	After a successful login the user the user will be presented with the list of previously designed models and also can create new models (see Figure 5).
4	Click on the “Create New Model” for creating an empty model and select “5G Model” option.	see Figure 6, 7
5	Click the green “Edit” control of the newly created model.	see Figure 8
6	Select asset icons from left panel and drag the assets into the model canvas.	see Figure 9.
7	Set connections between assets.	see Figure 10,11.
8	Validate the model by clicking on the green triangle at the top right corner of the design canvas.	see Figure 12.
9	Click on the MME-H asset (indicated by green frame around MME-H).	On the right panel we can see all threats that can be associated with MME-H, see Figure 13, 14.
10	The threats can be resolved by selecting options under the “Control Set” tab. A list of controls available for MME-H.	see Figure 15.
11	Select “*Software Patching*” from the list of controls.	Several threats associated with MME-H will get resolved, see Figure 16,17.
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Use Case 9.3: Authentication of New Network Elements Feature-3.3.1: 5G Asset Model Feature-3.3.2: 5G Threat knowledge base v1 Feature-3.3.3: Graphical editor	
Attached files	Modelling_T_UC9.3_1_TrustBuilder_Release2 : Modelling_T_UC9.3_1_TrustBuilder_Release2.pdf Modelling_T_UC9.3_1_TrustBuilder_Release2.pdf	

Use Cases cluster 10 - 5G Enhanced Security Services

T_UC10.2_1 Nefarious activities: privacy violations

Description: Detailed description of threat and its importance	Mobile devices and the installed applications disclose a large amount of private information both personal and device related information mostly through misbehaving apps, PUAs (Potentially Unwanted Applications), adware and ransomware.
Potential effect: What global effect it will have on major 5G system domains (network, hosts, applications, end users, end devices, e2e effect…)	Threat effect: information leakage, disclosure of sensitive info, privacy violation in general.
Possible Mitigation Hints **(optional, if foreseen):** How can we protect against the threat?	Potential solutions include means to protect the user's privacy at the application layer. The 5G network adopts a privacy policy containing various privacy parameters (related to device and apps activity on user data) that can be controlled on user's demand or upon some anomalous event detection. The 5G network offers to subscribers a service that checks the privacy risk of devices and their installed apps. A useful tool for this service is to require the mobile applications and servers to declare a human readable privacy policy and to offer a tool to the user's device to verify it. 5G should support an application level service that provides privacy policy analysis.
Entry Points **(optional, if known):** What possible means does an adversary have?	Compromised devices by malicious app.
5G-ENSURE enablers **(optional, if covered for given threat):** What possible means does an adversary have?	The enabler is Policy Privacy Analysis

Test Case 5ge-142: Modelling T_UC10.2_1 Nefarious activities: "privacy violations"		
Summary: Nowadays, users of networked services are confronted with a plethora of services and applications that may put their privacy at risk right through the stack from the core network (potentially) to over-the-top application services. Currently it is difficult for a user to understand the privacy implications of using a mobile service or application: privacy policies (where they exist) are often not easy for users to read and commonly not presented upfront to the user. This issue is going to be even more pressing within 5G networks where a single service may be the result of a compositions of different layers managed by different parties with different views on privacy. For the detailed description of the test please refe to the attached document "Modelling T_UC10.2_1_PrivacyEnabler_R2.pdf".		
#:	Step actions:	Expected Results:
1	Start the client	
2	Cd into the client folder:	cd privacy-analysis-enabler/client

3	Run the client script	./run.sh
4	Start the server Cd into the server folder cd ../server	
5	Run the server script ./run.sh	
6	Check that all docker containers are running	
7	Access the enabler from the browser at http://localhost:8888/	
8	Login as normal user: password: user@privacy.org password: privacyuser	
9		
Execution type:	Manual	
Estimated exec. duration (min):		
Priority:	Medium	
Scenario evaluation score:	**3 - Testbed evaluation (simulation)**	
Requirements	Use Case 10.2: Privacy Violation Mitigation Feature-2.3.1: Privacy policy specification Feature-2.3.2: Privacy preferences specification Feature-2.3.3: Comparison of policies and preferences	
Attached files	Modelling T_UC10.2_1_PrivacyEnabler_R2 : Modelling_T_UC9.3_1_TrustBuilder_Release2.pdf	

E ANNEX : integration of enablers R1

	Enablers & Features Release R1									
enablers candidate for integration						features candidate for integration				
	Enabler	ID WP4	Integrated				feature	ID WP4	inetgrated	
1	GCI (Orange)	4.1	YES	1		1	Log and Event Processing	4.1.1	YES	1
2	IoT (SICS)	1.1	YES	2		2	Group authentication by extending the LTE-AKA protocol	1.1.1	YES	2
3	Fine-grained Authorization	1.2	YES			3	Basic Authorization in Satellite systems (TASE)	1.2.1	**NO**	
	Fine-grained Authorization			3		4	Basic distributed authorization Enforcement for RCDs (TS)	1.2.2	YES	3
4	Satellite Network Monitoring (TASE)	4.3	**NO**			(note1)	Pseudo real-time monitoring	4.3.1	**NO**	
	Satellite Network Monitoring (TASE)						Threat detection	4.3.2	**NO**	
5	Component-Interaction audits (NEC)	5.2	YES	4		5	Basic OpenFlow Compliance Checker	5.2.1	YES	4
6	Device identifier(s) privacy	2.2	YES	5		6	Enhanced privacy for network attachment protocols (OXFORD)	2.2.1	YES	5
7	Bootstrapping trust (SICS)	5.3	YES	6		7	Integrity Attestation of virtual network components	5.3.1	YES	6
8	Access control mechanism (NEC)	5.1	YES	7		8	Southbound Reference Monitor	5.1.1	YES	7
9	Microsegmentation (VTT)	5.4	YES	8		9	Dynamic Arrangement of Micro-Segments	5.4.1	YES	8
10	Security monitor for 5G microsegments (VTT)	4.2	YES	9		10	Complex Event Processing Framework for Security Monitoring and Inferencing	4.2.1	YES	9
11	Pulsar: Proactive security analysis and remediation (TS)	4.4	YES	10		11	5G specific vulnerability schema	4.4.1	YES	10
12	Trust builder (IT-INNOV)	3.3	YES	11		12	5G Asset Model	3.3.1	YES	11
	Trust builder (IT-INNOV)					13	Graphical editor v2	3.3.2	YES	12
13	Trust metric enabler (VTT)	3.2	YES	12		14	Trust metric based network domain security policy management	3.2.1	YES	13
14	VNF certification (TCS)	3.1	YES	13		15	VNF Trustworthiness Evaluation	3.1.1	YES	14
15	Privacy Enhanced Identity Protection (TIIT)	2.1	YES	14		16	Encryption of Long Term Identifiers(IMSI public-key based encryption)	2.1.1	YES	15
15 enablers			14 enablers integrated			16 features				15 features integrated
	14/15 enablers integrated						15/16 features integrated			
(note1) :	an enabler that is not delivered over the TestBed, or for which the packaging do not allow Testbed to perform integration test could not be referenced as delivering features.									

F ANNEX : integration of enablers R2

enablers R2 candidate for integration	Enablers R2	ID WP4	enablers integrated		Features R2 candidate for integration	features R2	ID WP4		number
1	System Security State Repository	4.5	NO		(note1)	Deployment model ontology (also known as 5G asset model	4.5.1	**NO**	
	System Security State Repository					System Security State Repository service	4.5.2	**NO**	
2	Microsegment monitor	4.2	YES	1	1	Complex Event Processing Framework for Security Monitoring and Inferencing	4.2.1	**NO**	
	Microsegment monitor				2	Risk-based adaptation of micro-segments	4.2.2	**NO**	
	Microsegment monitor				3	Extended data gathering	4.2.3	YES	1
	Microsegment monitor				4	Cross-domain information exchange	4.2.4	YES	2
3	Satellite Network Monitoring	4.3	NO			Pseudo real-time monitoring	4.3.1	**NO**	
	Satellite Network Monitoring				(note1)	Threat detection	4.3.2	**NO**	
	Satellite Network Monitoring					Active security analysis	4.3.3	**NO**	
	Satellite Network Monitoring					Pre-emptive mitigation security actions	4.3.4	**NO**	
4	PulSAR: Proactive Security Analysis and Remediation	4.4	YES	2	5	5G specific vulnerability schema	4.4.1	**NO**	
	PulSAR: Proactive Security Analysis and Remediation				6	5G specific vulnerability schema implementation	4.4.2	YES	3
	PulSAR: Proactive Security Analysis and Remediation				7	PulSAR interface with Generic Collector	4.4.3	YES	4
5	Component-Interaction Audits	5.2	YES	3	8	Basic OpenFlow Compliance Checker	5.2.1	**NO**	
	Component-Interaction Audits				9	Basic NFV Reconfiguration Compliance Checker	5.2.2	YES	5
6	Bootstrapping Trust	5.3	YES	4	10	Integrity Attestation of Virtual Network Components	5.3.1	**NO**	
	Bootstrapping Trust				11	Integrity Attestation of VNFs running in Docker containers	5.3.2	YES	6
7	Micro Segmentation	5.4	NO			Dynamic Arrangement of Micro-Segments	5.4.1	**NO**	
	Micro Segmentation				(note1)	Extended Northbound API	5.4.2	**NO**	
	Micro Segmentation					Support for multi-domain micro-segments	5.4.3	**NO**	
8	Flow Control: in-network Threat Detection and Mitigation for Critical Functions in Virtual Networks	5.6	NO		(note1)	Detection of malicious behaviours in virtual networks	5.6.1	**NO**	
	Flow Control: in-network Threat Detection and Mitigation for Critical Functions in Virtual Networks					Mitigation of detected network threats	5.6.2	**NO**	
9	(IoT)	1.1	NO			Group based AKA	1.1.1	**NO**	
	(IoT)				(note1)	Non-USIM based AKA (R2)	1.1.2	**NO**	
	(IoT)					BYOI (Bring Your Own Identity) (R2)	1.1.3	**NO**	
	(IoT)					vGBA (Vertical GBA) (R2)	1.1.4	**NO**	
10	Fine-grained Authorization R1	1.2	NO		(note1)	Basic Authorization in Satellite systems (R1)**	1.2.1	**NO**	
	Fine-grained Authorization R1					Basic distributed authorization Enforcement for RCDs (R1)	1.2.2	**NO**	
11	*Fine-grained Authorization R2*		YES	5	12	AAA integration with satellite systems (R2)**	1.2.3	**NO**	
	Fine-grained Authorization R2				13	Authorization and authentication for RCD based on ongoing IETF standardization (R2)**	1.2.4	YES	7
11	Privacy Enhanced Identity Protection	2.1	YES	6	14	Encryption of Long Term Identifiers (IMSI public-key based encryption) (R1)	2.1.1	**NO**	
	Privacy Enhanced Identity Protection				15	Home Network centric IMSI protection (R2)	2.1.2	**NO**	
	Privacy Enhanced Identity Protection				16	IMSI Pseudonymization (R2)**	2.1.3	YES	8
12	Device identifier(s) privacy	2.2	YES	7	17	Enhanced privacy for network attachment protocols (R1)	2.2.1	**NO**	
	Device identifier(s) privacy				18	Anonymous and optimised address selection for network attachment protocols (R2) ?	2.2.2	YES	9
13	Privacy policy analysis	2.4	NO			Privacy policy specification (R2)**	2.4.1	**NO**	
	Privacy policy analysis				(note1)	Privacy preferences specification (R2)**	2.4.2	**NO**	
	Privacy policy analysis					Comparison of policies and preferences (R2)**	2.4.3	**NO**	
14	Trust Builder	3.3	NO			5G Asset model (R1/R2)**	3.3.1	**NO**	
	Trust Builder				(note1)	Graphical editor (R1/R2)**	3.3.2	**NO**	
	Trust Builder					5G Threat knowledgebase (R2)**	3.3.3	**NO**	
15	Trust Metric Enabler	3.2	YES	8	19	Trust metric based network domain security policy management (R1)	3.2.1	**NO**	
	Trust Metric Enabler				20	Improved trust metric based on extended data (R2)**	3.2.2	YES	10
16	VNF Certification	3.1	YES	9	21	VNF Trustworthiness Evaluation (R1)	3.1.1	**NO**	
	VNF Certification				22	VNF Trustworthiness Certification (R2)**	3.1.2	YES	11
					22 candidate features				11 features integrated
16 enablers			9 enablers R2 integrated						

(note1) : an enabler that is not delivered over the TestBed, or for which the packaging do not allow Testbed to perform integration test could not be referenced as delivering features.