



SDN-μSense

Project No. 833955

Project acronym: SDN-microSENSE

Project title:

SDN - microgrid reSilient Electrical eNergy SystEm

Deliverable D3.5

SDN-microSENSE Risk Assessment Framework

Programme: H2020-SU-DS-2018

Start of the project: 01.05.2019

Duration: 36 months

Editor: UBITECH

Due date of deliverable: 31/08/2020

Actual submission date: 16/09/2020



Deliverable Description:

Deliverable Name	SDN-microSENSE Risk Assessment Framework
Deliverable Number	D3.5
Work Package	WP 3
Associated Task	T3.5
Covered Period	M3-M16
Due Date	M16
Completion Date	M17
Submission Date	16/09/2020
Deliverable Lead Partner	UBITECH
Deliverable Author(s)	UBITECH
Version	1.0

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

CHANGE CONTROL
DOCUMENT HISTORY

Version	Date	Change History	Author(s)	Organisation
0.1	09/06/2020	ToC	Entso Veliou, Sofia nna Menesidou	UBITECH
0.2	22/06/2020	Review of TOC and TOC changes based on review	Anastasios Drosou	CERTH
0.3	06/07/2020	Section 2 first draft	Entso Veliou, Dimitris Papamartzivanos	UBITECH
0.4	17/07/2020	Section 3 first draft	Dimitris Papamartzivanos, Giannis Ledaki	UBITECH
0.4.1	20/07/2020	Integration View – Presenting S-RAF Interfaces and components	Marisa Escalante Martinez	TECNALIA
0.4.2	22/07/2020	Integration View – Presenting S-RAF Interfaces and components	Angel Javier Jimenez Perez	AYESA
0.4.3	30/07/2020	Integration View – Presenting S-RAF Interfaces	Ruben Trapero	ATOS

0.5	03/08/2020	Pilots information from online workshops, section 5 first draft	Giannis Ledakis	UBITECH (with content from all pilots)
0.6.1	08/08/2020	SDN-microSENSE Risk Assessment Framework Architecture - first draft, section 2 updates	Dimitris Papamartzivanos, Giannis Ledaki	UBITECH
0.6.2	13/08/2020	SDN-microSENSE Risk Assessment Framework Architecture, section 2 updates	Dimitris Papamartzivanos	UBITECH (based on discussions with TECNALIA, AYESA, ATOS, IREC)
0.7.1	28/08/2020	SDN-microSENSE Risk Assessment Framework Implementation Details	Giannis Ledakis, Dimitris Papamartzivanos	UBITECH
0.7.2	04/09/2020	SDN-microSENSE Risk Assessment Framework Adoption and Usage, section 3 and 4 updates	Giannis Ledakis, Dimitris Papamartzivanos, Sofianna Menesidou	UBITECH
0.7.3	07/07/2020	Introduction and Conclusions finalized, ready for comments	Sofianna Menesidou	UBITECH
0.7.4	09/09/2020	Ready for first review	Giannis Ledakis, Konstantinos Oikonomou	UBITECH
0.7.5	10/09/2020	Review by CEZ	Yasen Todorov	CEZ
0.8	10/09/2020	Addressing comments, Final updates on section 3 and section 4	Giannis Ledakis, Dimitris Papamartzivanos, Sofianna Menesidou	UBITECH
0.9	15/09/2020	Final review by CERTH	Anastasis Drosou	CERTH
1.0	16/09/2020	Final Version	Giannis Ledakis, Kostas Oikonomou	UBITECH

DISTRIBUTION LIST

Date	Issue	Group
16/09/2020	Revision	UBITECH
16/09/2020	Acceptance	CEZ, CERTH, SAB
16/09/2020	Submission	AYESA

SAB APPROVAL

NAME	INSTITUTION	DATE
Dr. Marc Stauch on behalf of Prof. Dr. Tina Krügel	LUH	14/09/2020
Sokratis Katsikas	NTNU	13/09/2020

Academic and Industrial partner revision

NAME	INSTITUTION	DATE
Anastasis Drosou	Academic partner: CERTH	15/09/2020
Yasen Todorov	Industrial partner: CEZ	10/09/2020

Quality and Technical manager revision

NAME	INSTITUTION	DATE
Dimosthenis Ioannidis	CERTH	22/06/2020 & 10/09/2020

Table of contents

Table of contents.....	5
List of Figures.....	8
List of Tables	9
Acronyms	10
Executive Summary.....	11
1 Introduction	12
1.1 Purpose of the Document	12
1.2 Structure of the Document.....	12
1.3 Methodology	12
1.4 Relation to other Deliverables and Work Packages	12
2 SDN-microSENSE Risk Assessment Framework as part of the Risk Assessment Market	14
2.1 Risk Assessment Methods/Tools	14
2.1.1 Risk Identification	15
2.1.1.1 Asset Identification	15
2.1.1.2 Identify Threats.....	16
2.1.1.3 Construct Risk Scenarios	16
2.1.2 Risk Analysis.....	16
2.1.2.1 Determine Likelihood	17
2.1.2.2 Calculate Impact	17
2.1.3 Risk Calculation.....	17
2.1.3.1 Determine and Prioritise Risk	18
2.1.3.2 Risk Report.....	18
2.2 EPES Best Practices, Security Issues and Assessment	19
2.3 SDN-microSENSE Risk Assessment Framework (S-RAF)	19
2.3.1 SDN-microSENSE Risk Assessment methodology	20
2.3.2 OLISTIC as part of SDN-microSENSE Risk Assessment.....	22
2.3.3 SDN-microSENSE Risk Assessment Framework (S-RAF) Differentiations	25
2.3.3.1 Cumulative Risk Assessment.....	25
2.3.3.2 Risk Assessment with Focus on EPES	26
2.3.3.3 Deep Insights on the security issues of EPES	26

3	<i>SDN-microSENSE Risk Assessment Framework Architecture.....</i>	27
3.1	Architecture Overview - Conceptual Architecture.....	27
3.2	Structural and Components View - Presenting S-RAF Components.....	29
3.2.1	Functional View: Presenting S-RAF Components.....	29
3.2.1.1	Asset Identification Component.....	29
3.2.1.2	Threat Identification Component.....	29
3.2.1.3	S-RAF Dashboard and Controller	29
3.2.1.4	Impact Analysis Component.....	29
3.2.1.5	Honeypots Management Component.....	30
3.2.1.6	Vulnerability Management Component	30
3.2.1.7	Risk Level Assessment Component	32
3.2.1.8	Pub-Sub Queue (Kafka)	32
3.3	Integration View – Presenting S-RAF Interfaces	33
3.3.1.1	eVul to Impact Analysis Component Interface.....	33
3.3.1.2	AIDB to S-RAF Interface.....	34
3.3.1.3	S-RAF to SDN-SELF Interface	36
3.3.1.4	S-RAF to ARIEC Interface	38
3.3.1.5	XL-SIEM to RAF Interface.....	38
3.3.1.6	Controller to Dash Interface.....	43
3.3.1.7	Risk Level Assessment Interface to Controller	43
3.3.1.8	Impact Analysis to Risk Level Assessment Interface	44
3.3.1.9	Assets Identification to Impact Analysis	44
3.4	Behavioural View.....	45
4	<i>SDN-microSENSE Risk Assessment Framework Implementation Details</i>	49
4.1	Technologies and Standards.....	49
4.2	S-RAF Implementation.....	49
4.2.1	Packaging and Code structure	50
4.2.2	Data Model Highlights	51
4.3	Compilation Process	54
4.3.1	Prerequisites.....	54
4.4	Deployment Process.....	54
4.5	Risk Assessment Process Description	54
4.5.1	Risk Calculation.....	55

4.5.1.1	Individual Risk Calculation	55
4.5.1.2	Cumulative Risk Calculation.....	55
4.5.1.3	Risk Profiles.....	61
4.5.1.4	Integration with Personnel Assessment	62
4.6	Requirements and Unit Testing Coverage	62
5	<i>SDN-microSENSE Risk Assessment Framework Adoption and Usage</i>	64
5.1	Installation and Deployment	64
5.1.1	Preparing the environment	64
5.1.1.1	Prerequisites.....	64
5.1.2	Deployment	64
5.1.3	Verification	64
5.1.4	Undeployment.....	65
5.1.5	Deployment View	65
5.2	S-RAF Usage.....	65
5.2.1	Usage workflows	65
5.2.1.1	Asset management	65
5.2.1.2	Vulnerability and threat management.....	67
5.2.1.3	Risk assessment and evaluation	69
5.2.1.4	Controls management	72
5.2.2	Mapping between Assets, Threat and possible attack patterns	74
5.3	S-RAF in the Context of Demonstrators	80
6	<i>Conclusions</i>	82
7	<i>References.....</i>	83
8	<i>Annex I – Docker Compose for S-RAF installation.....</i>	84
9	<i>Annex II –Unit Tests.....</i>	86

List of Figures

Figure 1: Deliverable D3.5 relationship within the SDN-microSENSE project	13
Figure 2: Risk Calculation Impact Categories	17
Figure 3: Risk Matrix	18
Figure 4: ECRA basic steps	21
Figure 5: SDN-microSENSE Architecture Structural View	27
Figure 6: S-RAF in the Application Plane	27
Figure 7: General view of the components and the relationships with other components	28
Figure 8: Vulnerability info	31
Figure 9: Vulnerability manager – Vulnerability life cycle	31
Figure 10: S-RAF basic behaviour as defined in D2.3	45
Figure 11: Detailed Sequence diagram of S-RAF – New asset discovery scenario	46
Figure 12: Detailed Sequence diagram of S-RAF – New vulnerability discovery scenario	47
Figure 13: Detailed Sequence diagram of S-RAF – Re-assessment of risk scenario	48
Figure 14: Detailed Sequence diagram of S-RAF – Risk assessment due to alarm from XL-SIEM	48
Figure 15: Packaging of S-RAF code	50
Figure 16: Software modules as part of the main .pom file of S-RAF	51
Figure 17: Assets Representation	52
Figure 18: Threats Representation	52
Figure 19: Vulnerabilities Representation	53
Figure 20: Representation of Scenarios for Risk Assessment in EPES for S-RAF	53
Figure 21: Code snippet for attack path Identification in Attack Path Analysis Service	58
Figure 22: Code snippet for analysing attack paths in cumulative manner by the Attack Path Analysis Service	59
Figure 23: Code snippet of the business logic implemented by Attack Path Analysis Service	59
Figure 24: Code snippet of extracting attack paths from a graph of assets	60
Figure 25: Code snippet of attack path risk analysis	61
Figure 26: S-RAF component's deployment view	65
Figure 27: Asset management (List)	66
Figure 28: Asset management (Add)	66
Figure 29: Asset management (Visualisation)	67
Figure 30: Vulnerability management	67
Figure 31: Vulnerability management (CVE-2019-14931)	68
Figure 32: Threat management	68
Figure 33: Risk Appetite	69
Figure 34: Attack Scenario	69
Figure 35: Risk Assessment General Dashboard	70
Figure 36: Individual Risk Level Report	70
Figure 37: Executive IRL Summary	71
Figure 38: Cumulative Attack Path	71
Figure 39: Cumulative Risk Level Report	72
Figure 40: Executive CRL Summary	72
Figure 41: Controls for Risk Mitigation	73

Figure 42: Assign Control to SDN-Switch	73
Figure 43: Mitigation Comparison	74

List of Tables

Table 1: Vulnerability risk level ranges	31
Table 2: Details of the eVul-RA Interface	34
Table 3: Details of the AIDB-RAF Interface	35
Table 4: Details of the RAF-EDAE Interface	37
Table 5: Details of the RAF-ARIEC Interface	38
Table 6: Details of the CIS-RAF Interface	39
Table 7: Details of the CONT-DASH Interface	43
Table 8: Details of the RLA-CONT Interface	44
Table 9: Details of the IMP-RLA Interface	44
Table 10: Details of the ASI-IMPInterface	45
Table 11: Description of the high skilled Attacker Types according to NIST	56
Table 12: SDN-microSENSE Requirements relevant to the Risk Assessment	63
Table 13: Assets of the Electrical Power and Energy System ecosystem in the context of SDN-microSENSE	77
Table 14: Threats, attack patterns and affected assets for EPES	79
Table 15: Representative assets of demonstrators, for S-RAF	81

Acronyms

Acronym	Explanation
API	Application Programming Interfaces
APT	Advanced Persistent Threat
ARIEC	Anonymous Repository of Incidents
BCM	Business Continuity Management
BCMS	Business Continuity Management System
CAPEC	Common Attack Pattern Enumeration and Classification
CIL	Cumulative Impact Level
CRL	Cumulative Risk Level
CVE	Common Vulnerability and Exposure
CVL	Cumulative Vulnerability Level
CVSS	Common Vulnerability Scoring System
DCS	Distributed Control System
DoS	Denial of Service
EC	European Commission
ECRA	Energy Chain Risk Assessment
ECSO	European Cyber Security Organisation
ENISA	European Union Agency for Cybersecurity
EPES	Electrical Power and Energy System
ICS	Industrial Control Systems
IEC	International Electrotechnical Commission
IRL	Individual Risk Level
ISACA	Information Systems Audit and Control Association
ISMS	Information Security Management System
ISO	International Organization for Standardisation
MISP	Malware Information Sharing Platform
PLC	Programmable Logic Controller
PMU	Phasor Measurement Unit
RA	Risk Assessment
RTU	Remote Terminal Unit
S-RAF	SDN-microSENSE Risk Assessment Framework
SCADA	Supervisory Control and Data Acquisition
UDP	User Datagram Protocol
UML	Unified Modeling Language
WPs	Work Packages
XL-SIEM	Cross-Layer Security Information and Event Management

Executive Summary

This deliverable describes the final form of the SDN-microSENSE Risk Assessment Framework (S-RAF). S-RAF differentiates from other risk assessment tools and acts as a complementary tool for the security of EPES infrastructure. More specifically, the following four are the main key points of differentiation between the adoption of S-RAF methodology and other risk assessment solutions: a) the cumulative risk assessment, b) the calculation methodology, c) the EPES-focused risk assessment and d) the deep insights on the security of EPES.

S-RAF assesses the level of risk in all the involved EPES devices and systems by analysing a) current smart & IoT devices, b) legacy SCADA & ICS devices, c) smart meters d) other software/hardware devices connected to EPES network and e) all the energy-related personnel and stakeholders (energy operators, consumers, prosumers, energy utilities, energy generators, energy actors & agents and energy retailers). S-RAF considers all the aforementioned assets and utilises the models produced from task T3.2, the output of honeypots developed in task T3.3 and the including readiness results acquired from T3.4, to provide a more holistic approach.

Due to the distributed nature of the decentralised energy system, S-RAF takes into account the collaborative aspects needed to involve all stakeholders of the energy components, making the calculation of the cumulative risk of paramount importance. In other words, S-RAF cumulative risk assessment approach enables one to perceive the security state at the level of mission-critical assets that belong either in the same business workflow, or in the same physical (or virtual) networks.

In addition, S-RAF extends UBITECH's OLISTIC Enterprise Risk Management Suite. In the context of SDN-microSENSE, several updates have been performed to address the need for a collaborative risk assessment framework for EPES compared with OLISTIC. These updates can be summarised as the following eight:

1. Usage of updated model that supports EPES
2. Collaborative Risk Assessment with the cumulative RA
3. Connecting with AIDB for EPES asset retrieval
4. Connection with eVul for automated analysis of vulnerabilities in the EPES environment
5. Extending OLISTIC model and components for retrieving alerts from XL-SIEM
6. Providing Incidents based on the Risk Assessment as output to SDN-SELF and ARIEC components of SDN-microSENSE
7. Integrating Apache Kafka as message queue that can be used by both internal and external components
8. Transform data to MISP format

1 Introduction

Risk assessment is of paramount importance for the efficient operation of Information and Communications Technology (ICT) deployments and hence for the EPES field. The SDN-microSENSE Risk Assessment Framework (S-RAF) component is tailored to the security requirements of EPES and the requirements defined in D2.2 [1], while it takes into account the collaborative aspects needed to involve all stakeholders of the energy components. S-RAF provides both the individual and cumulative risks in alignment with the proposed Energy Chain Risk Assessment (ECRA) Methodology.

1.1 Purpose of the Document

The main purpose of this deliverable is to document the results of the collaborative risk assessment (RA) Framework for energy chain S-RAF. As this is a public document, special care has been taken in order to handle privacy restrictions of the work performed in the scope of WP3.

1.2 Structure of the Document

The rest of the document is organised as follows. In Chapter 2, we describe the background of RA in EPES, by examining existing Risk Assessment tools and presenting the methodology of S-RAF. In Chapter 3, we describe the architecture of S-RAF, while Chapter 4 provides implementation details. Chapter 5 includes installation and usage instructions. Finally, Chapter 6 concludes the deliverable.

1.3 Methodology

Both for the implementation of S-RAF and the writing of this deliverable we have mainly relied on the work performed on tasks T3.1 and T3.2 that described the RA in the scope of EPES, and the overall platform architecture defined in WP2. Based on the requirements and the design aspects of these deliverables, the OLISTIC RA engine of UBITECH has been extended to provide the needed functionality. In addition, in order to ensure that the developed solution conforms to SDN-microSENSE use case needs, we also proceeded with the organisation of preliminary workshops with the use case leaders.

1.4 Relation to other Deliverables and Work Packages

Deliverables D3.1 [2] and D3.2 [3] served as the basis to the D3.5 SDN-microSENSE Risk Assessment Framework. D3.2 [3] considers the user security and privacy requirements described in D2.2 [1], the overall SDN-microSENSE architecture analysed in D2.3 [4] and the Risk Assessment Methodology of Energy Chain described in D3.1 [2]. Figure 1 depicts the dependencies of the deliverable to the other deliverables and Work Packages (WPs).

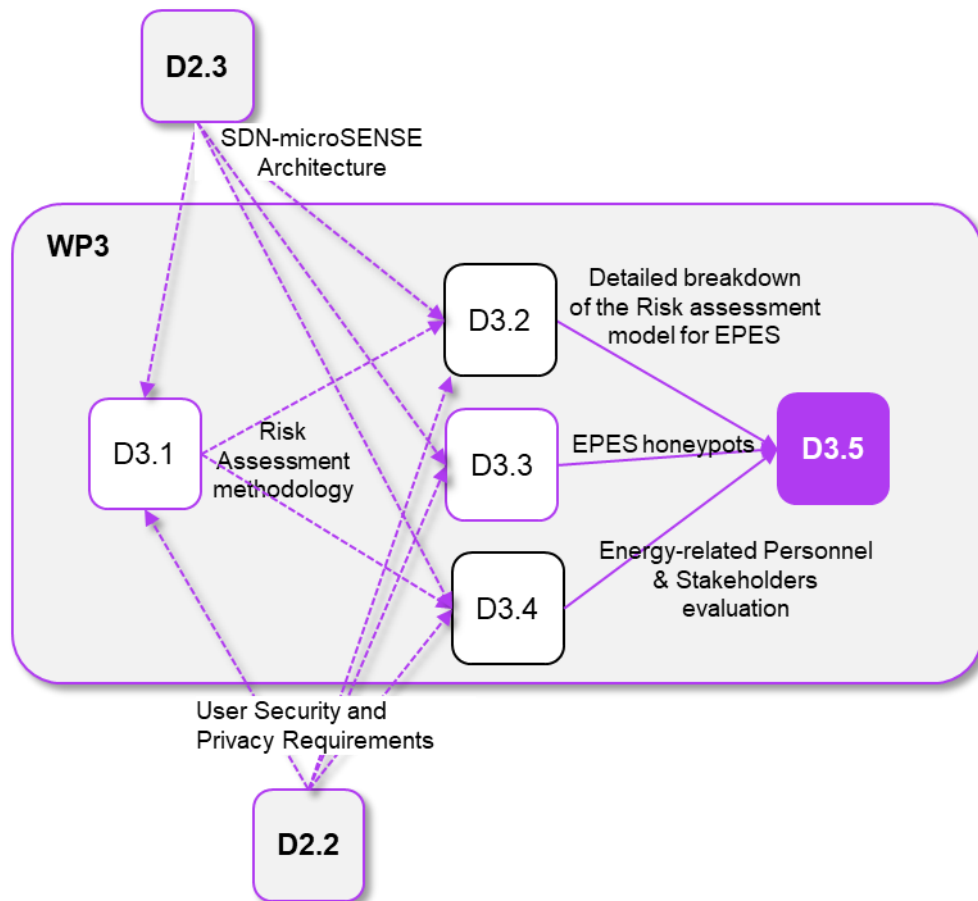


Figure 1: Deliverable D3.5 relationship within the SDN-microSENSE project

2 SDN-microSENSE Risk Assessment Framework as part of the Risk Assessment Market

Risk management is the process of identifying, analysing and then responding to any risk that arises over the life cycle of any business operation and can potentially harm the output of that operation and any other related operation. Risk management isn't reactive only; it should be part of the planning process to figure out risk that might happen in the business operation and how to control that risk if it in fact occurs.

A risk is anything that could potentially impact timeline, performance or budget. Risks are potentialities, and in a risk management context, if they become realities, they then become classified as "issues" that must be addressed. So, risk management then, is the process of identifying, categorising, prioritising and planning for risks before they become issues.

While risk management processes and searching activities are taking place at the development stage, there is another metric which calculates the same concept described above and is called risk assessment. Risk assessment is operating while the business is already in place and the development phase has been completed. There are multiple risk assessment metrics which can be calculated, however for the scope of the SDN-microSENSE project our main concern is cyber security risk assessment and the harm of the EPES infrastructure concerning the cyber security risk.

Fortunately, there are several platforms that gather third-party cyber risk data and provide a risk score or security rating for companies. The information gathering is done by a method called "passive scan" where non-intrusive methods are used and company assets remain untouched. It is basically a hacker's view of the third-parties external cyber risk. The open-source intelligence data is collected from many feeds such as reputation services, hacker sites/forums, vulnerability databases, Internet-wide scanners, social media, paste sites, black markets, underground forums, etc. Information gathering should be done for the company of interest and any related third-party company.

2.1 Risk Assessment Methods/Tools

The top players that provide such cyber risk scoring through passive scanning are presented below:

- BitSight [5], provides a security rating for an organization which lies between 250 to 900. This number is characterized by the company more like an insight into a hacker's point of view on an organization's IT security posture.
- NormShield [6], provides different kind of metrics. It provides a security metric which states the organization's cyber security readiness that lies between A to F. Moreover, it provides a metric for the security controls that are in place from 0 to 100 and also a financial metric which indicates the financial loss in monetary amount.
- Security Scorecard [7], provides APT security risk assessment and protection. It uses data gleaned from traffic to/from an organization, as well as other publicly accessible data to build security ratings for evaluating vendors and partners and pricing cyber risk insurance policies, among other use cases. The platform also monitors so-called "hacker chatter", social networks, and public data breach feeds for indicators of compromise. Furthermore, they contain and provide as a service a threat database with their security data collected from all the above information. Security Scorecard provides A to F and 0 to 100 security rating system.

- UpGuard [8], provides a metric which is called CSTAR score. This score is between 0 and 950 and takes into consideration not only security but compliance and integrity as well. It helps to identify and quantify the risk but also make the correct business decisions. Finally, it provides metrics that will help the organization identify their position compared to other organizations of the same industry.
- Riskrecon [9], makes it easy to gain deep, risk contextualized insight into the cybersecurity risk performance of all third-parties by continuously monitoring across 11 security domains and 41 security criteria. It distils its assessment criteria into a simple score from 0-10.

They all provide risk scores or security ratings for any company. This type of cyber risk assessment can be used for suppliers, joint-ventures, target acquisitions, franchisees, cyber insurance customers, etc.

Since the methodology and data resources are similar, the differentiating factors are data quality, technical depth, reliability (true positive), coverage (true negative), usability and reporting capabilities. The ideal scorecard should sufficiently cover the target company's and related third-parties' assets and must exclude any findings that belong to other companies. In other words, the scoring system should be highly reliable (high true positive rate) and consistent (less false negative).

There are no standards on scoring methodologies (yet) for risk scoring products. An easy-to-understand and consistent scoring is very important when assessing the cyber risk posture of your own company and your third-parties.

Some companies use numeric scoring (0 to 900 scoring) and some use letter grades (A to F scoring).

2.1.1 Risk Identification

2.1.1.1 Asset Identification

As the old security adage goes, "You can't protect what you don't know." Therefore, the first task is to identify and create an inventory of all physical and logical assets that make up the system that is within the risk assessment scope. When identifying the assets, it is important to take note of those that are:

- Crown jewels - These assets are critical to achieving the overall business objectives and are usually what the attackers would actively seek to exploit. Example: In a EPES infrastructure of a power station, a Programmable Logic Controller (PLC) controlling the main infrastructure is likely to be considered a crown jewel as it directly affects the consumption of electricity – the overall business objective of the power station. An attacker who wants to disrupt power generation is likely to want to compromise and manipulate the logic within the PLC.
- Stepping stones - These assets are resources that attackers would want to take control and leverage to pivot across network segments before reaching the crown jewels. Example: In a typical Windows environment, an Active Directory (AD) server that maintains/validate user login credentials to multiple servers is likely to be considered a stepping stone, as it provides a bridge for attackers to pivot into these servers.

The asset inventory list is used to consolidate and create a network architecture diagram that provides a visual representation of the interconnectivity and communication paths between the assets. It identifies and labels all entry points (i.e. attack vectors) into the system, as well as the stepping stones and crown jewels. This would help facilitate the next task to identify threats.

2.1.1.2 Identify Threats

With the asset inventory list and network architecture diagram, it is possible to identify the threat events that could exploit the vulnerabilities for each asset. There are many publicly available sources with threat libraries that can be referenced for identifying threat events. Threats events can be systematically identified by taking the steps below:

- I. Apply the threat events in the referenced libraries to each asset that presents an entry point (i.e. attack vector) to the system.
- II. Document relevant threat events that are applicable to each asset.
- III. Enumerate through the assets and repeat steps (i) and (ii) above until all key assets (especially crown jewels and stepping stones) are included.

When enumerating through the assets to identify possible threat events, always keep in mind the attack stages of the Cyber Kill Chain. The Cyber Kill Chain is a useful model that maps out steps and goals of a typical real-world attack. Threat events that are relevant to assets at the system perimeter are typically categorised in the early stages of the Cyber Kill Chain. As we move deeper into the system, threat events that are more relevant would be categorised in the later stages of the Cyber Kill Chain (e.g. lateral movement, command and control).

2.1.1.3 Construct Risk Scenarios

Constructing risk scenarios is the last task to complete the Risk Identification. This task aims to create “what could go wrong” scenarios that provide a realistic and relatable view of risks based on the business context, system environment and pertinent threats. A well-constructed risk scenario facilitates communication to stakeholders and allows for structured analysis of risks in subsequent steps. A risk scenario should articulate the following four (4) key elements:

- Asset - An object of value that has been identified
- Threat event - An attack event that has been identified
- Vulnerability - A weakness in the asset or processes supporting the asset that can be exploited by the identified threat event. The vulnerability may have surfaced in recent audits and/or penetration-tests or may be relevant to the environment due to the use of certain technologies.
- Consequence- The direct result of the threat event.

2.1.2 Risk Analysis

Risk analysis refers to the review of risks associated with the particular action or event. The risk analysis is applied to information technology, projects, security issues and any other event where risks may be analysed based on a quantitative and qualitative basis. Risks are part of every IT project and business organisations. The analysis of risk should be occurred on a regular basis and be updated to identify new potential threats. The strategic risk analysis helps to minimise the future risk probability and damage. The step 1 of the risk analysis process on the approach of S-RAF is to determine the likelihood.

2.1.2.1 Determine Likelihood

As a general guidance, the likelihood of cybersecurity risks should be assessed from the perspective of threats and vulnerabilities. One method to determine the cybersecurity risk likelihood is to consider the following factors:

- Discoverability - How easy would an adversary be able to discover the vulnerability of an asset? This is dependent on the availability of information about the vulnerability and the exposure of the vulnerable asset.
- Exploitability - How easy would an adversary exploit the vulnerability of an asset? This is dependent on the access rights, complexity of tools, as well as technical skills required to carry out the attack.
- Reproducibility - How easy would an adversary be able to reproduce the attack on the asset? This is dependent on the complexity of the exploit customisation and the environmental conditions required to carry out the attack.

2.1.2.2 Calculate Impact

In general, the manifestation of a risk scenario can compromise the confidentiality, integrity and/or availability of assets (e.g. data, equipment, operations). Any compromise of the assets will translate to adverse impact.

2.1.3 Risk Calculation

Each top-level criterion is represented with a tree of subcriteria. Criteria not composed of other criteria are assigned a score either by experts or based on a set of metrics. For example, the score of the financial gain criterion of risk may be estimated either directly by experts or based on monetarily measurable metrics.

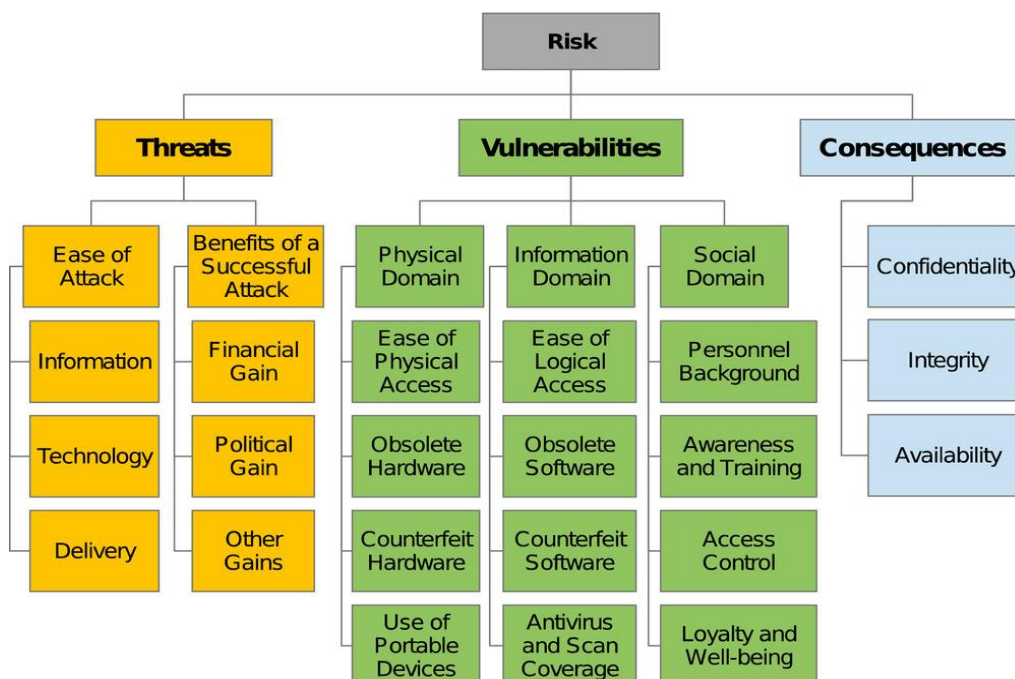


Figure 2: Risk Calculation Impact Categories

2.1.3.1 Determine and Prioritise Risk

Risk is a function of the likelihood of a given threat event exploiting a potential vulnerability of an asset and resulting impact. This can be diagrammatically presented using a risk matrix. The following Figure 3 below is a sample 5-by-5 risk matrix for determining risk level for each risk scenario, where risk level is a correlation of “Likelihood” and “Impact”, determined from the Risk Analysis conducted in the previous step.

Very High	5	7	8	9	10
High	4	6	7	8	9
Medium	3	5	5	7	8
Low	2	4	4	5	6
Very Low	1	2	3	4	5
	Rare	Unlikely	Possible	Likely	Very Likely

Figure 3: Risk Matrix

For each risk level derived, compare it against the risk tolerance level defined by the organisation. Risk scenarios with risk levels above the tolerance level must be prioritised for treatment until the risk levels fall to within the tolerance level. When prioritising risks for treatment, the expected duration should also be established.

2.1.3.2 Risk Report

A risk assessment is incomplete without documentation. The outputs from previous steps must be clearly documented in a Risk Report for communication to stakeholders. A Risk Register is a record of all the risk scenarios identified, including their determined risk level. The Risk Register is a living document to be regularly reviewed and updated to ensure that the organisation’s management has an up-to-date picture of the organisation’s cybersecurity risks when making risk-informed decisions. It should minimally contain the following:

- Risk scenario – A scenario articulating how a threat event could exploit a potential vulnerability of an asset to create an adverse impact.
- Identification date – The date when the risk scenario is identified.
- Existing measures – The current measures in place to address the risk scenario.
- Current risk – The determined risk level (combination of likelihood and impact) of risk scenario after taking into account existing measures.
- Treatment plan – The planned activities (e.g. deploying additional measures) and timeline to treat the current risk to an acceptable level (i.e. within organisation’s risk tolerance level).
- Progress Status – The status of implementing the treatment plan.

- Residual risk – The determined risk level (combination of likelihood and impact) of risk scenario after treatment plan is implemented (i.e. current risk with additional measures applied).
- Risk Owner – The individual or group responsible for ensuring that the residual risks remain within the organisation's tolerance level.

2.2 EPES Best Practices, Security Issues and Assessment

The starting point of the analysis has been a description of challenges in the energy sector that need to be addressed. In order to derive the challenges in the energy sector, high-level objectives have been agreed, which are expected to be common targets among all stakeholders in the energy sector. Today's energy infrastructure and market have been reflected against these high-level objectives and challenges have been derived accordingly. The challenges described are based on an operational viewpoint, i.e. they reflect challenges in daily operation but do not necessarily imply that support from a governmental authority is required to overcome these challenges, as some may be solved with other means.

Critical infrastructures provide essential services that underpin the smooth functioning of a modern society and serve as the backbone for the economic activities. These critical infrastructures include the energy, telecommunication, finance, health, and transport sectors. The energy infrastructure is arguably among one of the most complex and critical infrastructures as these other sectors depend upon it to deliver their essential services. Therefore, unavailability in supply of energy has a high potential impact on economy and proper functioning of the civil society that can last longer than the time of the incident itself. A potential disruption for a long period of time could affect the society, industry and trade with a high risk of impact on the modern society.

Digital technologies are playing an increasingly important role in the energy sector. A smarter energy system can perform power generation, transmission, network management and market related tasks with better precision and faster response times than a human-dependent system, thereby optimising energy management, prioritising usage, and setting policies for quick response to outages. Energy control systems include a hierarchy of interconnected physical and electronic sensing, monitoring, and control devices, mostly acting in real-time and typically connected to a central supervisory station or a control centre, but also extending up to customers with devices such as Smart Meters. Industrial control systems (ICS) encompass supervisory control and data acquisition (SCADA) systems used to monitoring and control operations that in case of energy transport and distribution networks are widely dispersed. Distributed control systems (DCS) are used for single facilities or small geographical areas. Control systems are connected to remote components such as remote terminal units (RTU) and programmable logic controllers (PLC) that monitor system data and initiate programmed control activities in response to input data and alerts. SCADA systems collect, display and store information from remotely located data collection transducers, sensors, control equipment's, devices and automated functions. The lack of real information and research on the EPES section portrays the huge gap between Cybersecurity and the aforementioned digital systems.

2.3 SDN-microSENSE Risk Assessment Framework (S-RAF)

For the scope of SDN-microSENSE we focused on creating an S-RAF solution which will differentiate from the already developed one and will complement the security of EPES infrastructure. In the rest

of this section we present a wrap-up of the methodology developed and the key points of differentiation between the developed solution and the methods described above.

2.3.1 SDN-microSENSE Risk Assessment methodology

The collaborative Energy Chain Risk Assessment (ECRA) methodology of SDN-microSENSE has been defined and presented in D3.1 [2], but for comprehension purposes it is briefly presented here as well.

Due to the distributed nature of the decentralised energy system, the ECRA methodology takes into account the collaborative aspects needed to involve all stakeholders (i.e. personnel at different places or task roles) of the energy components.

Additionally, this distributed nature makes it important to calculate the cumulative risk assessment. Cumulative manner considers the risk imposed to a specified asset as a product of the propagated risk which derives by the exploitation of a vulnerability of an asset that belongs in the same group of assets with the former one. **This risk assessment approach enables one to perceive the security state at the level of mission-critical assets that belong either in the same business workflow, or in the same physical (or virtual) networks.** In contrast, Individual risk refers to the risk imposed by an identified threat to a specific asset. This risk measurement is asset-centric as it enables a security specialist to focus on an asset of special interest and analyse its attack surface. This approach is of major importance when it comes to the protection of mission-critical assets and services in the EPES.

The methodology follows standardised notations and consists of seven steps (step 0 to step 6), which are presented below, along with a short description.

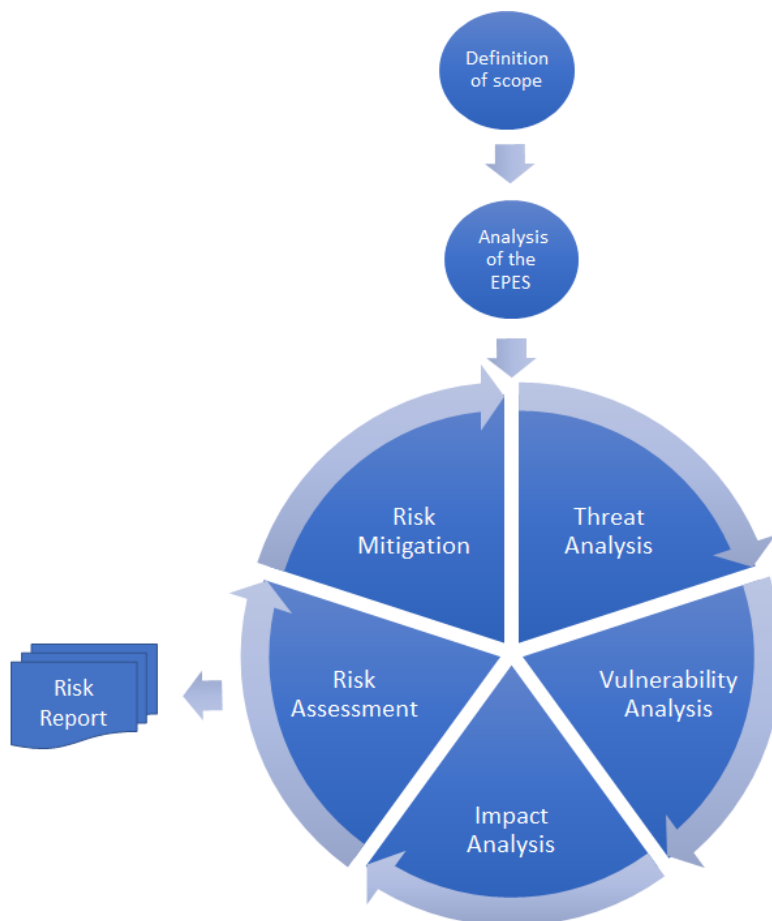


Figure 4: ECRA basic steps

- Step 0: Scope of the Energy Chain Risk Assessment (ECRA)

The initial step of the assessment, where the risk assessor selects the Energy Chain under consideration and defines the goal, the scope, and the outcome of the assessment.

- Step 1: Analysis of the EPES

The EPES under examination is decomposed, and its business processes and the cyber assets that comprise it (along with their interrelations) are defined and modelled using the SDN-microSENSE taxonomy (defined in D3.2 [3]).

- Step 2: EPES cyber threat analysis

Individual cyber Threats against the EPES cyber assets are identified based on Energy Chain participants expertise and knowledge, with usage of existing repositories of cyber threats.

- Step 3: Vulnerability Analysis

Vulnerabilities that exist in the cyber assets of the EPES are identified, based on data extracted from existing repositories of vulnerabilities and vulnerability scanners. Each individual vulnerability is first assessed and attack paths that would allow an attacker to reach one or more target assets

from one or more entry points/assets are discovered. An assessment is performed to calculate the cumulative vulnerability level of the attack paths.

- Step 4: Impact Analysis

The impact of the successful exploitation of each vulnerability is defined for individual assets and then the notion of cumulative impact is provided.

- Step 5: Risk Assessment

In this step, the methodology performs the risk assessment and computes the risk of individual assets and the commutative risk for asset chains.

- Step 6: Risk mitigation: Selection of security controls

Appropriate security actions are performed for mitigating attacks. To do so, security controls (extracted by guidelines, standards and concretely defined actions) will be selected and the risk will be recalculated.

2.3.2 OLISTIC as part of SDN-microSENSE Risk Assessment

OLISTIC is UBITECH's Enterprise Risk Management Suite and, among other complementary functionalities, is designed to offer an enterprise-scale cybersecurity risk management framework. In short, OLISTIC enables risk management across all operational domains of a company/organization and promotes the adoption of a security-by-design *modus operandi*.

OLISTIC acknowledges that an efficient and comprehensive cybersecurity risk management policy is the foundation of any proper business continuity plan. OLISITC builds upon the following principles:

Risk Assessment Quality is Key: The vast majority of existing cybersecurity risk management solutions model risk as a static property of each asset. OLISTIC provides a graph-based asset inventory module, so that to facilitate the visualization of asset interdependencies that can be used by attackers to cause damage.

Act Proactively, instead of Reactively: OLISTIC aims to provide the means to a security administrator of an organization to act proactively, instead of reactively. As there is no such thing as being 100% secure 100% of the time, being proactive pays off. OLISTIC innovates by offering an environment to serve as a security playground. Security and compliance officers can assess cyber risks and consider the existence and manage defensive strategies to retain confidentiality, integrity and availability risks at a minimal level.

Interoperability & Completeness: OLISTIC is built to be interoperable. Given the cybercrime epidemic, it embraces a best-of-breed approach to promote security-by-design. Its purpose is not to replace existing niche cybersecurity solutions (e.g. intrusion detection & prevention, anomaly detection, etc.). On the contrary, it acts complementary and offers an insightful risk analysis across all business processes of an organization.

The Risk Assessment Methodology is based on the following workflow:

1. Build the Asset Inventory

2. Define all Business Services (optional)
3. Assess the Cyber Risk
4. Evaluate the existence of different Defensive Strategies
5. Select and apply the optimal defensive strategy and repeat steps 4-5 (or 1-5) (unfortunately security is an iterative process).

Asset inventory: OLISTIC enables the analyst to treat every part of an organization as an asset (both tangible and intangible, such as servers, workstations, people etc.), as it capitalizes in a rather abstract model that offers great flexibility. In fact, OLISTIC allows the declaration of any, customer or domain-driven, arbitrary number of asset properties (e.g. multiple IP addresses, MAC addresses, tag numbers, product identifiers, manufacturer settings etc.) to offer a perfect fit to the peculiarities of the underlined infrastructure. Based on this approach, it employs a graph to model the asset inventory and describe all possible interdependencies among assets. Thus, OLISTIC deliver an asset-centric risk assessment approach. In addition, it syncs vulnerabilities, threats and controls from numerous well-known security standards and libraries (e.g. NIST CVE, ISO 27001 & 27005, OWASP, etc.), while it provides a generic and extensible system model capable of supporting any custom set of vulnerabilities, threats and controls. Based on the above, OLISTIC could be seen as an embedded asset management system. Asset Inventory is supported through OLISTIC's native asset editor, by using predefined CSV templates, or through the integration of vulnerability scanners, such as OpenVAS, in order to import assets, attributes and relationships.

Business Services: These are logical groups of assets that define the granularity in monitoring the cyber risk. This feature enables the analyst to mask organizational units, departments, and ERP business processes as business services and manage risks from an operational standpoint. OLISTIC supports unlimited business services, while an asset can be associated with multiple business services. OLISTIC ships with a default business service that refers to the asset inventory as a whole.

Risk Assessment: It calculates the cyber security risks for all assets or for an individual business service. OLISTIC offers analytical reporting dashboard functionalities that aim to assist a) IT people in addressing security issues and b) executives in having a broad overview of the risk within the organization. Each risk assessment execution is effectively a gap analysis between the current state and the desired security goal, while it also assists to monitor the security state of the monitored infrastructure in time. OLISTIC's risk engine is highly optimized and scales almost linearly to the asset inventory size. Through reporting dashboards, the analyst is in position to review the vulnerabilities and threats pending treatment and analyse vulnerabilities in accordance with their impact on confidentiality, integrity, and availability of each asset.

Defensive Strategies: OLISTIC offers a module for managing the consideration of defensive strategic for mitigating risks. In this way, the security analyst is able to confirm the status of security and privacy within the organization and estimate the distance towards the ultimate security goal. OLISTIC gives the ability to consider and evaluate an infinite number of defensive strategies based on controls from numerous well-known security standards and libraries. After evaluating the applicability of defensive actions, the security analyst repeats steps 4-5 (or 1-5) of the methodology described in order to maintain a timely overview of the cyber security risk level of the organisation.

The abovementioned operations form the basic functional structure of OLISTIC. The interoperable and flexible modelling of OLISTIC enables the adoption to several business environments, but its integration requires the development of additional features to fit to the peculiarities of each case. Thus, in the context of SDN-microSENSE, major updates have been triggered in order to address the need for a collaborative risk assessment framework for EPES.

1. Usage of updated model that supports EPES

The distributed nature of the EPES posed the requirement for the development of a methodology which considers the collaborative aspects and the involvement of multiple stakeholders (i.e., personnel at different places or task roles) in the risk assessment process. In this direction, and as will be highlighted also in the following listed items, several changes were triggered to OLISTIC's engine to support the adoption of the EPES ecosystem. The support of assets, considering of a wide range of legacy ICT and industrial devices, including older legacy SCADA and ICS devices, IoT components, and SDN assets required the extension of the inherent model. OLISTIC has been updated to become compatible with industrial and Power-specific standards for enabling risk assessment and the utilization of controls for the energy chain. Last but not least, several updates have been triggered to enable interoperable interaction with other tools. The aforementioned are a fraction of the changes made to OLISTIC's model for fitting to the EPES and energy chain. More specific points are listed below.

2. Collaborative Risk Assessment with the cumulative RA

In order to address the need for a collaborative risk assessment framework, we took advantage of the graph-based modelling of assets interdependencies used in OLISTIC to deliver a collaborative scheme of measuring the risk in a cumulative manner. More specifically, through the interdependency graphs, the risk assessment methodology has been extended to, not solely focus on measuring the risk for individual assets, but to uncover the risks which can be raised as a result of propagated threats or the ability of an attacker to penetrate further into the network. The collaborative aspect of the risk calculation is the utilisation of globally accepted standards such as CVSS. This option enabled the risk quantification to be compatible to the wide variety of legacy ICT and energy and SDN-specific assets.

3. Connecting with AIDB for EPES asset retrieval

As mentioned before, asset inventory is supported through OLISTIC's native asset editor or by using predefined CSV templates to import assets, asset attributes and relationships. Although those functionalities are important for managing the assets participating in the risk assessment process, these methods do not scale when the assessment methodology targets dynamic and distributed infrastructures like EPES. To this end, the asset inventory process has been extended to retrieve asset information from the Asset Inventory Database of the SDN-microSENSE architecture. By integrating this functionality to the already established asset management processes, S-RAF guarantees the always up-to-date overview of the infrastructure and, consequently, the accurate risk assessment.

4. Connection with eVul for automated analysis of vulnerabilities in the EPES environment

The dynamic asset inventory implies the existence of a dynamic process to offer enhanced observability to assets for identifying possible vulnerabilities. To do so, OLISTIC has been extended by integrating eVul tool for dynamically scanning the network assets and detect vulnerabilities. This feature supports and completes the inherent model that considers vulnerabilities from numerous well-

known security standards and libraries (e.g. NIST CVE, OWASP, etc.), while potentially being used interchangeably with the integrated OpenVAS vulnerability scanner.

5. Extending OLISTIC model and components for retrieving alerts from XL-SIEM

The model of OLISTIC has been extended in order to handle the input from threat detection tools. More specifically, S-RAF is able to dynamically consume alerts coming from the XL-SIEM deployed in the SDN-architecture. In this way, the dynamic asset inventory and vulnerability discovery is supported by the detection of threats against the assets. This feature offers a more detailed overview of the cyber risks and the security status of the EPES infrastructure.

6. Providing Incidents based on the Risk Assessment as output to SDN-SELF and ARIEC components of SDN-microSENSE

Given the extensions for dynamically managing multiple inputs sources, OLISTIC has been extended to deliver the Risk Assessment output to the SDN-SELF components, so that to base self-healing actions on evidences (i.e., incidents) and proceed to informed decisions. The consideration of information regarding the assets, the vulnerabilities and threats in the risk calculations and the calculation of cumulative risk due to attack propagation, can offer added value to components that exploit the output of S-RAF.

7. Integrating Apache Kafka as message queue that can be used by both internal and external components

In order enable interoperability with other tool, S-RAF adopts the utilisation of Apache KAFKA to act as an integration point and offer data persistency. In this way, a dynamic data pipeline is formed for sharing the outcome of S-RAF functions through a message queuing architecture.

8. Compatibility with MISP format

MISP is the prominent open standard for threat information sharing. In the context of SDN-microSENSE, upon the detection of offensive actions, the XL-SIEM will forward alarms to the CIS component. The CIS transforms the output of XL-SIEM into MISP format. OLISTIC has been extended for being compatible with MISP format so that to ensure interoperability with other architectural components.

2.3.3 SDN-microSENSE Risk Assessment Framework (S-RAF) Differentiations

In this section we highlight three key points of differentiation between the adoption of ECRA/S-RAF in comparison to other risk assessment solutions.

2.3.3.1 Cumulative Risk Assessment

Current tools assess the risks from threat events as a combination of likelihood and impact. The level of risk associated with identified threat events represents a determination of the degree to which organisations are threatened by such events. Organisations make explicit the uncertainty in the risk determinations, including for example, organisational assumptions and subjective judgments/decisions. Organisations can order the list of threat events of concern by the level of risk determined during the risk assessment—with the greatest attention going to high-risk events. Organisations can further prioritise risks at the same level or with similar scores. Each risk corresponds

to a specific threat event with a level of impact if that event occurs. In general, the risk level is typically not higher than the impact level, and likelihood can serve to reduce risk below that impact level. However, when addressing organisation-wide risk management issues with a large number of missions/business functions, mission/business processes and supporting information systems, impact as an upper bound on risk may not hold. For example, when multiple risks materialise, even if each risk is at the moderate level, the set of those moderate-level risks could aggregate to a higher level of risk for organisations. To address situations where harm occurs multiple times, organisations can define a threat event as multiple occurrences of harm and an impact level associated with the cumulative degree of harm. In order to tackle this lack of concrete methodology due to the difficulty of automating such process, we decided to add the cumulative part of risk assessment and calculation of risk. The calculation methodology has already been presented on deliverable D3.2 [3] and the formula used for the cumulative risk calculation as described also in section 4.5.

2.3.3.2 Risk Assessment with Focus on EPES

Finally, a critical infrastructure such as EPES contains many assets which may be cyber and physical. In order to infer the risk per asset, information regarding its vulnerabilities and impact level upon exploitation must be combined. The identification of such control elements constitutes the optimal defence strategy (Mitigation Strategy) tailored to the calculated cyber-risks. In the context of SDN-microSENSE, these controls basically reflect the controls identified from analysis of the standards in D3.1 [2]. Last but not least, the Risk levels of an asset is associated with the two variants of risk, namely the Individual Risk Level and the Cumulative Risk Level.

2.3.3.3 Deep Insights on the security issues of EPES

Through the inclusion of an engine that detects and manages vulnerabilities (eVul) and through integration with the other components of SDN-microSENSE (AIDB, XL-SIEM, output from EPES related honeypots), S-RAF is capable to provide also deep insights on the security issues, thus allowing administrators to monitor easily the security status of the examined EPES infrastructure.

3 SDN-microSENSE Risk Assessment Framework Architecture

In this section we provide information about the design of the S-RAF and how it works as part of the overall SDN-microSENSE framework.

3.1 Architecture Overview - Conceptual Architecture

This section presents how the components developed in this task fit in the general architecture of SDN-microSENSE, as presented in deliverable D2.3 [4].

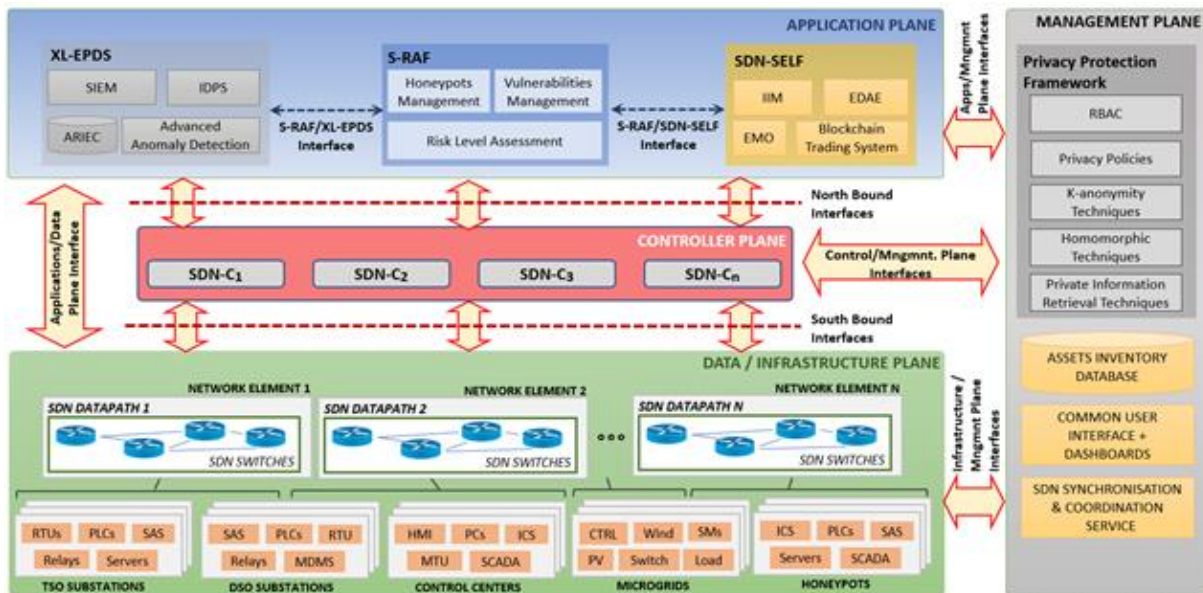


Figure 5: SDN-microSENSE Architecture Structural View

S-RAF is placed in the application plane and has been developed mainly in task 3.5, and is interacting with two other main components of SDN-microSENSE framework; the XL-EPDS responsible for identifying issues in the infrastructure, and the SDN-SELF that is responsible for the adaptations in the SDN fabric of the deployments.

Taking a closer look on the initial conceptual architecture, as presented in Figure 6, S-RAF includes the Risk Level Assessment and the Vulnerabilities Management components and also includes the Honeypot Manager component that has been developed in task 3.3. Honeypots and the Honeypot Manager are described in detail in D3.3 [10], and therefore will not be examined in this deliverable.

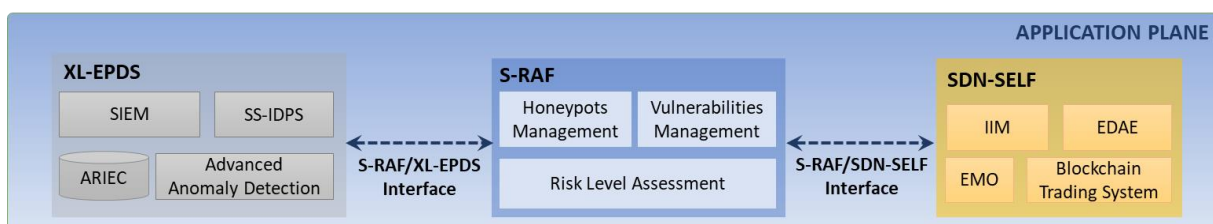


Figure 6: S-RAF in the Application Plane

Based on the work performed in WP3 and as the focus of this deliverable is S-RAF itself, we provide in Figure 7 below the internal conceptual architecture of S-RAF, along with the components of SDN-microSENSE that it has direct integration.

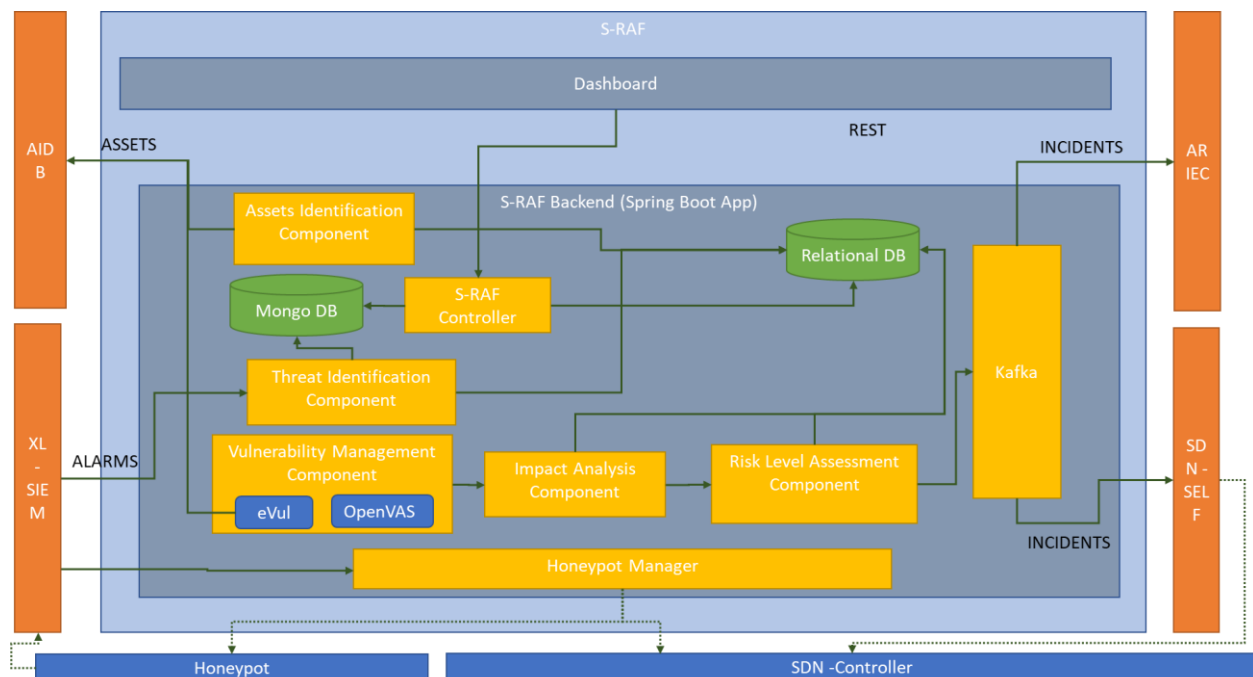


Figure 7: General view of the components and the relationships with other components

The internal components of S-RAF are presented in section 3.2. The S-RAF also interacts with other tools in the SDN-microSENSE ecosystem that are presented briefly below to assist the reader on understanding the design aspects of S-RAF.

- S-RAF communicates with the **Asset Inventory Database (AIDB)**: AIDB is a cross component which, not only supplies information to the Application/Management Plane and the Control / Management Plane layers, but also interacts with EPES in order to obtain information about Grid assets, and additionally SDN network assets. The planned AIDB usage by S-RAF is focussed on the query of network assets in order to get information about all hosts connected to the SDN network, regardless whether they present vulnerabilities or not.
- S-RAF communicates with **XL-EPDS** through **XL-SIEM** to gather alerts and their corresponding metadata. These alerts are sent through RabbitMQ.
- S-RAF provides to the **SDN-SELF (through the Electrical Data Analysis Engine (EDAE) component)** information regarding the vulnerable assets and gathered alerts from the XL-EPDS, enhanced with risk information.
- S-RAF provides to the Anonymous Repository of Incidents (**ARIEC**) information regarding the cybersecurity events, enhanced with risk information

3.2 Structural and Components View - Presenting S-RAF Components

In this section we describe the components of S-RAF and also the interfaces laying among them or with other external components.

3.2.1 Functional View: Presenting S-RAF Components

In this section we provide a description of the components of S-RAF with more details.

3.2.1.1 *Asset Identification Component*

This component is intended to store the assets (e.g. devices) that are connected to the EPES network and gathers as much information as possible from these devices.

S-RAF will employ a graph to model the asset inventory and describe all possible interdependencies among the discovered assets. In this way, a comprehensive mapping is generated for the multiple assets of the EPES network. S-RAF offers a generic and extensible system model capable of supporting any custom set of vulnerabilities, threats and controls, which are related to the identified assets. In other words, it allows the declaration of any, customer and use case driven, arbitrary number of asset properties (e.g. multiple IP addresses, MAC addresses, tag numbers, product identifiers, manufacturer settings etc.). As assets are described as a topology, it is important to be able to store the assets based on the idea of interdependency graphs, and thus being able to store relationships between the assets. For this reason, Neo4j [11] is used. Neo4j has a flexible structure defined by stored relationships between data records, as each data record, or node, stores direct pointers to all the nodes it is connected to.

The asset inventory can be created in different ways, mainly statically at this stage, but it is currently being integrated with the AIDB component for retrieving assets automatically.

3.2.1.2 *Threat Identification Component*

The threat identification component aims to identify the threats which are applicable to the EPES ecosystem. EPES is typically a mosaic of several technologies and hence, there is a wide range of threats that may target the components which may be distributed among the critical infrastructure and the power supply chain. The threat identification component uses a comprehensive dictionary of known patterns of attack employed by adversaries to exploit known weaknesses in cyber-enabled capabilities of EPES, and will enable the mapping between attack patterns and proper controls, which can be applied so as to minimise the risk level of EPES.

The threat identification component receives input from the XL-EPDS tools of the SDN-microSENSE architecture regarding security events and attacks that threaten the infrastructural assets.

3.2.1.3 *S-RAF Dashboard and Controller*

S-RAF Dashboard has been implemented using Thymeleaf [12] and the S-RAF Controller is responsible to provide all REST calls needed by the Dashboard of S-RAF. This is a core component of OLISTIC that has been extended for the purpose of S-RAF adaptations.

3.2.1.4 *Impact Analysis Component*

The impact analysis component defines a mapping from the three-security criteria (Confidentiality, Integrity and Availability) covered in the CVSS Impact metric onto the five-tier scale ranging from “Very Low” to “Very High”. This provides a single estimation for the overall impact of a specific

asset/vulnerability combination. Given that EPES is by-design a distributed ecosystem of diverse software and hardware components, the impact analysis component will be based on asset chains in order to estimate the cumulative impact that derives from the interconnections of the assets.

3.2.1.5 *Honeypots Management Component*

The Honeypot Manager and the whole integration of honeypots in SDN-microSENSE has been described in detail in deliverable D3.3 [10]. In short, the Honeypot Manager has two main functionalities in the SDN-microSENSE:

1. Provides means to develop in an automatic way those honeypots that the security operator or manager decide to deploy in the network.
2. Processes the information about unknown anomalies (like zero-day attacks) provided by the XL-SIEM and indicates to the SDN controller the required information to re-arrange the honeypots network in order to collect information for these unknown anomalies detected by the ML models of the XL-EPDS.

This component consists mainly in two modules:

- The **front-end** is composed by subcomponents in charge of presenting to the Security operator all the information that could be useful for managing the honeypots deployed in the EPES network. This module is a web application developed in HTML5 + CSS3 technologies. The selected scripting language to perform all the operations is AngularJS
- The **back-end** component oversees the execution of the two functionalities explained above. The Honeypot Manager's back-end consists of a Spring Boot application, managed by Maven, and contains the following modules:
 - o Database: A MySQL database is accessed using Spring JPA Repository technology.
 - o REST API: A definition of REST services using Spring REST technology.
 - o Security: All REST services are secured through a role-based system. In addition, to access each of the REST services it is necessary to be authenticated through JSON Web Tokens (JWT technology).

Therefore, design wise, since D2.3 [4] we consider Honeypot Manager as part of the overall S-RAF solution; however the communication with the other parts of S-RAF is performed through the integration with the XL-SIEM.

3.2.1.6 *Vulnerability Management Component*

The vulnerability Management Component is utilized to analyse the vulnerabilities of assets in the SDN-microSENSE infrastructure. This component capitalises on the use of eVul tool, offered by Ayesa, for detecting and managing vulnerabilities on hosts connected to the SDN network. The integration of the eVul Vulnerability manager aims to discover what network assets present vulnerabilities and may trigger the raise of risk level.

The risk level is based on the standard CVSS and depends on the vulnerability type, i.e., the exact CVE id. The Risk level is a number between 0 and 10, and can be classified following severity ranges, as shown in Table 1.

CVSS	Severity
0	Informative
(0, 4)	Low
[4,7)	Medium
[7,9)	High
[9,10]	Critical

Table 1: Vulnerability risk level ranges

Nonetheless, the vulnerability manager might intentionally change the risk level taking into account concrete hosts. The following diagram depicts the information returned by the Vulnerability Manager to the Impact Analysis component of the S-RAF risk assessment engine.

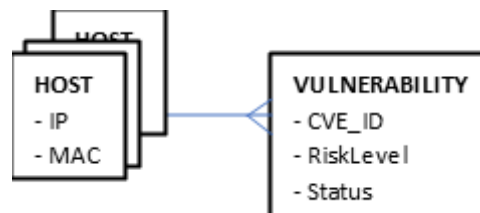


Figure 8: Vulnerability info

The vulnerability manager component creates a new instance workflow to follow up every detected vulnerability. Below, the vulnerability life cycle is depicted by a state diagram. The Vulnerability API will return active vulnerabilities.

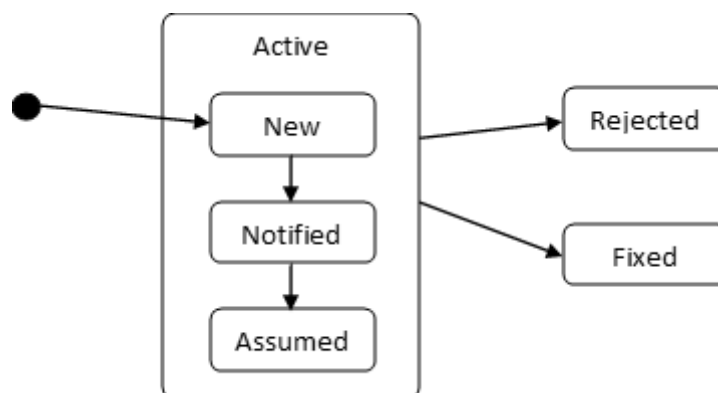


Figure 9: Vulnerability manager – Vulnerability life cycle

This component counts on two main modules:

- **Front-end.** It is a Web application based on HTML5, Java, Angular 7 technologies. This application allows a user to manage detected vulnerabilities through a workflow with different stakeholders complying a preconfigured BPM, configure notification and make follow up.
- **Back-end.** The back-end is based on a microservices architecture and it is integrated seamlessly with the GridPilot platform. Microservices devoted to eVul employs:
 - o **Database:** the RDMS PostgreSQL
 - o **External communications:** API REST, API manager WSO2.
 - o **Internal communication:**
 - synchronous by REST services
 - asynchronous by AMQP, RabbitMQ

On top of the competitive advantages gained by the eVul integration, S-RAF inherits the existence of OpenVAS vulnerability scanner, which comes as an integrated tool in OLISTIC. Both tools can complement each other and offer enhanced visibility to the SDN-microSENSE infrastructure.

3.2.1.7 Risk Level Assessment Component

This component aggregates the inputs of the above-mentioned components of the S-RAF in order to perform the final risk assessment. The mapping generated among all the assets in the EPES, the identified vulnerabilities, the attacks and the possible applied controls will be used to estimate the overall security risk.

3.2.1.8 Pub-Sub Queue (Kafka)

An extension performed on OLISTIC for facilitating easier integration with other components of the SDN-microSENSE. More specifically, in the Publish-Subscribe system, messages are persisted in a topic. Unlike point-to-point systems, consumers can subscribe to one or more topics and consume all the messages in that topic. In the Publish-Subscribe system, message producers are called publishers and message consumers are called subscribers. Apache Kafka can handle a high volume of data and enable one to pass messages from one endpoint to another. Kafka is suitable for both offline and online message consumption. Kafka messages are persisted on the disk and replicated within the cluster to prevent data loss. Hence, Kafka is used as an intermediate entity which guarantees the continuous and fault-tolerant message and data exchanging among the sub-components of the SDN-microSENSE architecture that need to consume information from the risk assessment process.

Kafka integration offers the following qualities to the S-RAF framework:

- **Reliability** – Kafka is distributed, partitioned, replicated and fault tolerant.
- **Scalability** – Kafka messaging system scales easily without down time. The pool of Producers and Subscribers can be extended effortlessly and can cover future integration needs of the project.
- **Durability** – Kafka uses distributed commit log which means messages persist on disk as fast as possible, hence it is durable.
- **Performance** – Kafka has high throughput for both publishing and subscribing messages. It will maintain a stable performance even if more components are going to be integrated in the final framework.

A stream of messages belonging to a particular category is called a topic. Producers are the publishers of messages to one or more Kafka topics. Producers send data to Kafka brokers, which are responsible

for maintaining the published data. Instead of storing all the messages of a partition in a single file, Kafka splits them into chunks called segments. Every time a producer publishes a message to a broker, the broker simply appends the message to the last segment file. Consumers read data from brokers. Consumers subscribe to one or more topics and consume published messages by pulling data from the brokers. The exact Kafka topics which will act as the actual integration endpoints with the rest tools will be documented in the context of the integration actions of WP7.

3.3 Integration View – Presenting S-RAF Interfaces

Based on the design and implementation aspects of S-RAF, the following interfaces were identified and implemented.

- eVul to Impact Analysis Component Interface
- AIDB to S-RAF Interface
- S-RAF to SDN-SEL Interface
- S-RAF to ARIEC Interface
- XL-SIEM to RAF Interface
- Controller to Dash Interface
- Risk Level Assessment Interface to Controller
- Impact Analysis to Risk Level Assessment Interface
- Assets Identification to Impact Analysis

These interfaces are presented below in detail, as agreed and implemented by the corresponding partners in order to facilitate the actual integrated version of S-RAF. It has to be mentioned that as the integration of S-RAF with other SDN-microSENSE components will take place in the forthcoming months in the scope of WP7, the interfaces and dependencies mentioned here between S-RAF and external SDN-microSENSE components (AIDB to S-RAF Interface, S-RAF to SDN-SEL Interface, S-RAF to ARIEC Interface, XL-SIEM to RAF Interface) might be updated or changed. The examples that accompany the interfaces definitions are based on fictional data and not on real assets and scenarios, due to privacy reasons.

3.3.1.1 eVul to Impact Analysis Component Interface

Name: EVUL-RAF			
Description	This interface aims to define the interaction between eVul and the core risk assessment engine of S-RAF. Note that, this interface refers to the internal integration of eVul to the vulnerability management component of S-RAF.		
Component providing the interface	eVul		
Consumer components or External Entities	Impact Analysis Component		
Type of Interface	REST		
Input data / Output Data	Methods or endpoints of the interface	Parameters of the method	Return Object or Values of the method
	Get active vulnerabilities per asset	assetID	Json object with vulnerabilities (see example below)
Constraints / Comments	AssetID should be known		
Responsibilities	AYESA		

Table 2: Details of the eVul-RA Interface

An example of the response of the RAF-eVUL interface is provided below.

```
[
  {
    "invExternalId": "0.0.0.1",
    "sdnIP": "0.0.0.1",
    "sdnMAC": "00:00:00:00:00:01",
    "vulnerability": [
      {
        "cve_id": "CVE-2013-7512"
        "cvss_score": 5.45
        "sdnPort": null
      },
      {
        "cve_id": "CVE-2014-1334"
        "cvss_score": 9.45
        "sdnPort": null
      }
    ]
  },
  {
    "invExternalId": "0.0.0.2",
    "sdnIP": "0.0.0.2",
    "sdnMAC": "00:00:00:00:00:02",
    "vulnerability": [
      {
        "cve_id": "CVE-2013-7512"
        "cvss_score": 5.45
        "sdnPort": null
      },
      {
        "cve_id": "CVE-2014-1334"
        "cvss_score": 9.45
        "sdnPort": null
      }
    ]
  }
]
```

3.3.1.2 AIDB to S-RAF Interface

Name: AIDB-RAF			
Description	This interface aims to define the interaction between the Asset Inventory Database (AIDB) and the core risk assessment engine of S-RAF. Note that, this interface refers to the integration of AIDB with the Asset Identification Component of S-RAF.		
Component providing the interface	AIDB		
Consumer components or External Entities	Asset Identification Component, eVul		
Type of Interface	REST		
Input data / Output Data	Methods or endpoints of the interface	Parameters of the method	Return Object or Values of the method
	/topology_query	N/A	[{ "invExternalId": "string", }]

			"installations": "Yes", "elements": "Yes", "downstream": "Yes", "topology": "Yes" }]
	/assets_inventory_query	N/A	{ "invExternalId": "string", "differentNet": "Yes", "border": "Yes", "invAssetType": "Yes" }
	* AIDB offers an extended API that exposed with Swagger ¹ , and in the scope of WP7 integration further REST calls might be used.		
Constraints / Comments	-		
Responsibilities	AYESA		

Table 3: Details of the AIDB-RAF Interface

An example of the response of the AIDB-RAF interface is provided below.

```
[
  {
    "invExternalId": "EX_SDN_CONTROLLER_UC0",
    "invExternalName": " EX_SDN_CONTROLLER_UC0",
    "invClass": " ELEMENT",
    "invNetLevel": "SDN_CONTROLLER_LEVEL",
    "invAssetType": "SDN Controller",
    "invState": "CONN_E",
    "invFather": "0.0.0.1",
    "invAttributeValues": [
      {
        "attribute": "sdnDriver",
        "value": "Northbound"
      },
      {
        "attribute": "sdnEndpoint",
        "value": "sdnEndpoint1"
      }
    ],
    "relationships": null
  },
  {
    "invExternalId": "000000000000000001",
    "invExternalName": "000000000000000001",
    "invClass": "ELEMENT",
    "invNetLevel": "SDN_SWITCH_LEVEL",
    "invAssetType": "SDN_SWITCH",
    "invState": "CONN_E",
    "invFather": null,
    "invAttributeValues": [
      {
        "attribute": "InvDescription",
        "value": null
      }
    ]
  }
]
```

¹ <https://swagger.io/>

```
[
  {
    "attribute": "sdnManufacturer",
    "value": "Nicira, Inc."
  },
  {
    "attribute": "sdnSwDesc",
    "value": "2.3.90"
  }
],
"relationships": [
  {
    "relationshipId": "1",
    "externalIdA": "0000000000000001",
    "externalIdB": "0000000000000002",
    "enable": true,
    "weight": 1.0
  },
  {
    "relationshipId": "2",
    "externalIdA": "0000000000000002",
    "externalIdB": "0000000000000001",
    "enable": true,
    "weight": 1.0
  },
  {
    "relationshipId": "3",
    "externalIdA": "0000000000000001",
    "externalIdB": "0000000000000003",
    "enable": true,
    "weight": 1.0
  },
  {
    "relationshipId": "24",
    "externalIdA": "0000000000000003",
    "externalIdB": "0000000000000001",
    "enable": true,
    "weight": 1.0
  },
  {
    "relationshipId": "4",
    "externalIdA": "0000000000000001",
    "externalIdB": "0000000000000004",
    "enable": true,
    "weight": 1.0
  },
  {
    "relationshipId": "5",
    "externalIdA": "0000000000000004",
    "externalIdB": "0000000000000001",
    "enable": true,
    "weight": 1.0
  }
]
]
```

3.3.1.3 S-RAF to SDN-SELF Interface

Name: RAF-EDAE

Description	This interface aims to define the interaction between the core risk assessment engine of S-RAF and EDAE tool from the SDN-SELF. This interface aims to make available the risk assessment output to the self-healing component through a Kafka data pipeline.		
Component providing the interface	S-RAF (Kafka)		
Consumer components or External Entities	SDN-SELF (ADAE)		
Type of Interface	Kafka		
Input data / Output Data	Methods or endpoints of the interface	Parameters of the method	Return Object or Values of the method
	sraf.out.adae	None. It is a pub sub interface.	Json object with output of alarms and risk assessment results (see example below)
Constraints / Comments	-		
Responsibilities	UBITECH		

Table 4: Details of the RAF-EDAE Interface

An example of the response of the RAF-EDAE interface is provided below.

```
{
  "timestamp": "2020-07-15T12:34:28.743+0000",
  "xl_siem_rule": {
    "crit_level": 3,
    "description": "Nmap Scanning.",
    "sourceIP": "0.0.80.7",
    "firedtimes": 5,
    "groups": [
      "pam",
      "syslog"
    ],
    "assets": [
      "asset_00001",
      "asset_00002"
    ]
  },
  "risk_assessment": {
    "ass1": {
      "ip": "0.0.0.1",
      "id": "asset_00001",
      "previous_risk": "L",
      "current_risk": "M",
      "vulnerabilities": [
        "CVE-2019-1347",
        "CVE-2019-1326"
      ],
      "dns_name": "asset1_name"
    },
    "ass2": {
      "ip": "0.0.0.2",
      "id": "asset_00002",
      "previous_risk": "VL",
      "current_risk": "M",
      "vulnerabilities": [
        "CVE-2019-1293"
      ],
      "dns_name": "asset1_name"
    }
  },
  "historical_risk_data": {
```

```

"previous_risk_level":"M",
"current_risk_level":"H"
},
"location":"/Path/To/JSON"
}

```

3.3.1.4 S-RAF to ARIEC Interface

Name: RAF- ARIEC			
Description	This interface aims to define the interaction between the core risk assessment engine of S-RAF and ARIEC. This interface aims to make available the risk assessment output that complements the identified security events of XL-EPDS components to ARIEC.		
Component providing the interface	S-RAF (Kafka)		
Consumer components or External Entities	ARIEC		
Type of Interface	Kafka		
Input data / Output Data	Methods or endpoints of the interface	Parameters of the method	Return Object or Values of the method
	sraf.out.irec	None. It is a pub sub interface.	<ul style="list-style-type: none"> Assets (from AIDB) Vulnerabilities on assets (eVul) Detected events against assets (from XL-SIEM through CIS. Note that this includes all the information as acquired from XL-SIEM (e.g. source IP of attack)) Individual Risk per Asset - including previous state Cumulative risk (including attack paths)
Constraints / Comments	As ARIEC implementation described in D5.5 is considered Classified Information only a high-level description of the agreed interface is provided in this document.		
Responsibilities	UBITECH		

Table 5: Details of the RAF-ARIEC Interface

3.3.1.5 XL-SIEM to RAF Interface

Name: CIS-RAF	
Description	Atos CIS uses this interface, supported by a RabbitMQ server, to export alarms triggered by the XL-SIEM after being transformed to MISP format.
Component providing the interface	Atos CIS

Consumer components or External Entities	Threat Identification Component		
Type of Interface	RabbitMQ		
Input data / Output Data	Methods or endpoints of the interface	Parameters of the method	Return Object or Values of the method
	Exchange queue to read MISP alarms: atos.exchange.alarms.sd nmsense.cis	None. It is a push interface	
Constraints / Comments	-		
Responsibilities	ATOS		

Table 6: Details of the CIS-RAF Interface

An example of the response of the CIS-RAF interface is provided below.

```
{
  "Event": {
    "id": "176565",
    "orgc_id": "1",
    "org_id": "1",
    "date": "2020-08-05",
    "threat_level_id": "1",
    "info": "Denial of Service alarm",
    "published": false,
    "uuid": "5f46xxx6-xxxx-xxx-xxx-6e060axxxx",
    "attribute_count": "7",
    "analysis": "2",
    "timestamp": "1598427862",
    "distribution": "0",
    "proposal_email_lock": false,
    "locked": false,
    "publish_timestamp": "0",
    "sharing_group_id": "0",
    "disable_correlation": false,
    "extends_uuid": "",
    "event_creator_email": "ruben.trapero@atos.net",
    "Org": {
      "id": "1",
      "name": "Atos ARI CS Lab",
      "uuid": "5a579708-xxxx-x-xx-12xxxxc8"
    },
    "Orgc": {
      "id": "1",
      "name": "Atos ARI CS Lab",
      "uuid": "5a579708-xxx-x-xx-12xxxxc8"
    },
    "Attribute": [
      {
        "id": "5070300",
        "type": "ip-src|port",
        "category": "Network activity",
        "to_ids": false,
        "uuid": "5f4612d6-xxx-xxx-xxxx-6e06xxx020f",
        "event_id": "1765565",
        "distribution": "5",
        "timestamp": "1598427862",
        "comment": "Source IP and port associated to the detected alarm.",
        "sharing_group_id": "0",
        "deleted": false,

```

```
"disable_correlation":true,
"object_id":"0",
"object_relation":null,
"value":"0.0.0.0|111",
"Galaxy":[

],
"ShadowAttribute":[

]
},
{
  "id":"5070301",
  "type":"ip-dst|port",
  "category":"Network activity",
  "to_ids":false,
  "uuid":"5f4612d6-xxxxxx-xxxx-xxxx-6e06xxx020f ",
  "event_id":"1765565",
  "distribution":"5",
  "timestamp":"1598427862",
  "comment":"Destination IP and port associated to the detected alarm.",
  "sharing_group_id":"0",
  "deleted":false,
  "disable_correlation":true,
  "object_id":"0",
  "object_relation":null,
  "value":"0.0.0.0|397",
  "Galaxy":[

],
  "ShadowAttribute":[

]
},
{
  "id":"5070302",
  "type":"other",
  "category":"External analysis",
  "to_ids":false,
  "uuid":"5f4612d7-xxx-xxx-xxx-6e06xxx020f ",
  "event_id":"1765565",
  "distribution":"5",
  "timestamp":"1598427863",
  "comment":"Risk value evaluated by XL-SIEM",
  "sharing_group_id":"0",
  "deleted":false,
  "disable_correlation":true,
  "object_id":"0",
  "object_relation":null,
  "value":"8",
  "Galaxy":[

],
  "ShadowAttribute":[

]
},
{
  "id":"507033",
  "type":"other",
```

```
"category": "External analysis",
"to_ids": false,
"uuid": "5f4612d7-xx-xxx-xxxx-6exxxx020f",
"event_id": "1765565",
"distribution": "5",
"timestamp": "1598427863",
"comment": "Priority value evaluated by XL-SIEM",
"sharing_group_id": "0",
"deleted": false,
"disable_correlation": true,
"object_id": "0",
"object_relation": null,
"value": "5",
"Galaxy": [

],
"ShadowAttribute": [

]
},
{
  "id": "500304",
  "type": "other",
  "category": "Internal reference",
  "to_ids": false,
  "uuid": "5f4612d7-xxxx-xxxx-xxx-6exxx020f",
  "event_id": "1765565",
  "distribution": "5",
  "timestamp": "1598427863",
  "comment": "Organization where the XL-SIEM Agent has been deployed",
  "sharing_group_id": "0",
  "deleted": false,
  "disable_correlation": true,
  "object_id": "0",
  "object_relation": null,
  "value": "ATOS",
  "Galaxy": [

],
  "ShadowAttribute": [

]
},
{
  "id": "507305",
  "type": "other",
  "category": "Other",
  "to_ids": false,
  "uuid": "5f4612d7-xxxx-xxxx-xxxx-6e0xxx20f",
  "event_id": "1765565",
  "distribution": "5",
  "timestamp": "1598427863",
  "comment": "Userdata1",
  "sharing_group_id": "0",
  "deleted": false,
  "disable_correlation": true,
  "object_id": "0",
  "object_relation": null,
  "value": "SURICATA",
  "Galaxy": [
```

```
    ],
    "ShadowAttribute":[
    ]
  },
  {
    "id":"507306",
    "type":"other",
    "category":"Other",
    "to_ids":false,
    "uuid":"5f4xx2d7-xxxx-xxx-xxx-6e060xxx020f",
    "event_id":"1765565",
    "distribution":"5",
    "timestamp":"1598427863",
    "comment":"Userdata2",
    "sharing_group_id":"0",
    "deleted":false,
    "disable_correlation":true,
    "object_id":"0",
    "object_relation":null,
    "value":"Denial of service event detected by Suricata",
    "Galaxy":[
    ],
    "ShadowAttribute":[
    ]
  }
],
"ShadowAttribute":[

],
"RelatedEvent":[

],
"Galaxy":[
],
"Object":[
],
"Tag":[
  {
    "id":"18",
    "name":"xl-siem:category=\"info\"",
    "colour":"#3a0a00",
    "exportable":true,
    "hide_tag":false,
    "user_id":"0",
    "numerical_value":null
  },
  {
    "id":"57",
    "name":"xl-siem:sub-category=\"misc\"",
    "colour":"#571000",
    "exportable":true,
    "hide_tag":false,
    "user_id":"0",
    "numerical_value":null
  }
]
}
```

3.3.1.6 Controller to Dash Interface

Name: CONT-DASH			
Description	This interface aims to define the interaction between the S-RAF dashboard and the S-RAF controller residing at the backend. This REST interface enables a variety of actions for controlling the risk assessment workflow.		
Component providing the interface	S-RAF controller		
Consumer components or External Entities	Dashboard		
Type of Interface	REST		
Input data / Output Data	Methods or endpoints of the interface	Parameters of the method	Return Object or Values of the method
	*		
Constraints / Comments	* S-RAF controllers includes all REST interfaces required for the rendering of the UI in the dashboard, and facilitate the interaction with the other components and the repository related functions. Not provided in detail due to space and privacy limitations.		
Responsibilities	UBI		

Table 7: Details of the CONT-DASH Interface

3.3.1.7 Risk Level Assessment Interface to Controller

Name: RLA-CONT			
Description	This interface aims to define the interaction between the Risk Level Assessment Component and the S-RAF controller. This is an internal interface that enables the triggering of the risk assessment process.		
Component providing the interface	Risk Level Assessment Component		
Consumer components or External Entities	S-RAF Controller		
Type of Interface	REST		
Input data / Output Data	Methods or endpoints of the interface	Parameters of the method	Return Object or Values of the method
	POST /api/v1/riskassessment		Risk assessment id
	DELETE /api/v1/riskassessment	Risk assessment id	Html Response Code
	POST /api/v1/riskassessment/{id}/execute	Risk assessment id	Json object with risk assessment
	POST /api/v1/riskassessment/{id}/asset/{aid}/threat/{tid}	Risk assessment id, asset id	Html Response Code
	POST /api/v1/riskassessment/{id}/asset/{aid}/threat/{tid}	Risk assessment id, asset id, thread id	Html Response Code
	POST /api/v1/riskassessment/	Risk assessment id, asset id, vulnerability id	Html Response Code

	{id}/asset/{aid}/vulnerability/{vid}		
	POST /api/v1/riskassessment/ /{id}/asset/{aid}/control/ {cid}	Risk assessment id, asset id, control id	Html Response Code
	POST /api/v1/riskassessment/ /{id}/asset/{aid}/vulnerability/{vid}	Risk assessment id, asset id, vulnerability id	Html Response Code
Constraints / Comments	Main functions were presented		
Responsibilities	UBI		

Table 8: Details of the RLA-CONT Interface

3.3.1.8 Impact Analysis to Risk Level Assessment Interface

Name: IMP-RLA			
Description	This interface aims to define the interaction between the Impact Analysis Component and the Risk Level Assessment Component. This is an internal interface that enables the triggering of the risk assessment process given the analysis performed on the impact of identified threats and vulnerabilities.		
Component providing the interface	Impact Analysis Component		
Consumer components or External Entities	Risk Level Assessment Component		
Type of Interface	REST		
Input data / Output Data	Methods or endpoints of the interface	Parameters of the method	Return Object or Values of the method
	POST /api/v1/riskassessment/ /{id}/discoverattackpaths	Risk assessment id	Json with Attack path
	POST /api/v1/riskassessment/ /{id}/analyzeattackpaths	Risk assessment id	Json with Attack path
Constraints / Comments	-		
Responsibilities	UBI		

Table 9: Details of the IMP-RLA Interface

3.3.1.9 Assets Identification to Impact Analysis

Name: ASI-IMP	
Description	This interface aims to define the interaction between the Asset Identification component and the Impact Analysis Component. This is an internal interface that enables the impact analysis to consider the interdependencies of assets in order to calculate the impact of cascading effects on asset chains.
Component providing the interface	Assets Identification

Consumer components or External Entities	Impact Analysis Component		
Type of Interface	Java Method		
Input data / Output Data	Methods or endpoints of the interface	Parameters of the method	Return Object or Values of the method
	RetrieveAssets	Asset	List of Assets
Constraints	-		
Responsibilities	UBI		

Table 10: Details of the ASI-IMPInterface

3.4 Behavioural View

The planned integration with S-RAF components was initially defined in deliverable D2.3 [4], and is presented in Figure 10 below.

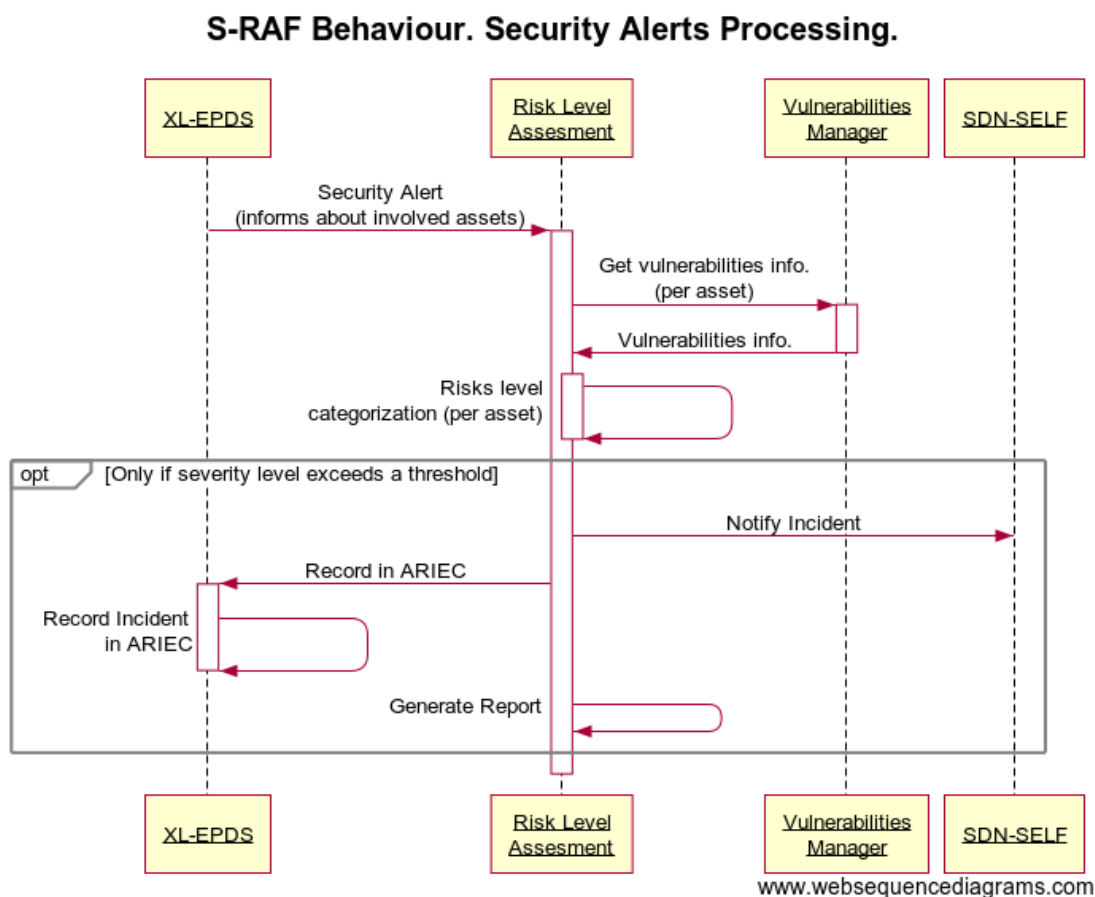


Figure 10: S-RAF basic behaviour as defined in D2.3

Based on the work performed since D2.3 and the more precise definition of the components of S-RAF and interaction between them and also the interaction with the external components, as presented in sections 3.3, we provide the detailed interaction on basic risk assessment scenarios. As can be seen also from the scenarios that are presented below, although the triggering event for assessing the risk

may differ, the impact analysis components requires and updated view on the vulnerabilities, the threats and the assets with their interdependencies.

For example, Figure 11 depicts the sequence of events that happens when a new asset is discovered or an existing is edited.

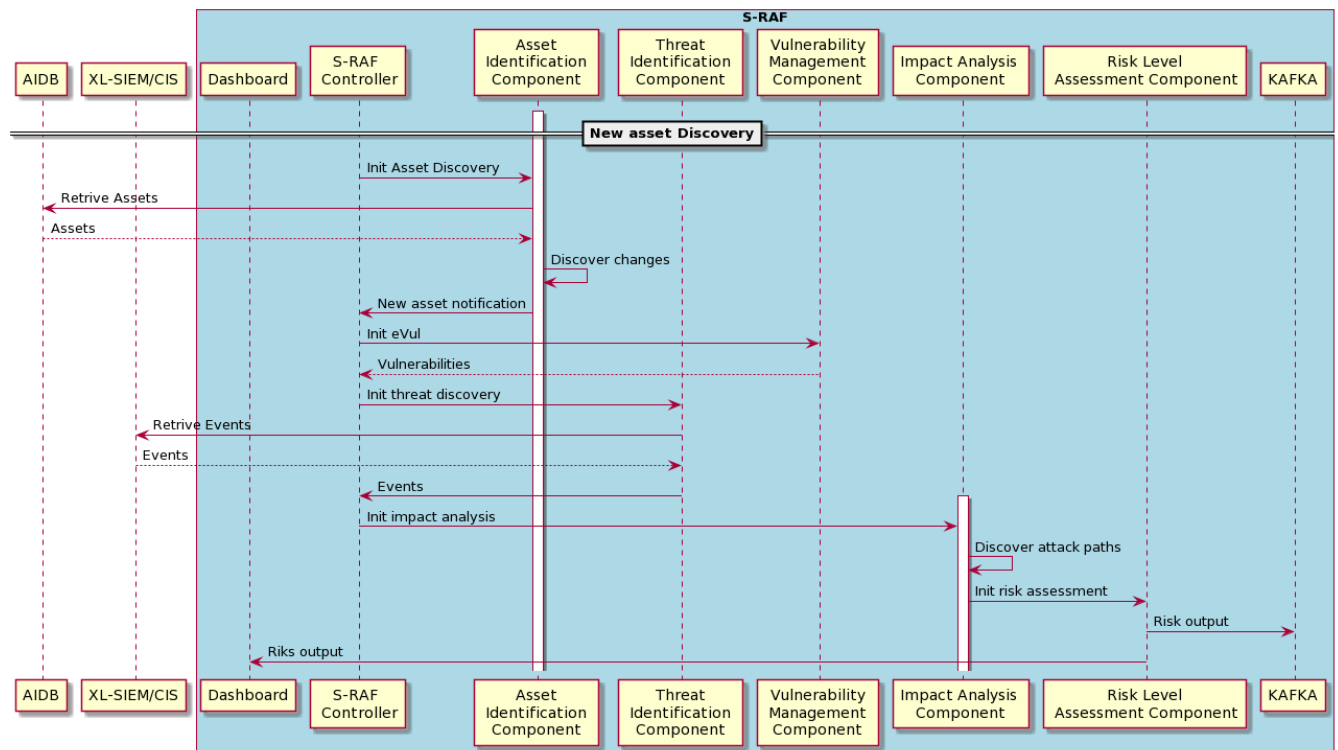


Figure 11: Detailed Sequence diagram of S-RAF – New asset discovery scenario

In a similar manner, Figure 12 depicts the sequence of events that happens when a new vulnerability is discovered in an asset.

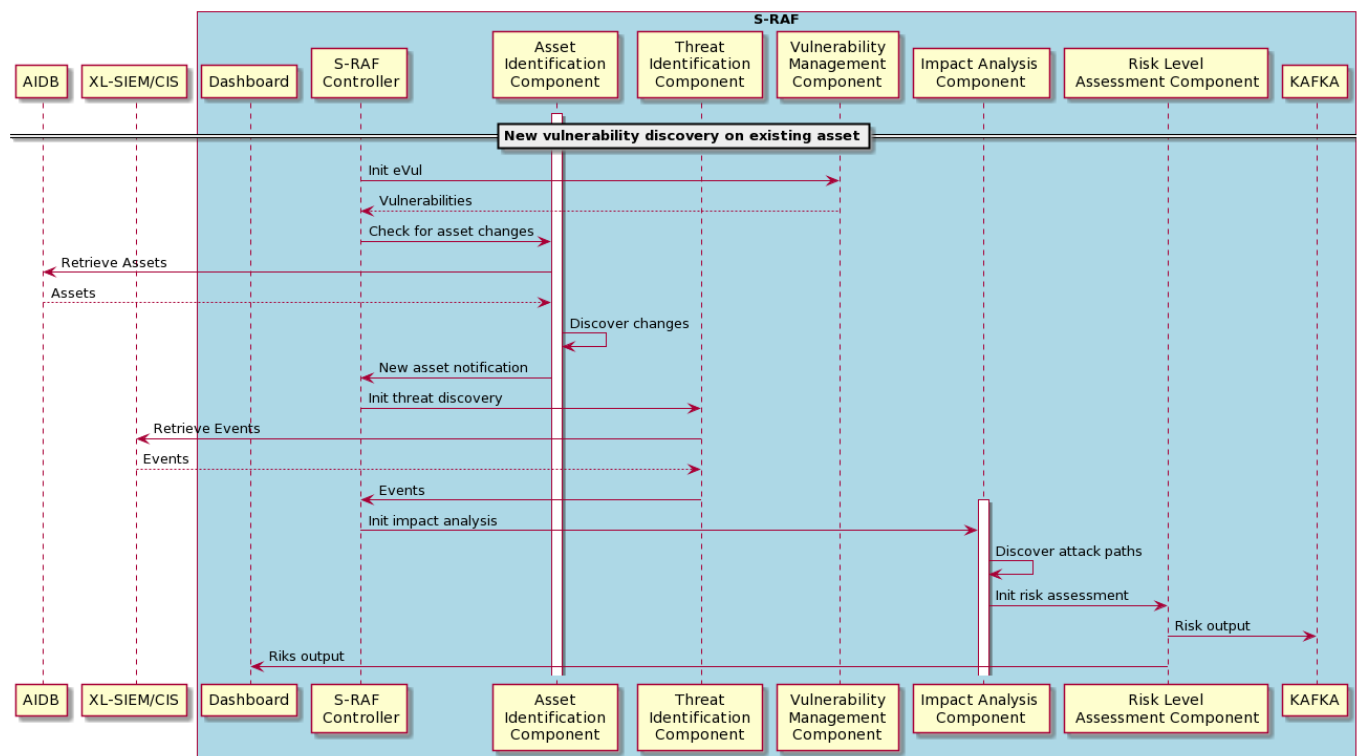


Figure 12: Detailed Sequence diagram of S-RAF – New vulnerability discovery scenario

When a vulnerability is patched, risk must be re-assessed as depicted in the sequence diagram of Figure 13.

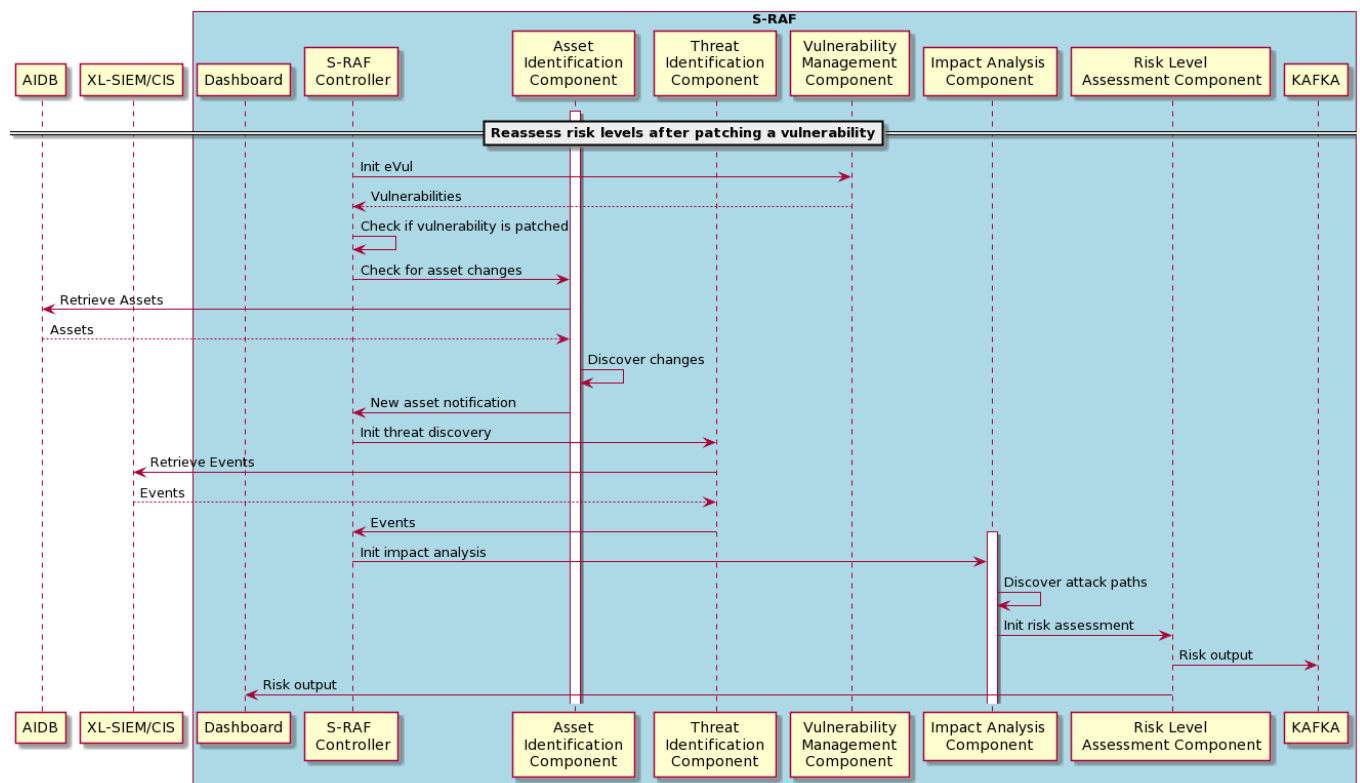


Figure 13: Detailed Sequence diagram of S-RAF – Re-assessment of risk scenario

Finally, Figure 14 depicts the sequence of events that happens when a new alarm is provided by XL-SIEM.

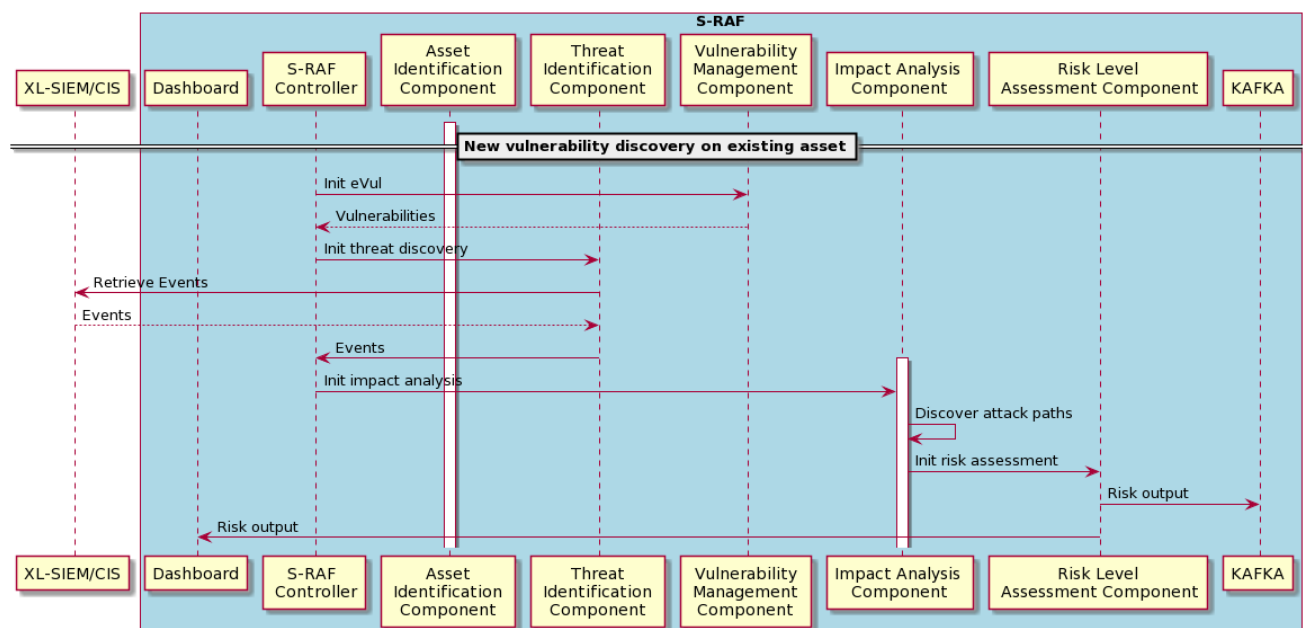


Figure 14: Detailed Sequence diagram of S-RAF – Risk assessment due to alarm from XL-SIEM

4 SDN-microSENSE Risk Assessment Framework Implementation Details

In this section some insights about the implementation of S-RAF are provided, by presenting information on the database schemas, used technologies and the compilation process. In addition, the implementation of the risk assessment calculation is described. Again, it has to be noted that no restricted data of the platform will be provided; the goal however of this section is to provide the reader a better understanding of the developed solution.

4.1 Technologies and Standards

For the implementation of S-RAF, UBITECH's OLISTIC Risk Assessment Framework has been used, so S-RAF has been built using Java 8 and Spring Boot Framework, while for the dashboard, Thymeleaf [12] template engine and has been used. For the storage, both a relational database (MySQL) and a document-oriented database (MongoDB) are utilized. The asset modelling component uses Neo4j [11] for the realisation of the interdependency graphs.

In addition, for enabling interoperability with other tools of the SDN-microSENSE architecture, S-RAF is compatible with open standards for threat information sharing and handles received inputs of indicators of compromise under MISP format. For sharing the output of the risk assessment operation, S-RAF integrates Apache KAFKA [13] to realise a publish/subscribe model and ease the integration with tools that could take advantage of the risk assessment output (e.g., SDN-SELF components).

As presented in Section 3, S-RAF also includes eVUL, a tool owned by AYESA, and is part of the GridPilot platform of AYESA. eVUL uses Java, PostgreSQL [14], REDIS [15] and Neo4j [11].

For the easier deployment of S-RAF components Docker can be used.

4.2 S-RAF Implementation

This section offers implementation details for S-RAF. S-RAF extends UBITECH's OLISTIC Risk Assessment Framework and integrates the eVul tool to enable vulnerability management. The reader can refer to section 2.3.2 to have an overview of the notable extensions of OLISTIC in the context of SDN-microSENSE.

S-RAF acts as an intermediate entity between the tools that aim to detect and log cybersecurity events against the SDN-microSENSE infrastructure, and tools that undertake self-healing actions to increase the resilience of the SDN-based Energy Ecosystem. Having said that, S-RAF utilises several models of storing input information and feed the risk assessment operation. That is, section 4.2.2 provides an overview of the database schema used in S-RAF that reveals the core entities and the corresponding relations.

In addition, S-RAF exposes a REST interface for enabling the communication between the S-RAF dashboard and the backend components. Section 4.2.1 exposes the code structure of S-RAF and offers an overview of the REST controllers used for supporting the risk assessment workflow. Finally, the section details on the compilation and deployment process of S-RAF components, including the deployment of Apache KAFKA which is used as an integration point for tools that exploit the output of the risk assessment process.

4.2.1 Packaging and Code structure

In this section we present an overview of the code structure of S-RAF, and provide a mapping to the architectural subcomponents as explained in section 3. In Figure 15, the packaging of S-RAF is depicted.

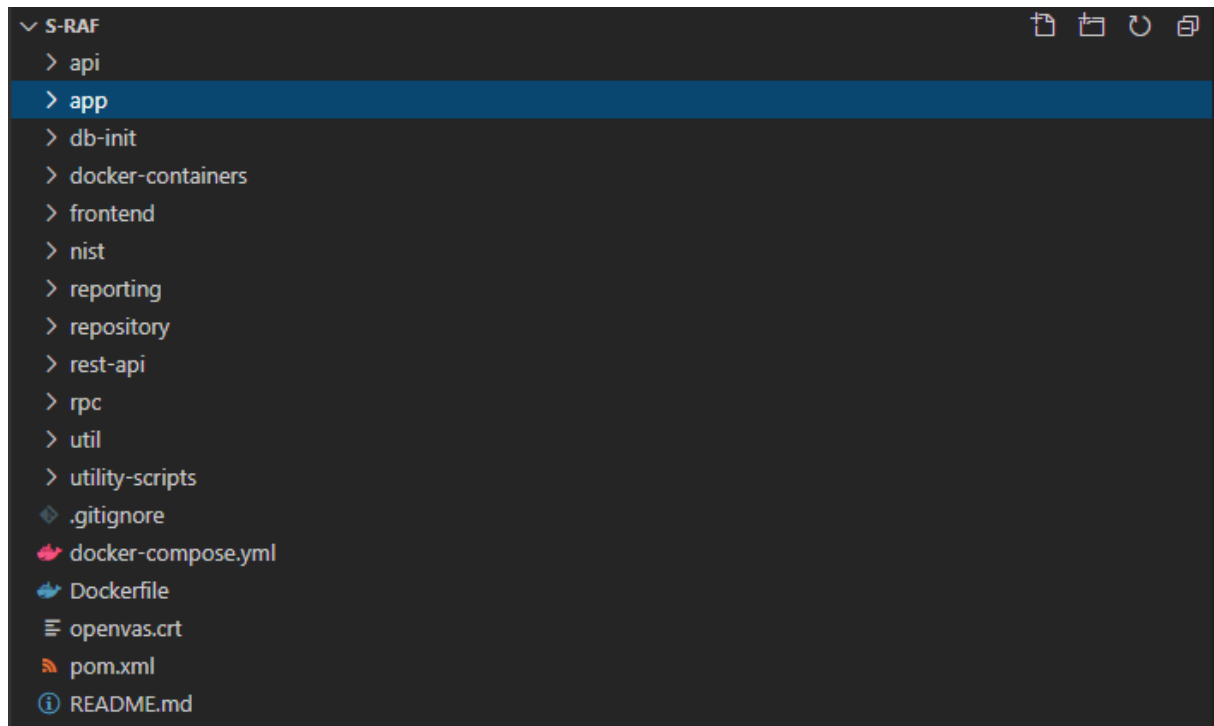


Figure 15: Packaging of S-RAF code

These folders include mainly the code of OLISTIC that is used and adapted for the scope of SDN-microSENSE, docker containers for any additional service needed, such as databases (MySQL, Mongo and Neo4j Graph database), eVul and Kafka. All the services and S-RAF are deployable with the same docker compose file (see Annex I – Docker Compose for S-RAF installation). In the following Figure 16, the code modules of the core application of S-RAF is presented.

```
project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">

  <modelVersion>4.0.0</modelVersion>
  <groupId>io.sraf</groupId>
  <artifactId>parent</artifactId>
  <packaging>pom</packaging>
  <version>0.1.0-SNAPSHOT</version>
  <name>sraf</name>
  <url>http://www.sdnmicrosense.eu/</url>
  <description>S-RAF Components</description>

  <!--Used to generate WAR for Tomcat deployment -->
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.5.6.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>

  <modules>
    <!-- Smart Controller artefacts should be compiled in the following order -->

    <!-- API is always the first module since all potential reference implementations import it -->
    <module>api</module>
    <!-- Module that encapsulate general purpose static libraries related to security, Kafka/RabbitMQ communication, etc-->
    <module>util</module>
    <!-- Reporting functionality -->
    <module>reporting</module>
    <!-- The data access layer of the project-->
    <module>repository</module>
    <!-- The module which is responsible for efficient search -->
    <module>indexer</module>
    <!-- Rest API has to be compiled just before the app since it used all RIs -->
    <module>rest-api</module>
    <!-- GRPC-based communication module between sraf and peripherals -->
    <module>rpc</module>
    <!-- The S-RAF Controller application that creates the actual backend together; should be the last one-->
    <module>app</module>
    <!-- Module that includes all the project's frontend assets -->
    <module>frontend</module>
    <!-- Testing module-->
    <module>nist</module>
  </modules>
```

Figure 16: Software modules as part of the main .pom file of S-RAF

The “api”, “util”, and “reporting” modules are providing the backbone part of the module “app” that builds the whole backend of the S-RAF application (and includes classes for the Assets Identification, Threat Identification, Impact Analysis, Risk Level Assessment components in relation to Figure 7 that presented the conceptual architecture of S-RAF). Updates were made on all the modules, with the most important for the scope of the deliverable being the implementation of the cumulative risk assessment logic, as presented in section 4.5.1.2. The “repository” module includes the data access layer of S-RAF to the databases (MySQL, MongoDB, Neo4j). The final part of the backend of S-RAF is the “rest-api” and the “rpc” modules. For the context of S-RAF updates were made only to the “rest-api” part as we currently didn’t need to extend any rpc based communication. Finally, the “frontend” module is responsible for building the actual dashboard of S-RAF.

4.2.2 Data Model Highlights

In the following part of the deliverable we provide an overview of some parts of the relational database, with the purpose of assisting the user to understand the way that the risk assessment is performed. Firstly, in Figure 17 we present the assets as defined in the S-RAF.

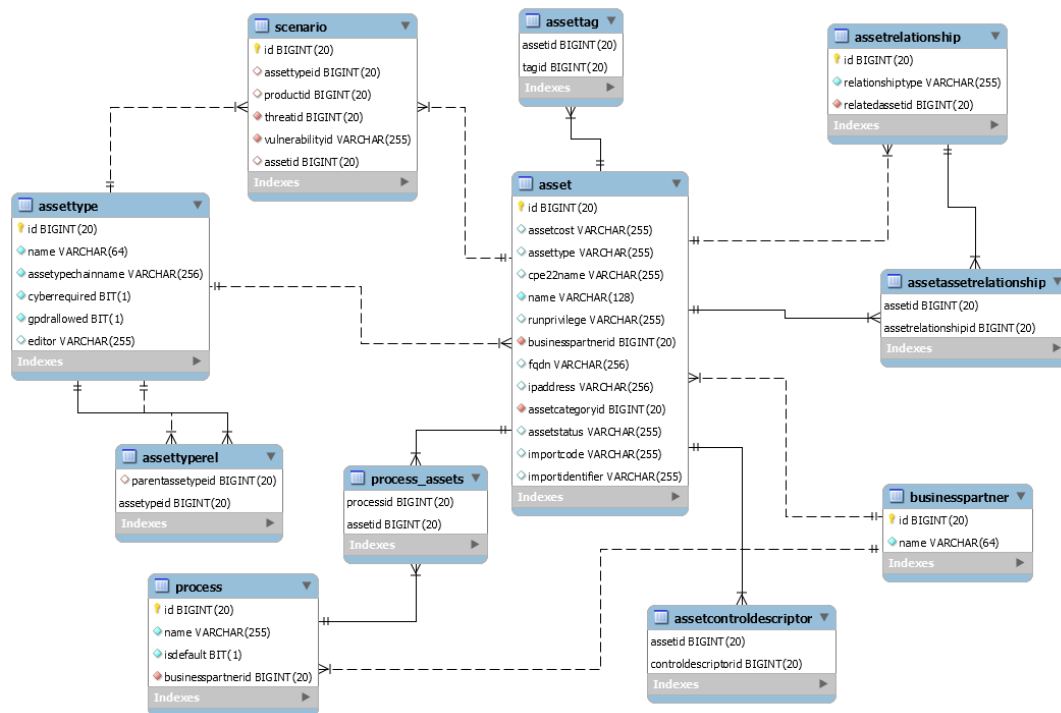


Figure 17: Assets Representation

Assets are the core concept of S-RAF risk assessment as everything that needs to be included in the risk assessment calculation can be defined as an asset. It has to be noted that for the definition of assets and their dependencies with the goal of creating the assets' graph, the Neo4j graph database is used due to its faster and more efficient finding of the attack paths and its integrated visual graph support.

In Figure 18 the threats definition as part of the S-RAF model is presented.

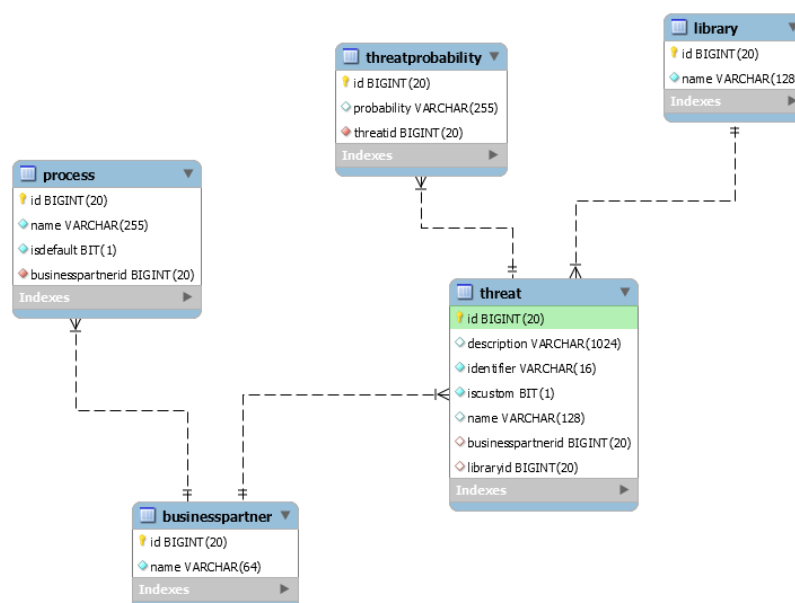


Figure 18: Threats Representation

Vulnerabilities are defined in detail through the dedicated table in the relational database, as presented in Figure 19.

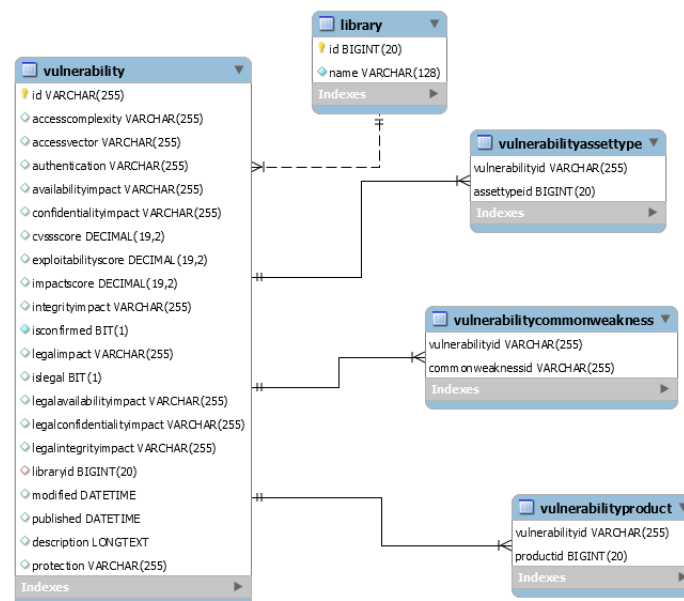


Figure 19: Vulnerabilities Representation

Finally, for the calculation of the risk assessment, S-RAF uses scenarios, that include the assets, the threats and the vulnerabilities, as depicted in Figure 20.

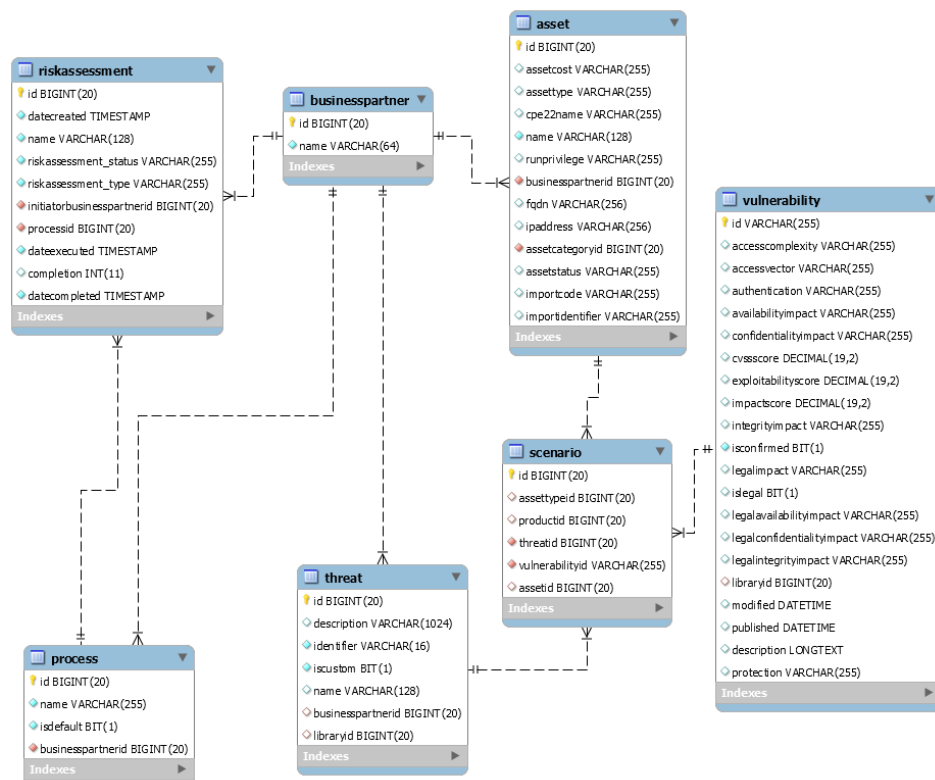


Figure 20: Representation of Scenarios for Risk Assessment in EPES for S-RAF

4.3 Compilation Process

4.3.1 Prerequisites

This project is written in **Java** and **Apache Maven** can be used for the compilation of the code, thus allowing library dependencies to be imported automatically, through the following command.

```
mvn clean compile
```

4.4 Deployment Process

S-RAF components are built as docker images and therefore can be deployed together with their dependencies (databases, Kafka queue, etc) through the usage of docker compose files. This part is described in section 5.1.

4.5 Risk Assessment Process Description

The Risk Assessment process undertakes the task of calculating the final risk scores based on the evidences collected from the various input sources of S-RAF. While D3.2 [3] provided the SDN-microSENSE Risk Assessment meta-model, including the risk calculation approaches, this section focuses on implementation aspects for exploiting the acquired inputs and deriving the final risks.

The risk assessment methodology is based on the following specific notation and considers a set of assumptions, especially for the case of cumulative risk calculation. In the following paragraphs we provide a set of key characteristics of the risk assessment methodology, which are important for grasping the rational of the risk calculation.

- The EPES ecosystem consists of interrelated assets A_1, \dots, A_n which are interconnected with each other based on their dependency. So far, in the SDN-microSENSE methodology the following interdependencies have been identified: *IsInstalledOn*, *IsLocatedIn*, *IsConnectedTo*, *IsUsedBy*, *IsProcessedBy* and *IsStoredOn*. Thus, the assets form a graph or network, namely the interdependency graph, where the assets are represented by the nodes and the interconnections among them are represented by the edges.
- Each asset A_i may have several vulnerabilities $V_{A_i,1}, \dots, V_{A_i,n_i}$. Hence, the network of assets can further be transformed in a graph, where the nodes are combinations of assets and vulnerabilities. The vulnerabilities for each asset can be found with the help of specific software. In our case, this task is undertaken by eVul. The information of eVul is integrated with the assets' information in the context of the vulnerability identification component of S-RAF.
- Each asset/vulnerability pair $V_{A_i,j}$ has two core characteristics (among others), which are important for further computation: one describing the Individual Vulnerability Level (IVL) $IVL(V_{A_i,j})$ and the other one describing the Impact Level $I(V_{A_i,j})$. These two characteristics are based on specific values coming from the Common Vulnerability Scoring System (CVSS) and they describe the severity of a vulnerability. For our further analysis, the interested reader can refer to D3.2 [3].

After defining some fundamental notions for the risk assessment approach, the following sections elaborate on the calculation of the two risk variants, namely the individual and the cumulative risks.

4.5.1 Risk Calculation

We will rely on qualitative values for representing the risk levels and the general Impact Level of a vulnerability (in terms of the three standard security criteria Confidentiality, Integrity and Availability). Similarly, we will use a semi-quantitative scale for the probability that a vulnerability is exploited by a specific attacker. To make the handling easier, we are using a five-tier scale for all three applications and will use this five-tier scale throughout the SDN-microSENSE methodology. These five categories are ranging from

- “Very Low” (VL)
- “Low” (L)
- “Moderate” (M)
- “High” (H)
- “Very High” (VH).

4.5.1.1 Individual Risk Calculation

The Individual Risk Level (IRL) represents how dangerous a threat is to the specific asset within EPES. More specifically, IRL quantifies the risk of an asset taking into consideration all the associated vulnerabilities ignoring the assets dependencies and relationships. The IRL can be calculated as a multiplication of the imposed Threat level (TL), Individual Vulnerability and Impact Levels (IVL and IIL, respectively) as follows:

$$IRL = TL \times IVL \times IIL$$

More details on the calculation methods of the factor of the abovementioned formula are given in D3.2 [3].

4.5.1.2 Cumulative Risk Calculation

The Cumulative Risk Level (CRL) refers to the risk level imposed to an asset (a target point), as a result of a vulnerability exploitation, given a threat, to an entry asset. The cumulative risk level can be derived if there is a path that connects the entry asset to the target asset. In other words, CRL quantifies the risk that is caused on a single asset using the vulnerability profiles of the adjacent assets taking into consideration all the possible attack paths that are generated towards this specific asset. The cumulative risk represents how dangerous a threat is to the specific asset and can be calculated as a multiply of threat level (TL), cumulative vulnerability level (CVL) and cumulative impact (CIL) as follows:

$$CRL = TL \times CVL \times CIL$$

More details on the calculation methods of the factors of the abovementioned formula are given in D3.2 [3].

Application of the Attacker Profile

In order to be able to materialize the cumulative methodology, it is vital to document the conditions under which an attacker can propagate in a network by exploiting vulnerabilities. In the Cumulative Risk Assessment methodology, the Cumulative Vulnerability Level (CVL) of an asset/vulnerability combination heavily relies on the attacker profile, i.e., the skills and characteristics the analyst grants the attacker in a specific risk assessment. Only by using the information coming from this specific attacker profile, we will be able to make some estimation on the probability, that a vulnerability can

be exploited. An attacker has two main properties: the capability (if the attacker is skilled or just an amateur) and his location (if he is attacking from the outside or from the inside).

The attacker's location can be used to reduce the number of potential entry points, since an outside attacker can just attack assets/vulnerability combinations with a CVSS Access Vector "Network", as already explained in D3.2 [3]. We will assume that if the Access Vector is "Adjacent" or "Local", an attacker from the outside will not be able to exploit this vulnerability.

In the context of SDN-microSENSE, due to the high criticality of the infrastructure aimed to be protected, we consider attackers that have "High" and "Very High" level of expertise. By considering these levels of expertise, the overall risk assessment framework adopts a rather risk-averse approach that, on the one hand may require a more intense engagement on behalf of risk assessors, but on the other, guarantees that risks corresponding to propagated threats become the focal point of the analysis. Although a detailed analysis on the attacker types have been conducted in D3.2 [3], Table 11 offers a summary of the characteristics of high-skilled attackers according to NIST. We consider that these attacker types have the expertise and the technical means to exploit potential vulnerabilities on SDN-microSENSE architectural assets.

Qualitative Values	Description of the Attacker's Capability	Description of the Attacker's Intent	Description of the Attacker's Targeting
Very High (VH)	The adversary has a very sophisticated level of expertise, is well-resourced, and can generate opportunities to support multiple successful, continuous, and coordinated attacks.	The adversary seeks to undermine, severely impede, or destroy a core mission or business function, program, or enterprise by exploiting a presence in the organization's information systems or infrastructure. The adversary is concerned about disclosure of tradecraft only to the extent that it would impede its ability to complete stated goals.	The adversary analyses information obtained via reconnaissance and attacks to target persistently a specific organization, enterprise, program, mission or business function, focusing on specific high-value or mission-critical information, resources, supply flows, or functions; specific employees or positions; supporting infrastructure providers/suppliers; or partnering organizations.
High (H)	The adversary has a sophisticated level of expertise, with significant resources and opportunities to support multiple successful coordinated attacks.	The adversary seeks to undermine/impede critical aspects of a core mission or business function, program, or enterprise, or place itself in a position to do so in the future, by maintaining a presence in the organization's information systems or infrastructure. The adversary is very concerned about minimizing attack detection/disclosure of tradecraft, particularly while preparing for future attacks.	The adversary analyses information obtained via reconnaissance to target persistently a specific organization, enterprise, program, mission or business function, focusing on specific high-value or mission-critical information, resources, supply flows, or functions, specific employees supporting those functions, or key positions.

Table 11: Description of the high skilled Attacker Types according to NIST

Identification of Attack Paths

Based on the attacker profile, which is chosen by the risk assessor performing a risk assessment, according to his domain knowledge and the current security status of the infrastructure, several attack paths can be computed based on the asset/vulnerability combinations and asset interrelations.

Given that high-skilled attackers can take advantage of vulnerabilities on assets, the propagation of an attack in a network depends to the vulnerability types per se. Thus, a vulnerability that enables propagation generates a new pivot for the attacker and, consequently, a new edge in an attack path. Having said that, it becomes obvious that the generation of a new edge depends on the type of the vulnerability and the type of the interdependence between the related assets. For instance, if the “installed_on” dependency associates a service and an operating system (i.e., the service is “installed_on” the operating system), the existence of a vulnerability that enables privilege escalation on the operating system generates a pivot to compromising the service. In this direction, attacks propagated through the network require the existence of a “connected_to” relation to realise new edges in the pats. Based on the above, we consider that the formation of attack paths, due to cyber security vulnerabilities, require the existence of “connected_to” and “installed_on” relations among the assets and a type of vulnerability that enables the exploitation of this relation. Our implementation traverses the assets interdependency graph and using the information on the attacker, the vulnerabilities and the relations, a set of attack paths is returned. These paths represent the potential ways through the graph from a given Entry Point to a specific Target Point. These attack paths are required to compute the CVL.

In general, the number of potential paths from the Entry Point to the Target Point grows exponentially with the size of the graph. Due to the restrictions we obtain from the attacker’s location and intention, the number of potential paths is reduced such that a computation stays feasible.

The code snippets given in the following figures reveal part of the implementation details of core functions. The discovery of attack paths and calculations, in the context of *Attack Path Analysis Service*, over the probabilities for vulnerability exploitation in the cumulative risk assessment approach are given in Figure 21: Code snippet for attack path Identification in Attack Path Analysis Service and Figure 22.

More specifically, Figure 21: Code snippet for attack path Identification in Attack Path Analysis Service presents the method used for traversing an attack path considering an attacker with very high skills, in order to calculate the probability of an attacker exploiting this path.

```

@Override
public Map<ProbabilityScale, Long> attackPathHistogram(AttackPath attackpath,
    AttackPathAnalysis attackpathanalysis) {

    Map<String, Object> parameters = new HashMap<String, Object>();
    parameters.put("riskassessmentId", attackpathanalysis.getRiskassessmentid());
    parameters.put("assetId", attackpath.getNodes().stream().filter(n -> n.getAssetid().startsWith("a"))
        .map(n -> Long.valueOf(n.getAssetid().substring(1))).collect(Collectors.toSet()));

    Map<Long, RiskassessmentRunAsset> rassetmap = PagingUtility
        .mongoAggregationPageable(mongoTemplate, parameters, new PageRequest(0, Integer.MAX_VALUE),
            new String[] { "assetId", "name", "type", "cpe22name", "risklevel", "threats" },
            new String[] {}, RiskassessmentRunAsset.class, RiskassessmentRunAsset.class)
        .getContent().stream().filter(a -> a.getThreats().size() > 0)
        .collect(Collectors.toMap(RiskassessmentRunAsset::getAssetId, Function.identity()));

    Iterator<AttackPathNode> attackPathNodeIterator = attackpath.getNodes().iterator();

    final AttackerCapability attackerCapability = (attackpathanalysis.getConfiguration().getAttackerCapability() != null)
        ? attackpathanalysis.getConfiguration().getAttackerCapability()
        : AttackerCapability.VH;

    double[] pathProbabilitiesMatrix = new double[] { 1d };

    while (attackPathNodeIterator.hasNext()) {

        AttackPathNode apn = attackPathNodeIterator.next();

        if (!apn.isAsset()) {
            continue;
        }

        Set<RiskassessmentRunAssetThreatVulnerability> vulns = rassetmap
            .get(Long.valueOf(apn.getAssetid().substring(1))).getThreats().stream()
            .flatMap(t -> t.getVulnerabilities().stream()).collect(Collectors.toSet());

        final int matrixSize = vulns.size() * pathProbabilitiesMatrix.length;
        double[] temp = new double[matrixSize];

        for (int i = 0; i < pathProbabilitiesMatrix.length; i++) {
            int j = 0;

            for (RiskassessmentRunAssetThreatVulnerability vuln : vulns) {

                double prob = ProbabilityScale.valueOf(ChainUtils
                    .calculateAttackProbability(attackerCapability, Ranking.valueOf(vuln.getLevel()).name())
                    .getMedium());
                double result = prob * pathProbabilitiesMatrix[i];
                temp[vulns.size() * i + j++] = result;

            }

            pathProbabilitiesMatrix = temp;

        }

        Map<ProbabilityScale, Long> result = ProbabilityScale.defaultZeroMap();
        result.putAll(Arrays.stream(pathProbabilitiesMatrix).mapToObj(d -> ProbabilityScale.getRank(d))
            .collect(Collectors.groupingBy(Function.identity(), Collectors.counting())));

    }

    return result;
}

```

Figure 21: Code snippet for attack path Identification in Attack Path Analysis Service

Figure 22 highlights the method used for analysing the attack propagation in a cumulative manner. Given an entry point and target point assets, this method traverses all the possible attack paths formed between the two points, in order to calculate the accumulated probability used in risk calculations.

```
@Override
public Map<ProbabilityScale, BigDecimal> accumulatedPathHistogram(long businesspartnerid, long riskassessmentid, long entrypoint, long targetpoint) {

    Map<String, Object> parameters = new HashMap<String, Object>();
    parameters.put("businesspartnerid", businesspartnerid);

    parameters.put("riskassessmentid", riskassessmentid);
    parameters.put("configuration.entrypointid", entrypoint);
    parameters.put("configuration.targetpointid", targetpoint);

    List<AttackPathAnalysis> analyses = PagingUtility
        .mongoAggregationPageable(mongoTemplate, parameters, new PageRequest(0, Integer.MAX_VALUE),
            new String[] { "id", "riskassessmentid", "businesspartnerid", "name", "configuration", "attackpaths" },
            new String[] {}, AttackPathAnalysis.class, AttackPathAnalysis.class)
        .getContent();

    Map<ProbabilityScale, Long> accumulatedHistogram = ProbabilityScale.defaultZeroMap();

    for (AttackPathAnalysis analysis : analyses) {
        for (AttackPath attackpath : analysis.getAttackpaths()) {
            addToProbabilityHistogram(accumulatedHistogram, this.attackPathHistogram(attackpath, analysis));
        }
    }

    return amortizeHistogram(accumulatedHistogram);
}
```

Figure 22: Code snippet for analysing attach paths in cumulative manner by the Attack Path Analysis Service

Figure 23 depicts the java interface used to instruct the business logic and call methods for the *Attack Path Analysis Service* implementation.

```
@Service
public interface IAttackPathAnalysisService<A, AP, PAGE> {

    void save(A attackPathAnalysis);
    Optional<A> findOne(String id);
    Iterable<A> findAll(PAGE page);
    Iterable<A> findByBusinesspartneridAndRiskassessmentid(long businesspartnerid, long riskassessmentid, PAGE page);
    void delete(String id);
    Map<? extends Enum<?>, Long> attackPathHistogram(AP attackpath, A attackpathanalysis);
    List<A> findAttackPathAnalyses(long businesspartnerid, long riskassessmentid, long... riskassessmentidsToCompare);
    Map<? extends Enum<?>, BigDecimal> accumulatedPathHistogram(long businesspartnerid, long riskassessmentid, long entrypoint, long targetpoint);
}
```

Figure 23: Code snippet of the business logic implemented by Attack Path Analysis Service

Figure 24 depicts the function responsible for extracting the attack paths from the graph of assets and responding to the SRAF dashboard with a visualisation object for representing the attack paths between entry point assets and selected target points. The traversal of the graph of assets is done using the Depth-First Search (DFS) method.

```

@RequestMapping(value = "/(id)/discoverattackpaths", method = RequestMethod.POST)
public GenericSRAFRestResponse discoverattackpaths(@RequestBody AttackPathRequestDTO requestDTO) {

    final VisualizationWrapper wrapper = new VisualizationWrapper();

    try {

        int idx = 0;

        for (Long entrypoint : requestDTO.getEntrypoints()) {
            for (Long targetpoint : requestDTO.getTargetpoints()) {

                if (requestDTO.getDisjointness()) {

                    // Retrieve graph nodes for computation.
                    Set<GraphNode> nodes = constructGraphNode(requestDTO.getRiskassessmentid(), AttackerCapability.valueOf(requestDTO.getAttackercapabilityrank()));
                    Set<GraphPath> paths = buildDisjointPaths(requestDTO, nodes, entrypoint, targetpoint);

                    Graph graph = new Graph(constructGraphNode(requestDTO.getRiskassessmentid(),
                        AttackerCapability.valueOf(requestDTO.getAttackercapabilityrank())), "a" + entrypoint);
                    graph.printStats();

                    graph.dfsPaths("a" + entrypoint, requestDTO.getMaxlength());
                    Graph.listPaths(wrapper, paths, "a" + entrypoint, idx++);
                    //graph.listPaths(wrapper, idx++);

                } else {

                    Graph graph = new Graph(constructGraphNode(requestDTO.getRiskassessmentid(),
                        AttackerCapability.valueOf(requestDTO.getAttackercapabilityrank())), "a" + entrypoint);
                    graph.printStats();

                    if (requestDTO.getPropagation()) {
                        graph.dfs("a" + entrypoint, requestDTO.getMaxlength());
                    } else {
                        graph.dfs("a" + entrypoint, "a" + targetpoint, requestDTO.getMaxlength());
                    }

                    graph.list(wrapper, idx++);

                }

            }
        }

    } catch (Exception ex) {
        ex.printStackTrace();
        Logger.getLogger(RiskAssessmentRestController.class.getName()).severe(ex.getMessage());
        return new GenericSRAFRestResponse(BasicResponseCode.INVALID, Message.R_ERROR, Optional.empty());
    }

    return new GenericSRAFRestResponse(BasicResponseCode.SUCCESS, Message.R_AMENDED, wrapper);
}

```

Figure 24: Code snippet of extracting attack paths from a graph of assets

Figure 25 depicts the method responsible for analysing the attack paths and triggering the *Attack Path Analysis Service* for calculating the path risks.

```

@RequestMapping(value = "/(id)/analyzeattackpaths", method = RequestMethod.POST)
public GenericSRAFRestResponse analyzeAttackPaths(@RequestBody ExtendedAttackPathRequestDTO requestDTO) {
    try {
        @SuppressWarnings("serial")
        Map<String, Object> parameters = new HashMap<String, Object>() {
            {
                put("riskassessmentid", requestDTO.getRiskassessmentid());
                put("assetId", Stream.concat(requestDTO.getEntrypoints().stream(), requestDTO.getTargetpoints().stream()).collect(Collectors.toSet()));
            }
        };

        Map<Long, RiskAssessmentRunAsset> rrasemap = PagingUtility
            .mongoAggregationPageable(mongoTemplate, parameters, new PageRequest(0, Integer.MAX_VALUE),
                new String[] { "assetId", "name" }, new String[] {},
                RiskAssessmentRunAsset.class, RiskAssessmentRunAsset.class)
            .getContent().stream()
            .collect(Collectors.toMap(RiskAssessmentRunAsset::getAssetId, Function.identity()));

        for (Long entrypoint : requestDTO.getEntrypoints()) {
            for (Long targetpoint : requestDTO.getTargetpoints()) {
                Set<GraphNode> nodes = constructGraphNodes(requestDTO.getRiskassessmentid(), AttackerCapability.valueOf(requestDTO.getAttackercapabilityrank()));
                Set<GraphPath> paths = buildDisjointPaths(requestDTO, nodes, entrypoint, targetpoint);

                AttackPathAnalysis analysis = new AttackPathAnalysis();
                analysis.setRiskassessmentid(requestDTO.getRiskassessmentid());
                analysis.setBusinesspartnerid(authUtil.getAuthenticatedUser().getBusinesspartner().getId());
                analysis.setName(requestDTO.getName());
                analysis.setCreateddate(LocalDateTime.now());
                analysis.getConfiguration().setAttackerCapability(AttackerCapability.valueOf(requestDTO.getAttackercapabilityrank()));
                analysis.getConfiguration().setAttackerLocation(requestDTO.getAttackerlocation());
                analysis.getConfiguration().setEntrypointId(entrypoint);
                analysis.getConfiguration().setEntrypointName(rrasemap.get(entrypoint).getName());
                analysis.getConfiguration().setTargetpointId(targetpoint);
                analysis.getConfiguration().setTargetpointName(rrasemap.get(targetpoint).getName());

                int idx = 1;
                for (GraphPath graphPath : paths) {
                    AttackPath attackPath = new AttackPath(idx++, graphPath.getLength());

                    for (int i=0; i<graphPath.getLength(); i++) {
                        AttackPathNode apNode = new AttackPathNode(i+1, graphPath.get(i).getId(), graphPath.get(i).getName(), graphPath.get(i).getRisk());
                        attackPath.getNodes().add(apNode);
                    }
                    analysis.getAttackpaths().add(attackPath);
                }
                attackPathAnalysisService.save(analysis);
            }
        }
    } catch (Exception ex) {
        ex.printStackTrace();
        Logger.getLogger(RiskAssessmentRestController.class.getName()).severe(ex.getMessage());
        return new GenericSRAFRestResponse(BasicResponseCode.INVALID, Message.R_ERROR, Optional.empty());
    }
    return new GenericSRAFRestResponse(BasicResponseCode.SUCCESS, Message.R_AMENDED, Optional.empty());
}

```

Figure 25:Code snippet of attack path risk analysis

4.5.1.3 Risk Profiles

A risk profile is an evaluation of an individual's or organisation's willingness and ability to take risks. It can also refer to the threats to which an organization is exposed and to the perception and the attitude of a security officer against risks.

In order to incorporate this aspect in the developed assessment methodology, we capitalise on the risk appetite notion. According to ISO 31000 risk management standard [16], the risk appetite is defined as the "Amount and type of risk that an organization is prepared to pursue, retain or take". This concept supports an organization's decisions for risk management.

Willingness to take on risk entails to the definition of the risk aversion. If an organization or the risk officer, expresses a strong desire to keep the risk at the minimum level given his/her domain knowledge, the security state of the infrastructure and the security awareness of the personnel, this person would have a low willingness to take on risk and is risk-averse.

Given the above, in the context of SDN-microSENSE the risk profile of each organization per use case can be reflected in the definition of the Risk Appetite level. The risk assessment model will be in position to consider this factor, which may vary depending on the threats, the security state of systems, the security awareness of the personnel and the attitude of the risk officer of the use cases.

The Risk appetite level will be used to regulate the final risk assessment formula by applying a percentage factor in the final calculation. The detailed magnitude of the risk appetite level will be decided per use case based on the prioritization of threats, the personnel assessment and other possible factors.

For the actual usage of the S-RAF tool for the risk assessment we provide more details in section 5.2.

4.5.1.4 Integration with Personnel Assessment

As explained also in D3.1 [2], company employees appear as one of the potential threat agents that can cause a cyber-attack. In most cases unintentionally, because of lack of training or awareness, but also in an intentional way (disgruntled employees), employees can put the company's assets at risk. S-RAF assets are not only referring to the physical infrastructures and equipment, but also covers information that the company manages (inventory of components, state of the network, information of its customers, etc.), communication networks, applications of control, databases, personal work devices (e.g. laptops, tablets, mobiles) and physical infrastructure. As presented in Figure 29, S-RAF allows adding personnel specific evaluation (e.g. Administrator), at role or at individual level, as part of the asset-based risk assessment.

The level of training and awareness of employees in adopting good practices is considered an important input in the risk assessment process in SDN-microSENSE. Therefore, companies must include an evaluation process of the readiness level of their employees as part of their security procedures. This can be reflected through S-RAF to the overall risk appetite of the overall risk assessment, as depicted in Figure 33. This assigned risk appetite will be used to regulate the final risk assessment formula by applying a percentage factor in the final risk.

Due to the distributed nature of the decentralized energy system, the developed methodology shall take into account the collaborative aspects needed by involving all stakeholders of the energy components. This includes mainly personnel at different places or task roles, but also include external stakeholders such as energy operators, consumers and energy retailers. Although there is no possibility to have a full evaluation as for the personnel, nevertheless stakeholders can be represented with an appropriate triplet of asset-vulnerability-criticality level.

4.6 Requirements and Unit Testing Coverage

Finally, in order to ensure the functional validity of the solution we examined the relevant requirements, as defined in D2.2 [1]. The following requirements presented in Table 12 have been covered.

ID	Description	Coverage by S-RAF
FR-UR-01	The system shall be able to perform a cybersecurity risk assessment process per six months.	Risk assessment can be able performed upon request whenever needed.
NFR-DPT-23	The system shall be able to handle a data breach, through an effective data breach management policy covering: <ul style="list-style-type: none"> • Containment and recovery • Assessment of ongoing risk • Notification of breach • Evaluation and response 	Risk assessment can be calculated both as individual risk and as cumulative risk
NFR-DPT-28	In case of testing, logical separation of the test site from the operational site shall be ensured, so that the risk to the actual infrastructure shall be minimised or eliminated in case of a breach.	Risk assessment can be performed in testing or operational sites.

NFR-SEC-01	For browsing web pages, the system shall be able to require HTTPS for all sensitive pages or where transmission of personal data takes place. Non-HTTPS requests to these pages should be redirected to the HTTPS page. Any included content such as images, JavaScript or CSS should also be provided over HTTPS in order to avoid 'mixed content' warnings in users' browsers.	HTTPS will be used on S-RAF deployments for the demonstrators
NFR-SEC-02	The system shall be able to employ HTTPS connections between all backend components and external systems.	HTTPS will be used on S-RAF deployments for the demonstrators
FR-GR-01	The system shall maintain an inventory of all the infrastructure elements with their exact location and their status (e.g., which devices are active or not, whether the cause is known (e.g. due to maintenance work) or unexpected (e.g., due cyber-attack or failure)). This applies to all types of devices, i.e. power, IT and OT network devices. All infrastructure elements shall be identified by a unique ID, which would be shared by the different databases in the system (if any).	SDN-microSENSE Risk Assessment Framework (S-RAF) uses the inventory of the assets through the Assets Identification Component.

Table 12: SDN-microSENSE Requirements relevant to the Risk Assessment

In addition, deliverable D2.4 [17] defined the validation procedures for the overall platform; based on this strategy owners of SDN-microSENSE components should include unit tests to test specific core functions of the components. The unit tests defined, implemented and tested for S-RAF are provided in Annex II –Unit Tests, while test related to the actual integration of SDN-microSENSE platform and the proper interconnections with other components, will be implemented in the Early Prototype of SDN-microSENSE platform, and reported in the deliverable D7.4.

5 SDN-microSENSE Risk Assessment Framework Adoption and Usage

In this section we explain how S-RAF can be installed and used as a standalone service. It has to be mentioned that the integration process with the components of SDN-microSENSE is still to be performed in the scope of WP7 of the project; however, the documentation provided below can be valuable for the integration process and also for assisting the adoption of S-RAF and SDN-microSENSE by the project's demonstrators.

5.1 Installation and Deployment

5.1.1 Preparing the environment

This section provides information regarding the installation of the S-RAF prototype implementation.

5.1.1.1 Prerequisites

For the installation a docker compose file is provided, and in an environment where Docker has already been installed, it can be used to easily.

- Deploy MySQL database
- Deploy MongoDB database
- Deploy OpenVAS
- Deploy eVul
- Deploy Kafka

All these services and components are required for the S-RAF tool to run properly. They may also be deployed in any other way, however the docker-compose.yml file must be consulted for the advertised names of the services above.

5.1.2 Deployment

For the first ever execution through console one must navigate to the location of the .yml file and type the command:

```
docker-compose up -d
```

This first execution shall delay a little because it pulls pre-built images from online repositories (e.g. Docker Hub). Although the specific docker compose will be in reality configured based on each demonstrator needs, the docker-compose.yml file currently used is provided in the Annex I – Docker Compose for S-RAF installation.

5.1.3 Verification

To verify that all is well one may type

```
docker ps
```

Apart from the actual service, three other running containers must be visible: a) the MySQL (database) b) the MongoDB (database) and c) the OpenVAS.

To open their logs one may type:

```
docker logs -f <container name>
```

To exit the log type **Ctrl+C**.

5.1.4 Undeployment

To stop and remove all containers one may type the command:

```
docker-compose down
```

5.1.5 Deployment View

This Deployment View is intended to describe the specific components that are used in practice to implement the different functions described in the previous structural views presented in section 3.

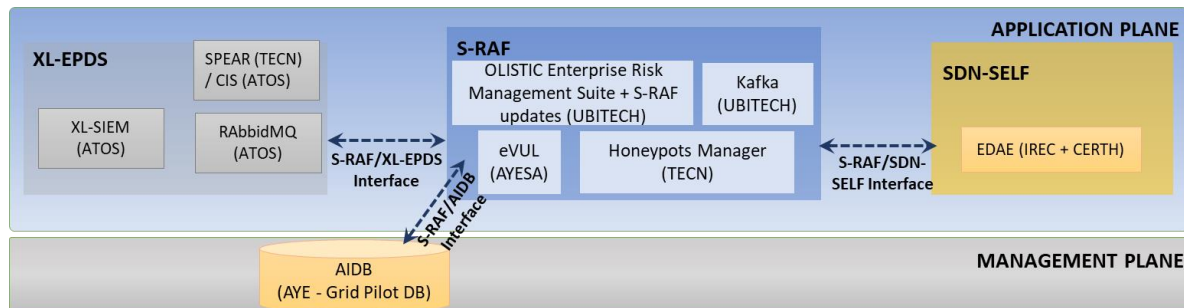


Figure 26: S-RAF component's deployment view

Based on the interfaces defined and the deployment view, we highlight the fact that S-RAF will need to have direct communication with the RabbitMQ of ATOS to retrieve alerts and the ADAE component of S-RAF, through the deployed Kafka of S-RAF. All these components are part of the application plane and their communication is considered easily supported. For the communication with the AIDB that is part of the management plane we will consider the networking options more suitable per pilot installation.

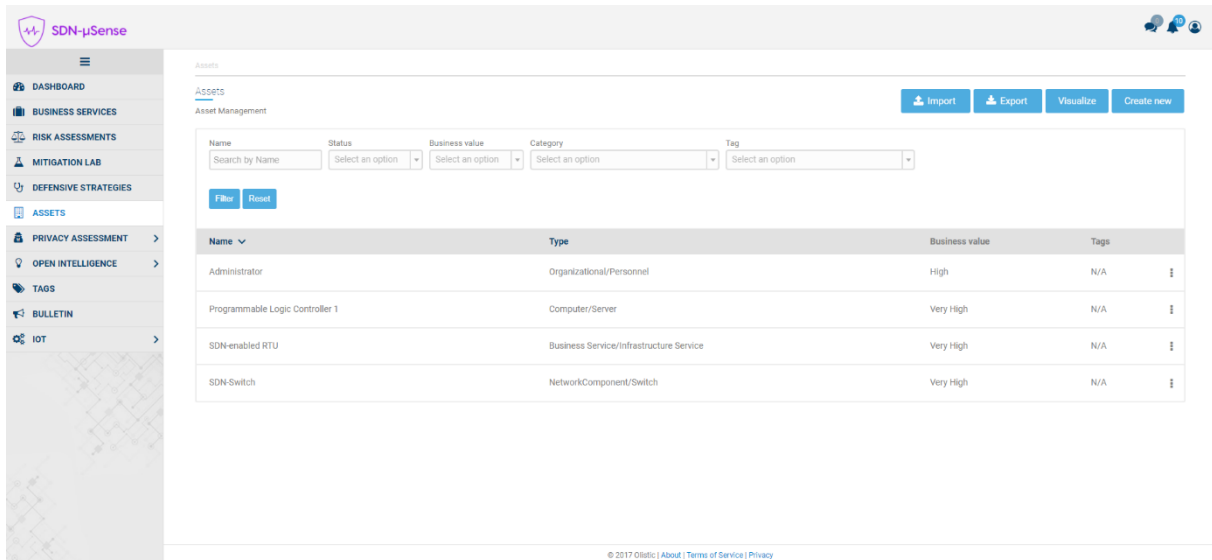
5.2 S-RAF Usage

This section provides an overview of the usage workflow of the S-RAF component. The aim is to steer the user of S-RAF through its graphical user interface and to assist to the exploration and exploitation of the competitive advantages of the tool.

5.2.1 Usage workflows

5.2.1.1 Asset management

Asset management is performed from the Asset Identification Component. Assets are populated from the Asset Identification Database (AIDB); however, S-RAF allows for asset management (e.g. add, edit and delete) from the UI. For instance, all the intangible Human Assets (see AST-33 from Table 13) are added from the UI by the security administrator. In addition, the security administrator is responsible to assign a Business Value to all the assets (both tangible and intangible) for the calculation of the risk. A snapshot of the populated (tangible) and manually added (intangible) assets are presented in the Figure 27 below, while in Figure 28 the UI to add a new asset is displayed. In addition, Figure 28 depicts all the S-RAF supported interdependencies (Relationship), the *IsInstalledOn*, *IsLocatedIn*, *IsConnectedTo*, *IsUsedBy*, *IsProcessedBy* and *IsStoredOn*. In D3.2 [3] three interdependency classes was identified (the *IsInstalledOn*, *IsConnectedTo* and *IsUsedBy*), but based on the SDN-microSENSE demonstrator analysis we further extend our classes for better representation of the interdependencies. Figure 29 displays the visualisation of the asset interdependencies (the asset interdependency graph), where each dependency is represented by the edges and each node by the assets.

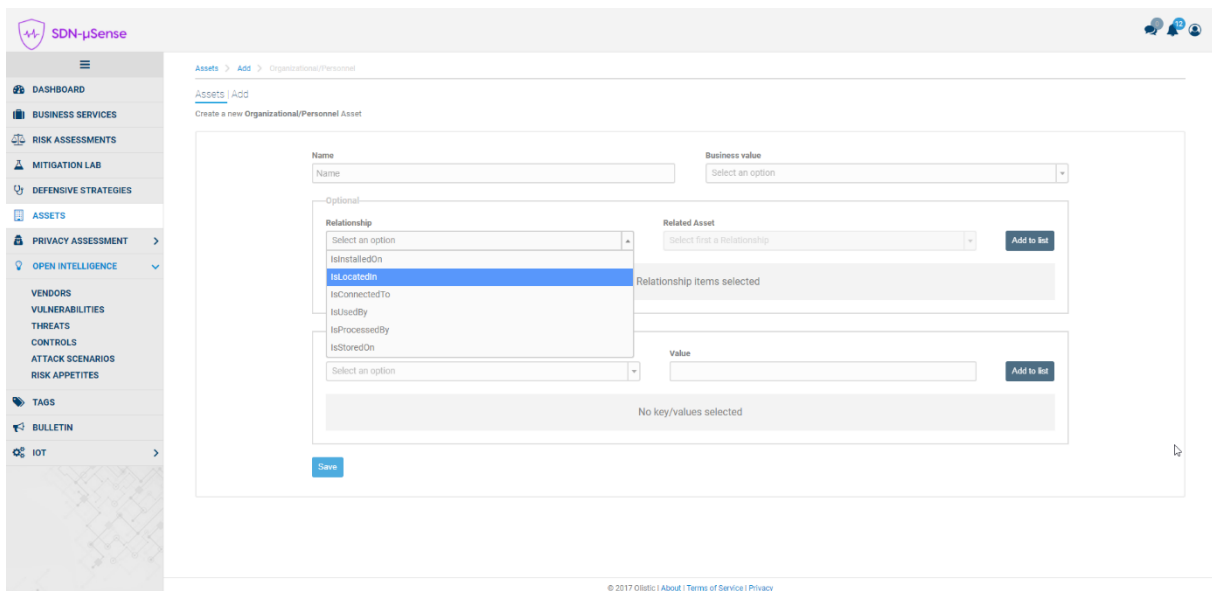


The screenshot shows the 'Assets' management interface. On the left is a sidebar with navigation options: DASHBOARD, BUSINESS SERVICES, RISK ASSESSMENTS, MITIGATION LAB, DEFENSIVE STRATEGIES, ASSETS (selected), PRIVACY ASSESSMENT, OPEN INTELLIGENCE, TAGS, BULLETIN, and IOT. The main content area is titled 'Assets' and 'Asset Management'. It includes filters for Name, Status, Business value, Category, and Tag. Below the filters is a table listing assets:

Name	Type	Business value	Tags
Administrator	Organizational/Personnel	High	N/A
Programmable Logic Controller 1	Computer/Server	Very High	N/A
SDN-enabled RTU	Business Service/Infrastructure Service	Very High	N/A
SDN-Switch	NetworkComponent/Switch	Very High	N/A

At the bottom of the page, there is a footer: © 2017 Orlatic | About | Terms of Service | Privacy.

Figure 27: Asset management (List)



The screenshot shows the 'Assets' management interface in 'Add' mode. The sidebar is the same as in Figure 27. The main content area is titled 'Assets > Add > Organizational/Personnel' and 'Assets | Add'. It prompts the user to 'Create a new Organizational/Personnel Asset'. The form includes fields for Name, Business value, and an 'Optional' section for relationships. The 'Optional' section has a 'Relationship' dropdown with options: 'Select an option', 'IsInstalledOn', 'IsLocatedOn' (selected), 'IsConnectedTo', 'IsUsedBy', 'IsProcessedBy', and 'IsStoredOn'. To the right of the 'Relationship' dropdown is a 'Related Asset' dropdown with the text 'Select first a relationship' and an 'Add to list' button. Below the 'Relationship' dropdown is a 'Value' field with an 'Add to list' button. At the bottom of the form is a 'Save' button. The footer is the same as in Figure 27.

Figure 28: Asset management (Add)

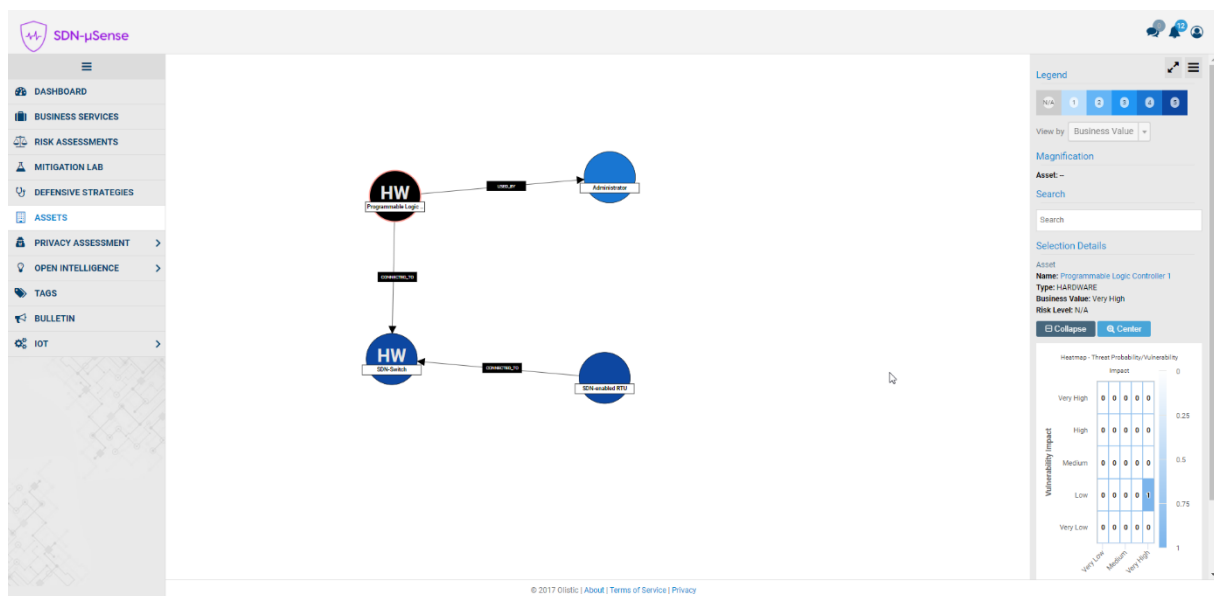
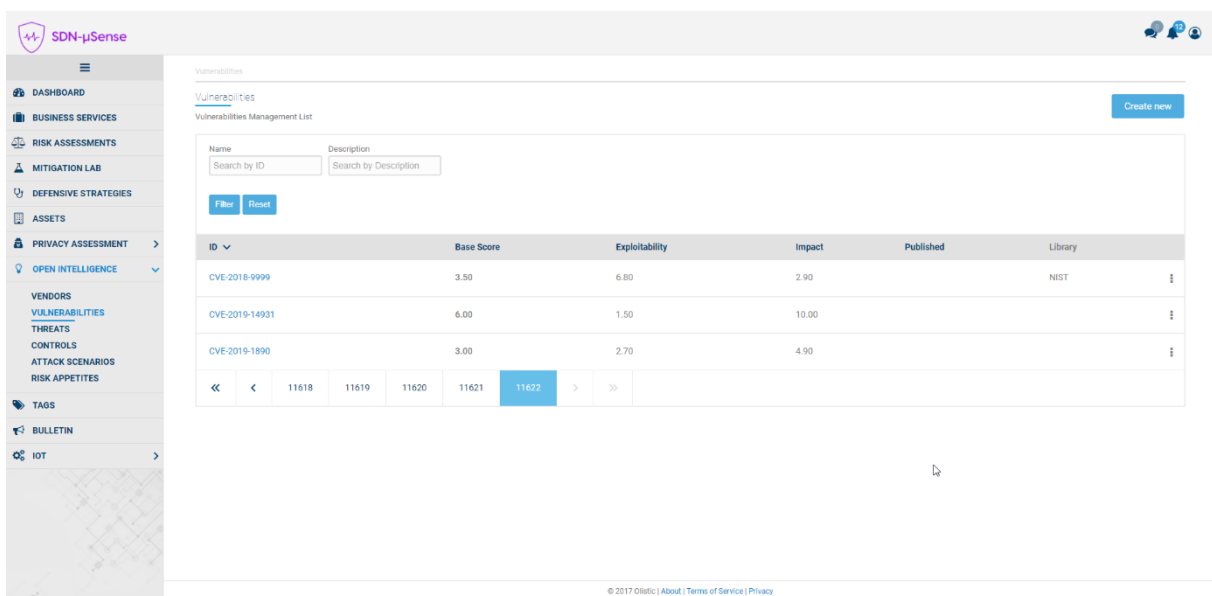


Figure 29: Asset management (Visualisation)

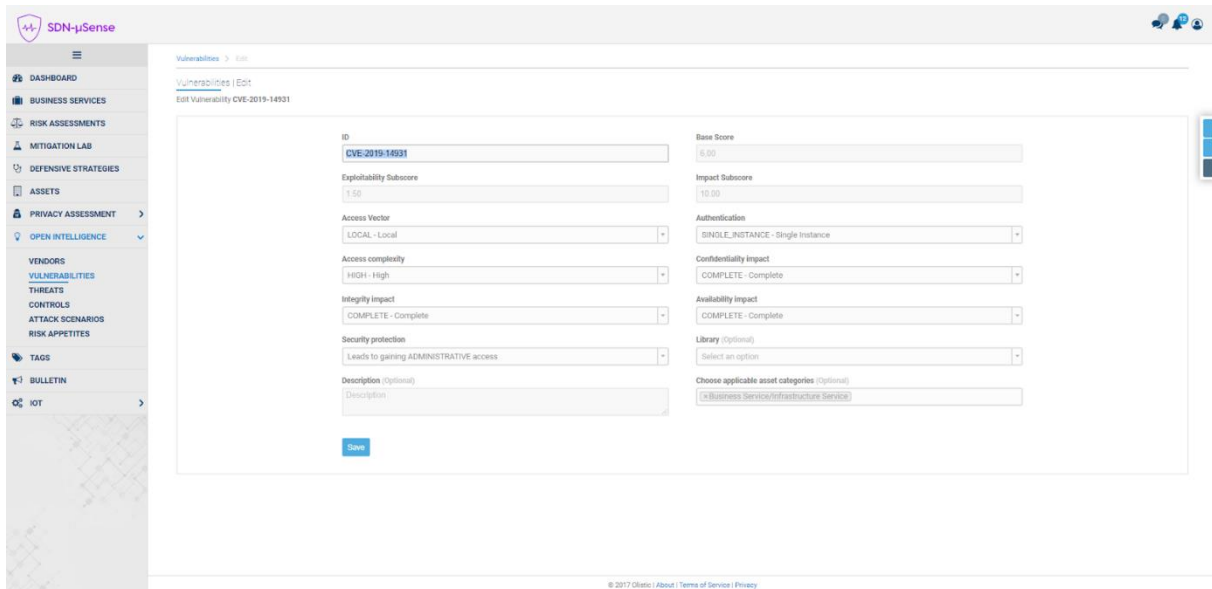
5.2.1.2 Vulnerability and threat management

Vulnerabilities are populated from the eVul tool and Threats from the XL-SIEM. Figure 30 presents a snapshot of the populated vulnerabilities from the eVul, while Figure 31 depicts the vulnerability representation in the S-RAF (e.g. the CVE-2019-14931) including the corresponding CVSS metrics. In D3.2 [3] we already referred to the CVSS metrics such as the *Exploitability* and *Impact*. Moreover, Figure 32 presents a snapshot of the populated threats from the XL-SIEM, and Figure 33 depicts the associated Risk appetite ranging from “Very Low” to “Very High” in each identified Threat to regulate the final risk assessment formula by applying a percentage factor in the final calculation. The security administrator is responsible to assign a specific Risk appetite per threat according to his background and experience. It is also possible to manage both vulnerabilities and threats from the UI.



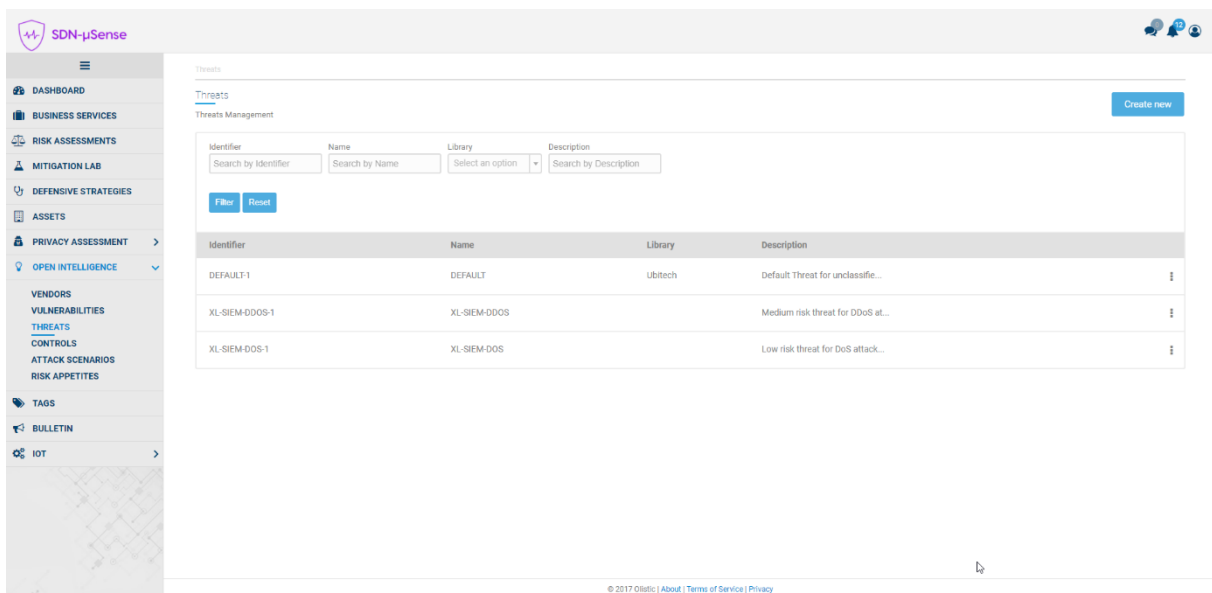
ID	Base Score	Exploitability	Impact	Published	Library
CVE-2018-9999	3.50	6.80	2.90		NIST
CVE-2019-14931	6.00	1.50	10.00		
CVE-2019-1890	3.00	2.70	4.90		

Figure 30: Vulnerability management



The screenshot shows the 'Edit Vulnerability' form for CVE-2019-14931. The form is divided into two columns. The left column contains fields for ID (CVE-2019-14931), Exploitability Subscore (1.50), Access Vector (LOCAL - Local), Access complexity (HIGH - High), Integrity impact (COMPLETE - Complete), Security protection (Leads to gaining ADMINISTRATIVE access), and Description (optional). The right column contains fields for Base Score (6.00), Impact Subscore (10.00), Authentication (SINGLE_INSTANCE - Single instance), Confidentiality impact (COMPLETE - Complete), Availability impact (COMPLETE - Complete), Library (optional), and Choose applicable asset categories (optional). A 'Save' button is at the bottom left.

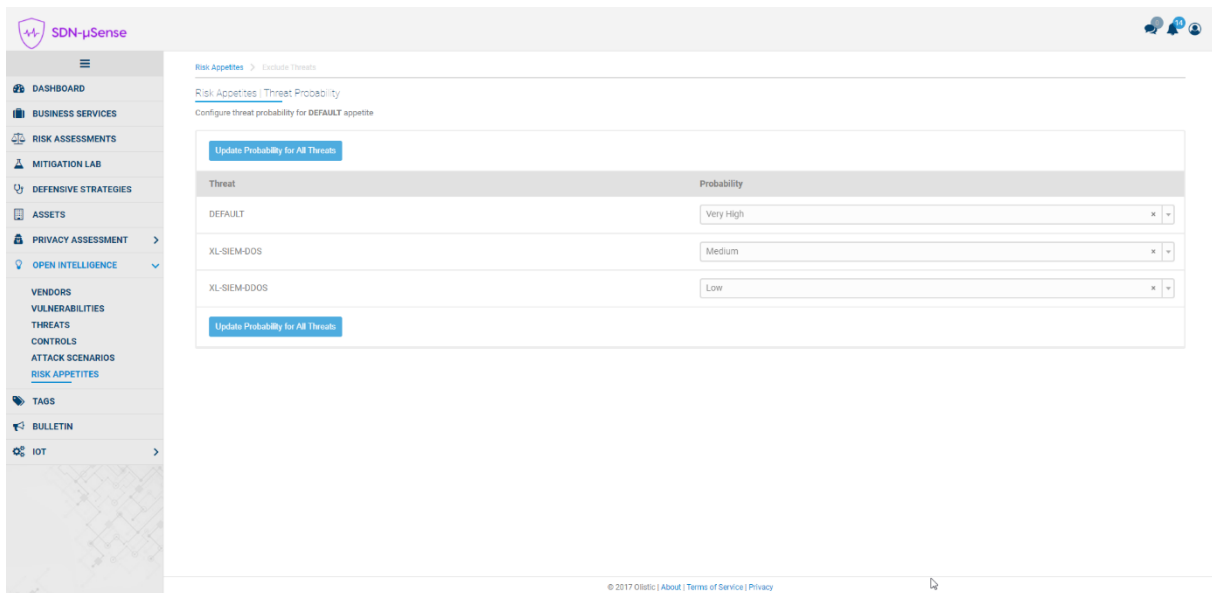
Figure 31: Vulnerability management (CVE-2019-14931)



The screenshot shows the 'Threats Management' interface. It includes a search bar with filters for Identifier, Name, Library, and Description. Below the search bar is a table with the following data:

Identifier	Name	Library	Description
DEFAULT-1	DEFAULT	Ubitech	Default Threat for unclassifi...
XL-SIEM-DDOS-1	XL-SIEM-DDOS		Medium risk threat for DDoS at...
XL-SIEM-DOS-1	XL-SIEM-DOS		Low risk threat for DoS attack...

Figure 32: Threat management



Risk Appetite > Evaluate Threats

Risk Appetites / Threat Probability

Configure threat probability for DEFAULT appetite

[Update Probability for All Threats](#)

Threat	Probability
DEFAULT	Very High
XL-SIEM-DOS	Medium
XL-SIEM-DDOS	Low

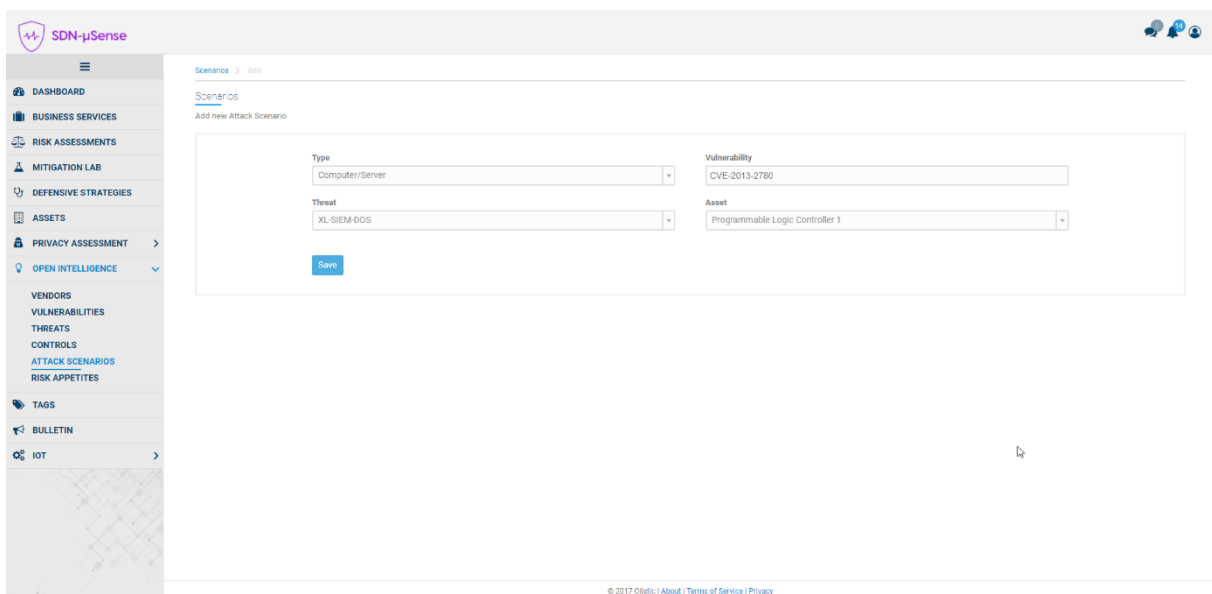
[Update Probability for All Threats](#)

© 2017 Orlitic | [About](#) | [Terms of Service](#) | [Privacy](#)

Figure 33: Risk Appetite

5.2.1.3 Risk assessment and evaluation

At this stage all the necessary information is already added/populated in the S-RAF and the security administrator is responsible to create the Attack Scenarios (see Figure 34). The Attack Scenario represents the combination triplet among the Vulnerability, the Threat and the Asset and is performed in order to initiate the risk evaluation procedure. S-RAF provides two Risk variants, namely the Individual and Cumulative Risk Levels (IRL and CRL respectively). A general dashboard that visualizes all the information, including both the IRL and CRL, is presented in Figure 35, where the security administrator has an overview of the risk in the EPES.



Scenarios > Add

Scenarios

Add new Attack Scenario

Type: Computer/Server

Vulnerability: CVE-2013-2780

Threat: XL-SIEM-DOS

Asset: Programmable Logic Controller 1

[Save](#)

© 2017 Orlitic | [About](#) | [Terms of Service](#) | [Privacy](#)

Figure 34: Attack Scenario

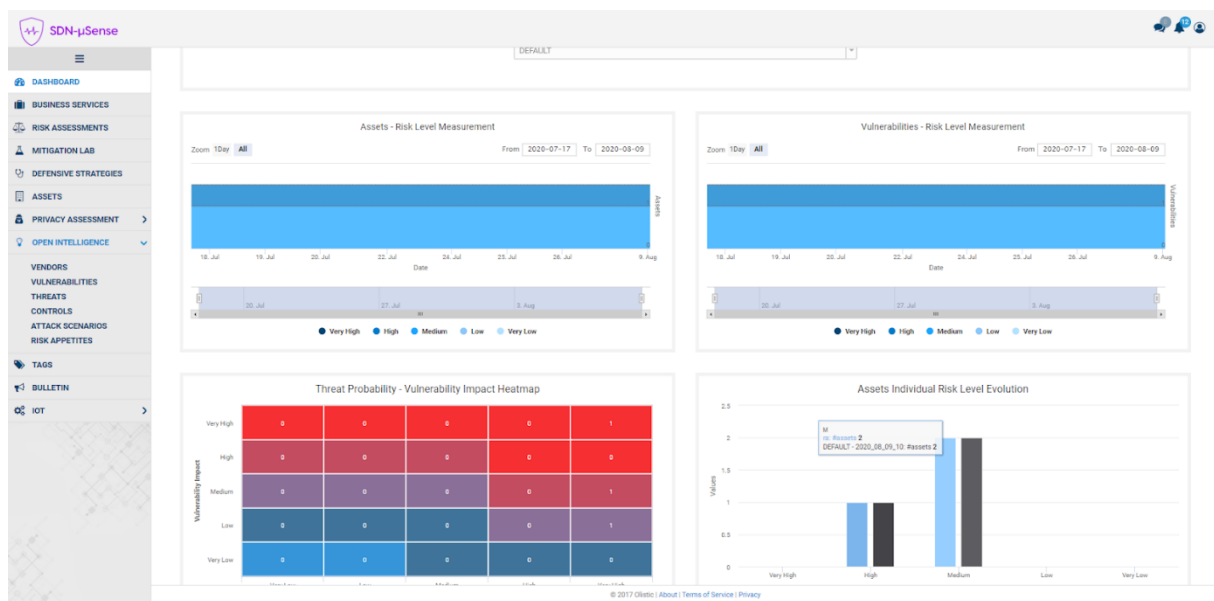


Figure 35: Risk Assessment General Dashboard

5.2.1.3.1 Individual Risk

The Individual Risk Level (IRL) quantifies the risk of an asset, taking into consideration all the associated vulnerabilities and ignoring the asset's dependencies and relationships. Figure 36 presents a detailed risk assessment report for the IRL. This report includes all the assets in the graph, the associated vulnerabilities and the calculated IRL per asset. As can be easily identified from both Figure 36 and Figure 37, in the first variant of the risk assessment (IRL), three tangible assets are identified in the attack graph where the two of them are of Medium IRL and one of High IRL.

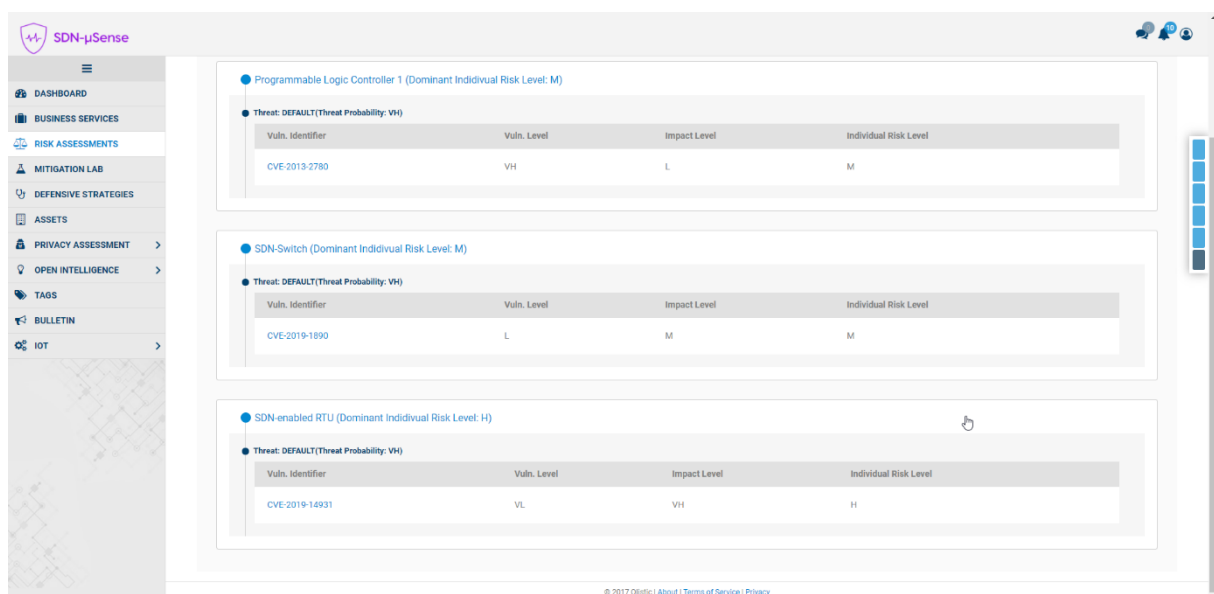


Figure 36: Individual Risk Level Report

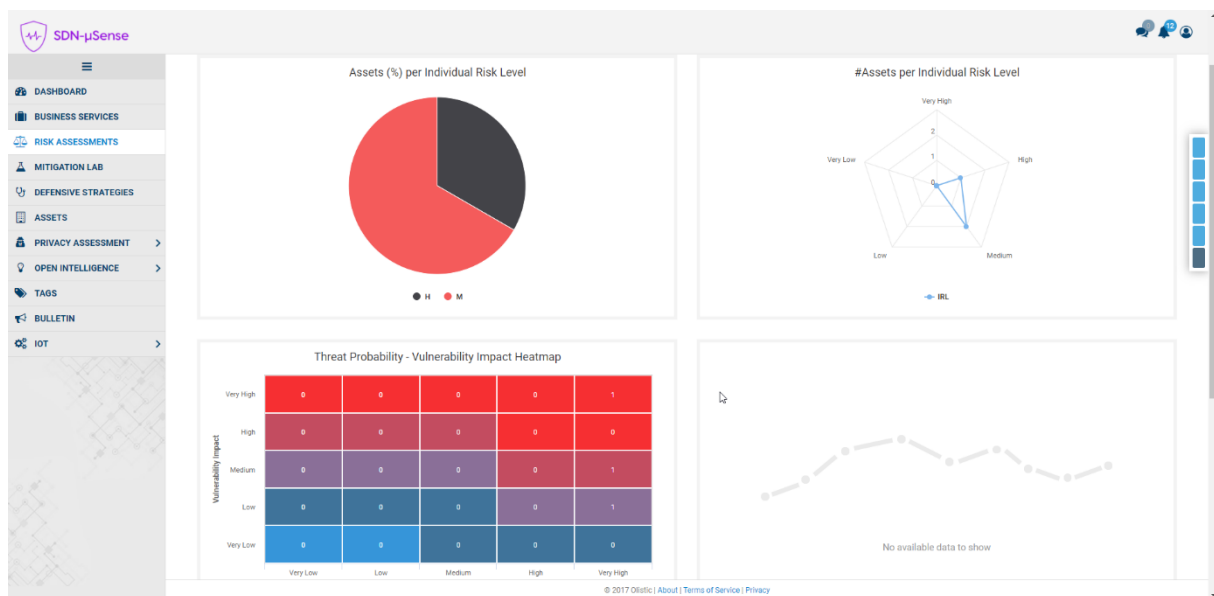


Figure 37: Executive IRL Summary

5.2.1.3.2 Cumulative Risk

The Cumulative Risk Level (CRL) refers to the risk level imposed to an asset (a target point), as a result of a vulnerability exploitation, given a threat, to an entry asset. The cumulative risk level can be derived if there is a path that connects the entry asset to the target asset. Figure 38 displays the attack path representation, where the SDN-Switch is the target point and the SDN-enabled RTU is the entry point. By performing the second variant of risk assessment the resulting report (see Figure 39) includes only the assets that are included in the attack path, while the cumulative risk is calculated in the target point (the SDN-Switch). CRL compared with the IRL in SDN-Switch (see Figure 36 and Figure 39) changed from Medium to High, since it also considers the risk of the SDN-enabled RTU asset. As can be easily identified from both Figure 39 and Figure 40, in the second variant of the risk assessment (CRL), two tangible assets are identified in the attack path of High risk (one IRL - entry point and one CRL - target point).

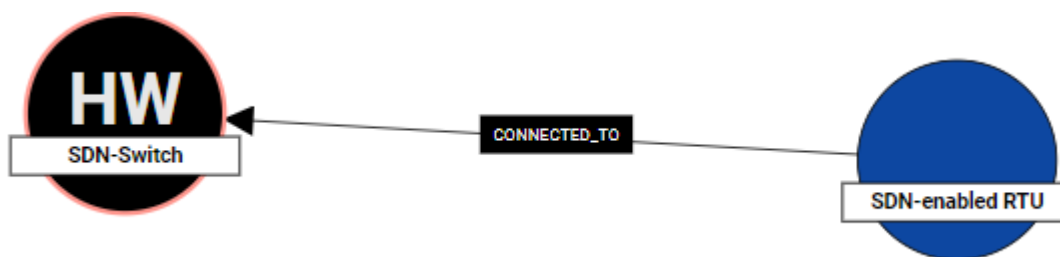


Figure 38: Cumulative Attack Path

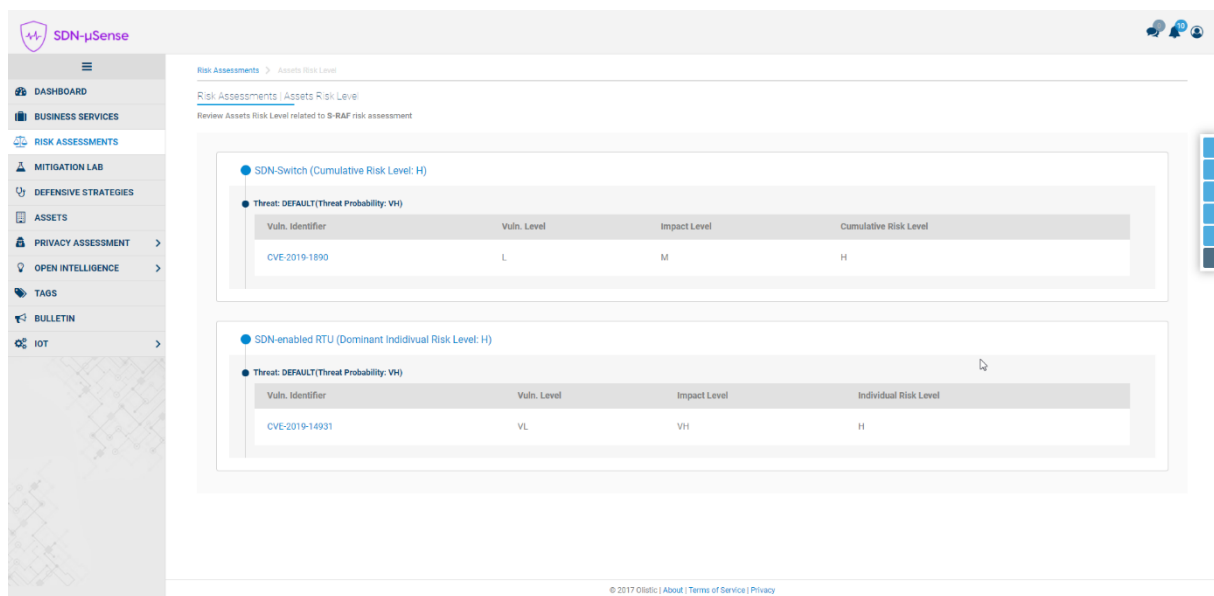


Figure 39: Cumulative Risk Level Report

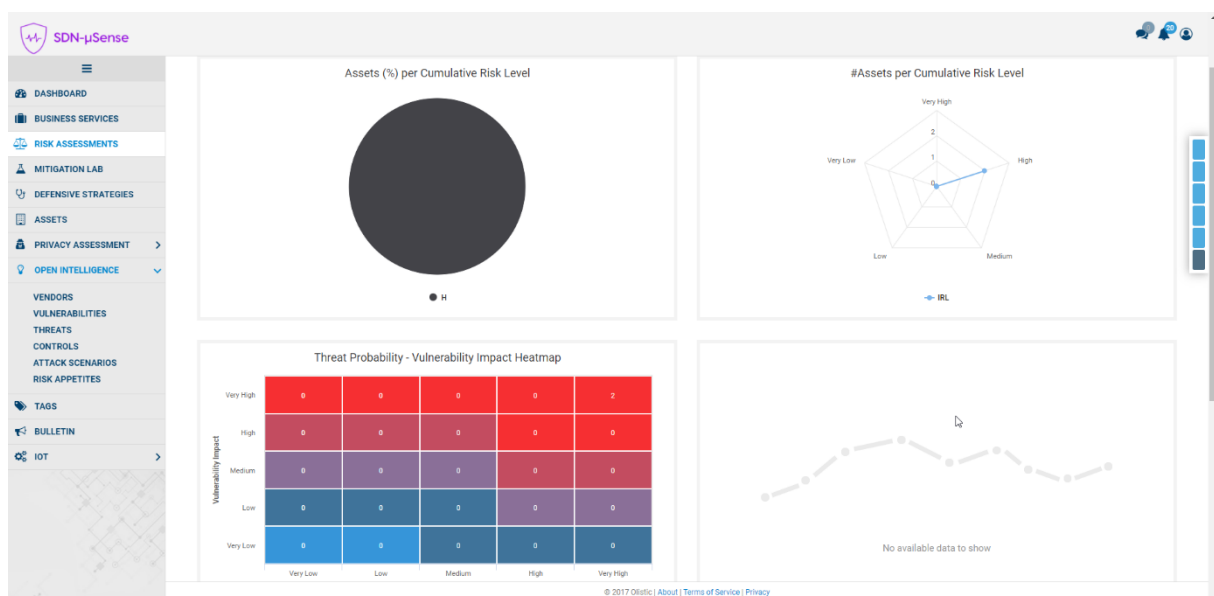
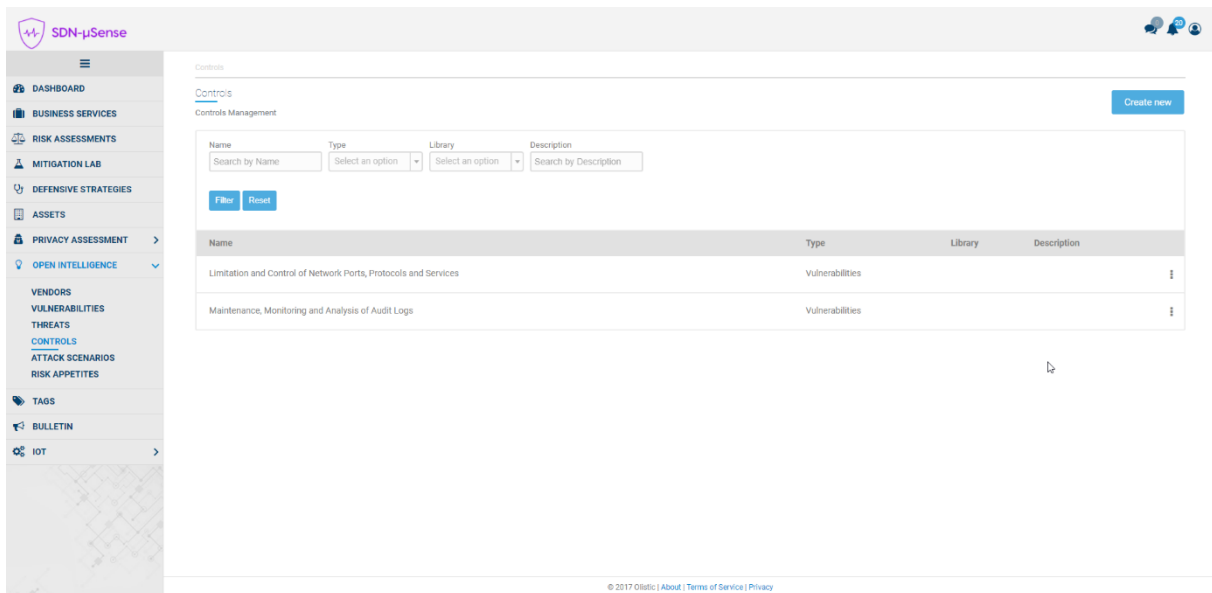


Figure 40: Executive CRL Summary

5.2.1.4 Controls management

The Mitigation Lab is the key aspect of the Controls management, where controls such as the CIS controls can be applied to mitigate a specific Attack Scenario. Figure 41 depicts the list of all identified Controls, while Figure 42 presents the Mitigation Lab, which gives the ability to manage, keep track and apply controls. In this way, the security administrator can manage the life cycle of the risk mitigation procedures and have a holistic view on the controls which have been adopted in order to mitigate the risks. Last but not least, the security administrator can compare (see Figure 43) how the applied controls affect the calculated risk and apply them accordingly.



Controls

Controls Management

Create new

Name Type Library Description

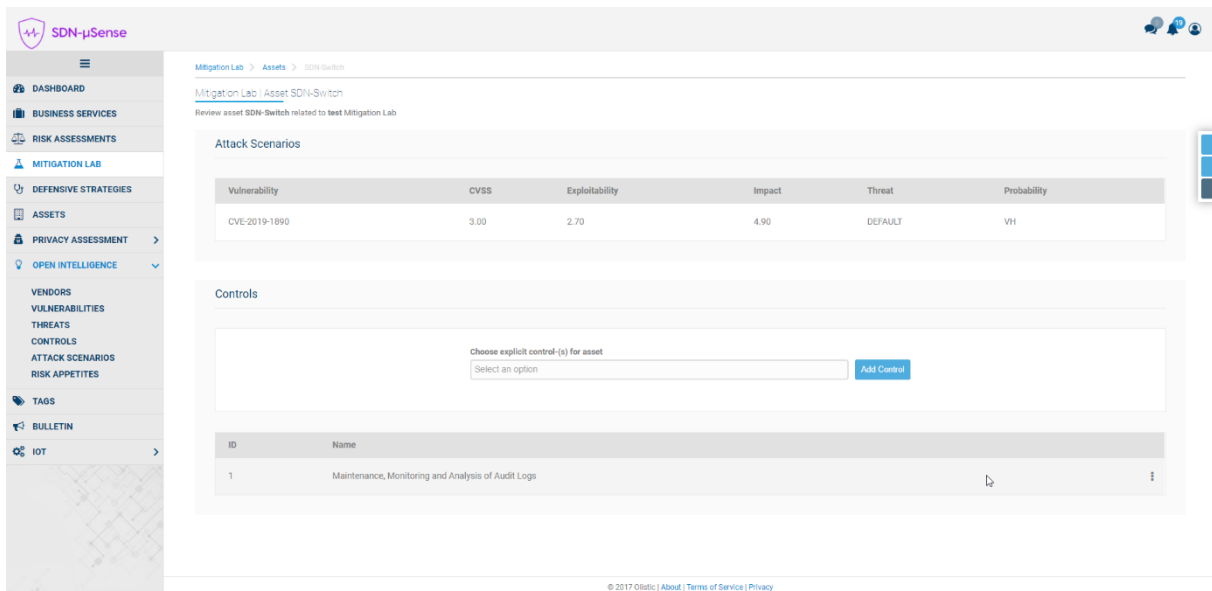
Search by Name Select an option Select an option Search by Description

Filter Reset

Name	Type	Library	Description
Limitation and Control of Network Ports, Protocols and Services	Vulnerabilities		
Maintenance, Monitoring and Analysis of Audit Logs	Vulnerabilities		

© 2017 Orlatic | About | Terms of Service | Privacy

Figure 41: Controls for Risk Mitigation



Mitigation Lab > Assets > SDN-Switch

Mitigation Lab | Asset SDN-Switch

Review asset SDN-Switch related to test Mitigation Lab

Attack Scenarios

Vulnerability	CVSS	Exploitability	Impact	Threat	Probability
CVE-2019-1890	3.00	2.70	4.90	DEFAULT	VH

Controls

Choose explicit control (-s) for asset

Select an option Add Control

ID	Name
1	Maintenance, Monitoring and Analysis of Audit Logs

© 2017 Orlatic | About | Terms of Service | Privacy

Figure 42: Assign Control to SDN-Switch

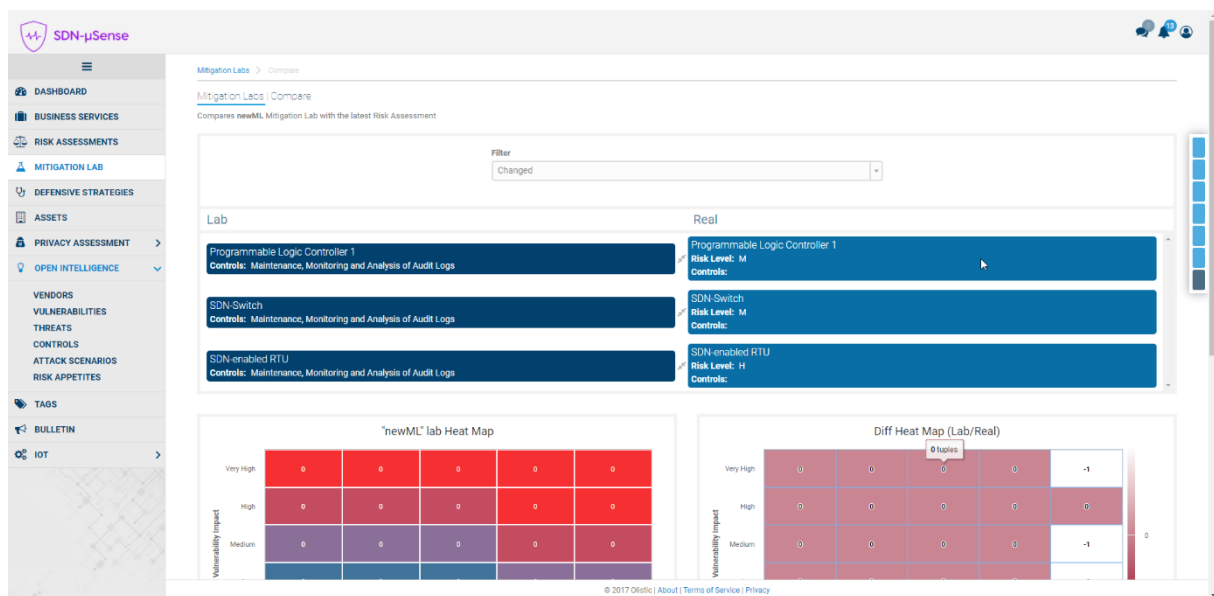


Figure 43: Mitigation Comparison

5.2.2 Mapping between Assets, Threat and possible attack patterns

This section documents part of the analysis conducted in the context of D3.2 [3] in order to steer S-RAF users in the process of assessing the cyber security risk and status of the EPES. More specifically, since S-RAF operations are asset-centric, it is vital to document the assets that can be included in the risk assessment. The SDN-microSENSE architecture is a mosaic of diverse technologies and consists of energy and SDN specific fields, while several mission-critical services are supported by legacy ICT assets. To this end, Table 13 offers a collection of assets which can be found in the EPES ecosystem and can be targeted by cyber security threats. It must be stated that Table 13 documents a subset of the assets that can be supported in S-RAF.

Given the list of assets, the S-RAF user can refer to Table 14, where a mapping among the threats, possible attacks and assets is provided. This mapping can steer S-RAF user through the interaction with the tool and offer a focused view on the case of EPES Risk Assessment. Each attack pattern captures knowledge about how specific parts of an attack are designed and executed and gives guidance on ways to mitigate the attack's effectiveness. Attack patterns help those developing defensive applications to better understand the specific elements of an attack and how to stop them from succeeding. Following this “know your enemy” strategy, a defender can increase the robustness of the deployed defensive mechanisms, but more importantly is in position to identify the imposed risks and have proper planning for mitigating them.

ID	Asset	Description
Assets related to the energy field		
AST-01	Smart Meter	A smart meter is an electronic device that records consumption of electric energy and communicates the information to the electricity supplier for monitoring and billing.
AST-02	Data Concentrators (Collectors)	An electronic device that interfaces with the sensors and transmits the obtained data to other system components.
AST-03	Advanced Metering Infrastructure (AMI) Head-end System	The head-end system (HES), also known as meter control system, is located within a metering company network (Distribution System Operator - DSO) and is directly communicating with the meters.

AST-04	IoT devices	Smart devices, mainly in possession of the end users of the smart grid, which may interact with smart meters.
AST-05	Programmable Logic Controller (PLC)	Digital computer used for automation of electromechanical processes, such as control of machinery on industrial ecosystems.
AST-06	Remote Terminal Unit (RTU)	Microprocessor-controlled electronic device that interfaces objects in the physical world to a distributed control system or SCADA system by transmitting telemetry data to a master system, and by using messages from the master supervisory system to control connected objects.
AST-07	Intelligent Electronic Device (IED)	Microprocessor-based controllers of power system equipment, such as circuit breakers, transformers, and capacitor banks.
AST-08	Distributed Control System (DCS)	System used to control a set of devices in a distributed environment.
AST-09	Meter Data Management System (MDMS)	Software that performs long-term data storage and management for the vast quantities of data delivered by smart metering systems.
AST-10	Master Terminal Unit (MTU) & Human Machine Interface (HMI)	A component responsible for the presentation of the data to human operators, usually including a console capable of monitoring and controlling the status of the operations.
AST-11	Phasor Measurement Unit (PMU)	A device used to estimate the magnitude and phase angle of an electrical phasor quantity (such as voltage or current) in the electricity grid using a common time source for synchronization.
AST-12	Phasor Data Concentrators (PDCs)	Receives and time-synchronizes phasor data from multiple phasor measurement units (PMUs) to produce a real-time, time-aligned output data stream. A PDC can exchange phasor data with PDCs at other locations.
AST-13	Control Centre / SCADA	The control centre undertakes all the monitoring and control processes. It is a control system consisting of computers, networked data communications and graphical user interfaces (GUI) for high-level process supervisory management.
AST-14	Distributed Energy Resources (DER)	Electric generation units (including Renewable Energy: solar and wind power) located within the electric distribution system located close to the load they serve. They are parallel to the electric utility or stand-alone units.
AST-15	Industrial control system (ICS)	Command and control systems designed to support industrial processes. These systems are responsible for monitoring and controlling a variety of processes and operations such as electricity distribution. The largest subgroup of ICS is SCADA (Supervisory Control and Data Acquisition) systems.
AST-16	Advanced Interrupting Switch	A distribution switch that can detect and interrupt faults quick and precisely.
AST-17	Microgrid Controller	Devices that control and enable the establishment of Microgrids.
AST-18	Microgrid	Electrical systems that include multiple loads and distributed energy resources that can be operated in parallel with the grid or as an electrical island
AST-19	Controllable/Regulating Smart Inverter	Inverter used to convert Direct Current (DC), the form of electricity produced by solar panels and batteries, to Alternating Current (AC). A controllable/regulating inverter can adjust its output to help control voltage and power factor.
AST-20	Automatic recloser (Recloser) (ACR)	Automatic circuit reclosers (ACRs) (also known as reclosers or autoreclosers) are a class of switchgear which are used for distribution automation. They detect and interrupt momentary faults. ACRs are high voltage rated circuit breakers used as an overhead network distribution protection asset.
AST-21	Backup power UPS	Short period fast response auxiliary power to support a control centre's operation for short power failures.
AST-22	Billing system	System responsible for analysing the energy consumption data for customer billing.
AST-23	Historian	A high-capacity system designed to collect and store the logs generated by the readings and operations of the sensors, assets, alarms and other events generated by plant devices, part of the network.
Assets related to the legacy ICT field		
AST-24	Databases	Organized collection of data generally stored and accessed electronically from a computer system. Several databases may exist in the EPES ecosystem to server diverse purposes.

AST-25	WLAN Access Point	Hardware networking device that allows other Wi-Fi devices to connect to a wired network. EPES devices may communicate over wireless networks.
AST-26	Services	Services which are offered to the entities of EPES (personnel, stakeholders, etc.), such as Mail, Terminal, Print, Authentication, File, Network, Name, Address Services. Those services are usually supported by general purpose servers and systems.
AST-27	Operating system (OS)	Operating system (OS) is a software which acts as an interface between the end user and a computer hardware. EPES devices may have different OSs, which in turn may have different vulnerabilities.
AST-28	Applications	The EPES ecosystem can be based on a plethora of applications. In the context of the asset documentation process in this deliverable we use this generic term to refer to any application which cannot specifically be assigned to an energy-specific application.
AST-29	Firewall	A network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules. A firewall typically establishes a barrier between a trusted internal network and untrusted external networks.
AST-30	DHCP	Network server that automatically provides and assigns IP addresses, default gateways and other network parameters to client devices.
AST-31	Domain Controller (DC)	Server that responds to authentication requests and verifies users on computer networks. DC is responsible for controlling host access to domain resources. It authenticates users, stores user account information and enforces security policy for a domain. Domains are a hierarchical way of organizing users and computers that work together on the same network.
AST-32	Network Components	Network components that can be found in legacy ICT topologies, such as Routers, Switches, Gateways, Workstations, Servers (Web, Mail, Authentication, Business).
AST-33	Human Assets (personnel)	This asset refers to any human in the SDN-microSENSE infrastructure from system and network administrators to simple end users.
Assets related to the SDN field		
AST-DP	Data Plane Assets	
AST-34	Programmable network components	In the context of an SDN the behaviour of network devices and flow control is handled by software that operates independently from network hardware. SDN - routers, -gateways, -switches are programmable network components that can be found in the SDN-microSENSE architecture.
AST-35	Control – Data plane Interface agent (CDPI agent)	The software component that realizes the northbound API of the network elements.
AST-36	SDN-enabled RTUs	RTUs able to operate in the context of Software Defined Networks. They can communicate via RTU controllers.
AST-37	SDN-enabled RTU Controller	SDN Controller for managing SDN-enabled RTUs to operate in the context of SDN-microSENSE project.
AST-38	Data plane software	Software used for supporting and managing the programmable network components in the data plane.
AST-CP	Control Plane Assets	
AST-39	Network Operating Systems	Assets that realize the control plane for a software-defined network (SDN), managing network components, such as switches and links, and running software programs controlling the creation and destruction of network flows and paths. (e.g. OpenDayLight, ONOS, etc.)
AST-40	Cryptographic Components	Provide encryption to the communications that pass through the SDN stack, among the assets of the Application, Control and Data planes.
AST-41	Control plane software	Software used for supporting and managing the programmable network services in the control plane.
AST-AP	Application Plane Assets	
AST-42	SDN Applications	Applications that manage specific operations of the complex SDN topology such as, Network Visualization, Service Provisioning, Network Management, load balancing applications.

AST-43	SDN user	This asset refers to any User that is using equipment attached to the Data plane of an SDN deployment through the application and control planes.
---------------	----------	---

Table 13: Assets of the Electrical Power and Energy System ecosystem in the context of SDN-microSENSE

Threat type	Threat	CAPEC	Assets
Nefarious Activity/Abuse	Manipulation of network configuration / Data forging	CAPEC-210: Abuse Existing Functionality CAPEC-113: API Manipulation CAPEC-148: Content Spoofing CAPEC-153: Input Data Manipulation CAPEC-141: Cache Poisoning CAPEC-166: Force the System to Reset Values CAPEC-165: File Manipulation CAPEC-176: Configuration/Environment Manipulation CAPEC-438: Modification During Manufacture CAPEC-439: Manipulation During Distribution CAPEC-137: Parameter Injection CAPEC-548: Contaminate Resource CAPEC-137: Parameter Injection	AST-[CP,DP, AP, 29, 30, 31, 32, 25, 26]
	Software/firmware exploits	CAPEC-113: API Manipulation CAPEC-184: Software Integrity Attack CAPEC-165: File Manipulation CAPEC-441: Malicious Logic Insertion CAPEC-137: Parameter Injection CAPEC-175: Code Inclusion CAPEC-175: Code Inclusion	Virtually all cyber-enabled assets are exposed to this threat. Special focus on APIs, SDN assets, and services/applications with large codebase.
	Denial of Service (DoS)	CAPEC-125: Flooding CAPEC-272: Protocol Manipulation CAPEC-152: Inject Unexpected Items CAPEC-113: API Manipulation CAPEC-130: Excessive Allocation CAPEC-624: Fault Injection CAPEC-594: Traffic Injection CAPEC-469: HTTP DoS	Any asset that exposes a network commutation port can be targeted by this threat. Special focus on AST-[DP, CP, AP, 32, 30, 26, 25, 01-13, 17]
	Remote SDN application exploitation	CAPEC-113: API Manipulation CAPEC-21: Exploitation of Trusted Credentials CAPEC-180: Exploiting Incorrectly Configured Access Control Security Levels CAPEC-50: Password Recovery Exploitation CAPEC-114: Authentication Abuse CAPEC-115: Authentication Bypass CAPEC-225: Subvert Access Control	AST-AP
	Remote access exploitation	CAPEC-114: Authentication Abuse CAPEC-115: Authentication Bypass CAPEC-225: Subvert Access Control CAPEC-151: Identity Spoofing CAPEC-113: API Manipulation CAPEC-22: Exploiting Trust in Client CAPEC-50: Password Recovery Exploitation	Any cyber-enabled asset that exposes a remote access service visible either inside or outside of the internal network zone

			can be affected. Particular focus on AST-[13, 15, 18, 26, 28, AP]
	SDN API exploitation	CAPEC-113: API Manipulation CAPEC-225: Subvert Access Control	AST-[DP, CP, AP]
	Malicious code/Software and Malicious software updates	CAPEC-186: Malicious Software Update CAPEC-185: Malicious Software Download CAPEC-187: Malicious Automated Software Update CAPEC-533: Malicious Manual Software Update CAPEC-17: Using Malicious Files CAPEC-636: Hiding Malicious Data or Code within Files CAPEC-444: Development Alteration CAPEC-523: Malicious Software Implanted CAPEC-441: Malicious Logic Insertion CAPEC-542: Targeted Malware CAPEC-17: Using Malicious Files CAPEC-442: Infected Software	Any cyber-enabled assets which includes a software stack could be affected. Among other, AST-[10, 24, 27 28, 31, 42, 24, 22]
	Unauthorized activities	CAPEC-114: Authentication Abuse CAPEC-21: Exploitation of Trusted Credentials CAPEC-115: Authentication Bypass CAPEC-122: Privilege Abuse CAPEC-233: Privilege Escalation	Several assets can be affected by unauthorised access. AST-[DP, CP, AP, 29, 31, 24, 27, 13, 33, 42, 53]
	Virtualisation threats	CAPEC-480: Escaping Virtualization CAPEC-189: Black Box Reverse Engineering	AST-[DP, CP, AP, 32, 24, 28, 27]
	Identity theft/spoofing (e.g. Credentials stealing trojans)	CAPEC-151: Identity Spoofing CAPEC-473: Signature Spoof CAPEC-21: Exploitation of Trusted Credentials	Any asset which engages an authentication process for granting access can be affected. AST-[DP, CP, AP, 13, 22, 23, 24, 26-29, 31]
	Social Engineering	CAPEC-403: Social Engineering CAPEC-416: Manipulate Human Behaviour CAPEC-185: Malicious Software Download CAPEC-23: File Content Injection CAPEC-41: Using Meta-characters in E-mail Headers to Inject Malicious Payloads	AST-[33, 43]
	Advanced Persistent Threats (APTs)	CAPEC-185: Malicious Software Download CAPEC-187: Malicious Automated Software Update CAPEC-17: Using Malicious Files CAPEC-636: Hiding Malicious Data or Code within Files CAPEC-523: Malicious Software Implanted CAPEC-542: Targeted Malware CAPEC-17: Using Malicious Files CAPEC-442: Infected Software CAPEC-118: Collect and Analyse Information CAPEC-152: Inject Unexpected Items	This threat implies the unauthorised access of assets using various combinations of attacks (malware, social engin., Identity theft, etc.). Virtually

			any cyber-enabled asset can be affected.
	Exploit Protocol vulnerabilities	CAPEC-272: Protocol Manipulation CAPEC-192: Protocol Analysis CAPEC-210: Abuse Existing Functionality CAPEC-125: Flooding	AST-[DP, CP, AP, 01-20, 25, 26, 28, 32]
Eavesdropping/Interception/ Hijacking	Traffic diversion	CAPEC-272: Protocol Manipulation CAPEC-94: Man in the Middle Attack CAPEC-272117Interception CAPEC-216: Communication Channel Manipulation CAPEC-272: Protocol Manipulation CAPEC-142: DNS Cache Poisoning	AST-[DP, CP, 32]
	Side channel attack	CAPEC-189: Black Box Reverse Engineering CAPEC-622: Electromagnetic Side-Channel Attack CAPEC-118: Collect and Analyse Information CAPEC-118: Fingerprinting	AST-[DP, CP, 32]
	Memory scraping	CAPEC-545: Pull Data from System Resources CAPEC-546: Probe Application Memory CAPEC-123: Buffer Manipulation CAPEC-540: Overread Buffers CAPEC-129: Pointer Manipulation CAPEC-456: Infected Memory	AST-[AP, 26, 27, 28, 33]
	Traffic sniffing	CAPEC-158: Sniffing Network Traffic CAPEC-31: Accessing/Intercepting/Modifying HTTP Cookies	AST-[DP, CP, AP, 33]
	Man in the middle (MITM)	CAPEC-117: Interception	AST-[DP, CP, AP, 03, 33]
	Interception of Information	CAPEC-651: Eavesdropping CAPEC-94: Man in the Middle Attack	AST-[DP, CP, AP]
	Replay of messages	CAPEC-60: Reusing Session IDs (aka Session Replay) CAPEC-272: Protocol Manipulation	Affects assets that communicate using protocols without message replay protection.
	Network Reconnaissance and Information gathering	CAPEC-118: Collect and Analyse Information CAPEC-117: Interception CAPEC-169: Footprinting CAPEC-224: Fingerprinting CAPEC-188: Reverse Engineering CAPEC-192: Protocol Analysis	Virtually all cyber enabled assets are prone to network reconnaissance threats.

Table 14: Threats, attack patterns and affected assets for EPES

It must be stated that the mapping of identified threats, attack patterns and assets are a subset of the possible combinations that can occur due to the constantly evolving threat landscape of EPES and the increased sophistication of cyber-attacks. However, Table 14 can be used as a concrete point of reference to steer S-RAF users to make informed decisions and gain a better understanding to the risk assessment process, and has been already used as a starting point for the demonstrators' scenarios planning.

5.3 S-RAF in the Context of Demonstrators

Although the validation performed in the context of the demonstrators is part of the WP8 that is officially at M22 of the project, we consider that providing early access of S-RAF to the demonstrators would help eliminate issues and possible delays. For this purpose, S-RAF (without the integration to other external components) has been installed and provided by UBITECH in an online demonstration setup, in order to start working in the scope of demonstrators and collect initial feedback. The most important part of the work performed at this stage was to start populating the model of the pilot EPES, by collecting and defining in EPES assets (virtual appliances and services, protocols used, hardware or even people, collecting vulnerabilities and threats (e.g. a sec admin uses an easy password, that doesn't comply to the rules).

For this purpose, dedicated virtual workshops with each demonstrator were organized during July 2020. Through this exercise we collected an initial set of assets used by the pilots (presented in Table 15), and also agreed on some conventions (e.g. that protocols can be defined as a separate asset and the relation IS_USED_BY can be used by all assets using this protocol.

Asset Name	Asset Type	Business Value	Details	Run Privileges	Asset Dependencies	CVEs	Possible Threat Scenarios
Ubuntu Server	Operating system	Low	Ubuntu 16.04 LTS	Local User, Local Admin	INSTALLED_ON: Server, IS_USED_BY: Administrator	CVE-2019-1000019	- Flooding - Denial of Service
Gateway	Hardware	Low	Cisco	Local Admin	IS_LOCATED_IN: At each customers site	CVE-2018-18068	- Flooding - Denial of Service
Raspbian OS	Operating system	Medium	Raspberry Pi 1 B+	Local Admin	INSTALLED_ON: Gateway	CVE-2018-18068	- Flooding - Denial of Service
SDN-Switch	Hardware	High	N/A	Local Admin	CONNECTED_TO: SDN-enabled RTU, Communication front-end	CVE-2019-1010245	- Flooding - Denial of Service
SDN-enabled RTU	Hardware	Medium	N/A	Local Admin	CONNECTED_TO: SDN-Switch	CVE-2020-15781	- Denial of Service - Execute protocol commands - Gain information (eavesdropping) - Message reply
LCU	Hardware	Medium	Local control Units	Local Admin	CONNECTED_TO: SDN-Switch	N/A	- MITM - Denial of Service - Spoofing - Gain information (eavesdropping) - Message reply
Electrical Protection s	Hardware	Medium	N/A	Local Admin	CONNECTED_TO: SDN-Switch, SDN-enabled_RTU	N/A	- Denial of Service - Execute commands
IEC-60870-5-104	Protocol	Low	N/A	N/A	IS_USED_BY: RTU, LCU, Electrical protections	N/A	- Eavesdropping to obtain parameters knowledge
IEC-61850	Protocol	Low	N/A	N/A	IS_USED_BY: RTU, LCU,	N/A	- Eavesdropping to obtain topology knowledge

					Electrical protections		
Communication Front-end	Hardware	High	N/A	Local Admin	CONNECTED_TO:SDN-Switch and Server	N/A	<ul style="list-style-type: none"> - Flooding - Denial of Service - Manipulation of network configuration
SCADA	Software	Very High	Schneider Electric	Local Admin	INSTALLED_ON: Server	CVE-2020-7523	<ul style="list-style-type: none"> - Gain information - Scale privileges - Unauthorized access (eavesdropping/execute commands) - SQL injection - Receive data from spoofed devices
Server	Hardware	Very High	Dell Poweredge T110	Local User, Local Admin	CONNECTED_TO: Communication front-end	CVE-2020-5330	<ul style="list-style-type: none"> - Exploit OS vulnerabilities
Administrator	Personnel	N/A	N/A	N/A	N/A	N/A	<ul style="list-style-type: none"> - Unauthorized usage

Table 15: Representative assets of demonstrators, for S-RAF

It has to be mentioned that this table includes only some representative data collected from all pilots and due to privacy reasons, this data is only examples and not real data. For the collection of this content all demonstrators have been provided with a dedicated spreadsheet that will be used further for the preparing the actual setup or work for WP8 needs.

In addition to the definition of the EPES model, through the workshops with the demonstrators we examined the deployment options of SDN-microSENSE and the topologies to be used by the pilots. Again, although the demonstrators are still in planning phase, the initial list of assets can be defined in S-RAF, and the plans for deployment of SDN-microSENSE (including S-RAF) shall not create any issue, as S-RAF actually needs to communicate only with some of the platform components (XL-SIEM, AIDB, AIREC) and not the demonstrator assets.

6 Conclusions

This document concluded the work performed in the scope of WP3 and had as result the creation of SDN-microSENSE Risk Assessment Framework (S-RAF). At first an analysis of existing tools for risk assessment are presented along with a wrap-up of the methodology of S-RAF that has been presented in deliverable D3.1 [2] with more details. The deliverable includes also the S-RAF architecture with the components and the corresponding interfaces, the implementation details as well as the installation and usage instructions.

There are multiple risk assessment metrics that can be adopted from the literature, however, there is no specific framework that focuses on the EPES infrastructure. S-RAF is an innovative risk assessment solution targeted on EPES that considers the cumulative aspects of the needed to involve all stakeholders of the energy components. Currently, there are several tools in the market that quantify the risk, but without considering the aforementioned cumulative aspects. S-RAF cumulative risk assessment approach enables one to perceive the security state at the level of mission-critical assets that belong either in the same business workflow, or in the same physical (or virtual) networks.

In addition, the main updates of S-RAF compared to the UBITECH's OLISTIC Enterprise Risk Management Suite, that it is based upon, are:

1. Usage of updated model that supports EPES
2. Collaborative Risk Assessment with the cumulative RA
3. Connecting with AIDB for EPES asset retrieval
4. Connection with eVul for automated analysis of vulnerabilities in the EPES environment
5. Extending OLISTIC model and components for retrieving alerts from XL-SIEM
6. Providing Incidents based on the Risk Assessment as output to SDN-SELF and ARIEC components of SDN-microSENSE
7. Integrating Apache Kafka as message queue that can be used by both internal and external components
8. Transforming data to MISP format

The architecture of S-RAF is presented in this document, with focus on the presentation of the interdependencies and integration of the various components, while we also present some of the key aspects of the implementation of S-RAF. As this document helps the reader to have a first acquaintance with S-RAF, section 5 provides basic information about the installation and the usage of the platform. Although the evaluation of S-RAF as part of SDN-microSENSE will be performed in the scope of WP8, the preliminary usage of the S-RAF by the demonstrators has been executed to validate the suitability of the developed solution and methodology. Last but not least, the actual interfaces and integration points (especially the external component interfaces) could be updated in the context of WP7.

7 References

- [1] microSENSE, D2.2 User & Stakeholder, Security and Privacy Requirements,, 2020.
- [2] microSENSE, D3.1 Risk Assessment Methodology of Energy Chain, 2020.
- [3] microSENSE, D3.2 Threat and Vulnerability Model for EPES, 2020.
- [4] microSENSE, D2.3 Platform Specification & Architecture, 2020.
- [5] Bitsight, <https://www.bitsight.com/>.
- [6] NormShield, <https://www.normshield.com/>.
- [7] SecurityScorecard, <https://securityscorecard.com/>.
- [8] UpGuard, <https://www.upguard.com/>.
- [9] RiskRecon, <https://www.riskrecon.com/>.
- [10] microSENSE, D3.3 Development of EPES Honeypots for Advanced Threat Detection, 2020.
- [11] neo4j, <https://neo4j.com/>.
- [12] Thymeleaf, <https://www.thymeleaf.org/>.
- [13] Apache Kafka, <https://kafka.apache.org/>.
- [14] PostgreSQL, <https://www.postgresql.org/>.
- [15] Redis, <https://redis.io/>.
- [16] ISO 31000 Risk Management, <https://www.iso.org/iso-31000-risk-management.html>.
- [17] microSENSE, D2.4 Pilot, Demonstration & Evaluation Strategy, 2020.

8 Annex I – Docker Compose for S-RAF installation

```
version: '3'
services:
  openvas:
    image: "mikesplain/openvas:9"
    ports:
      - "14434:443"
      - "19394:9390"
    volumes:
      - /media/your_path/sraf/openvas/data:/var/lib/openvas/mgr
  sraf:
    build: ./sraf
    ports:
      - "8091:8080"
    volumes:
      - ./sraf-init:/opt/sraf
    depends_on:
      - mysql.dev.sraf.io
      - mongo.dev.sraf.io
      - openvas
    environment:
      - sraf.management.vulnerability.auto=true
      - sraf.management.vulnerability.openvas.host=openvas
      - sraf.management.vulnerability.openvas.port=9390
      - sraf.management.vulnerability.openvas.user=you_name
      - sraf.management.vulnerability.openvas.pass=your_pass
      - sraf.startup.inventory.asset.location=/opt/sraf/initial_hosts
    restart: always
  mongo.dev.sraf.io:
    image: "mongo:3.4"
    volumes:
      - /media/your_path/sraf/mongodb:/data/db
  mysql.dev.sraf.io:
    build: ./db-init
    volumes:
      - /media/your_path/sraf/mysql:/var/lib/mysql
    environment:
      - MYSQL_ROOT_PASSWORD= your_pass
      - MYSQL_USER= you_name
      - MYSQL_PASSWORD= your_pass
      - MYSQL_DATABASE=databasename
  zoo1:
    image: zookeeper:3.4.9
    hostname: zoo1
    ports:
      - "2181:2181"
    environment:
      ZOO_MY_ID: 1
      ZOO_PORT: 2181
```

```
ZOO_SERVERS: server.1=zoo1:2888:3888
volumes:
- ./zk-single-kafka-single/zoo1/data:/data
- ./zk-single-kafka-single/zoo1/datalog:/datalog

kafka1:
  image: confluentinc/cp-kafka:5.5.1
  hostname: kafka1
  ports:
  - "9092:9092"
  environment:
    KAFKA_ADVERTISED_LISTENERS:
LISTENER_DOCKER_INTERNAL://kafka1:19092,LISTENER_DOCKER_EXTERNAL://{DOCKER_HOST_I
P:-127.0.0.1}:9092
    KAFKA_LISTENER_SECURITY_PROTOCOL_MAP:
LISTENER_DOCKER_INTERNAL:PLAINTEXT,LISTENER_DOCKER_EXTERNAL:PLAINTEXT
    KAFKA_INTER_BROKER_LISTENER_NAME: LISTENER_DOCKER_INTERNAL
    KAFKA_ZOOKEEPER_CONNECT: "zoo1:2181"
    KAFKA_BROKER_ID: 1
    KAFKA_LOG4J_LOGGERS:
"kafka.controller=INFO,kafka.producer.async.DefaultEventHandler=INFO,state.change.logger=INF
O"
    KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 1
  volumes:
  - ./zk-single-kafka-single/kafka1/data:/var/lib/kafka/data
  depends_on:
  - zoo1
```

9 Annex II –Unit Tests

The unit test cases reported using a table describing the test, its preconditions, the input and the actual steps. SPEC ID refers to relevant specifications as have been defined in SDN-microSENSE D2.3 deliverable. Finally, the results method tested and the actual test results are presented. For the implementation of the tests Junit framework² has been used.

Test Case ID	SRAF_01	Component	Threat Identification Component
Description	RabbitMQ consumer tester		
SPEC ID	SPEC-F3, SPEC-F4	Priority	Medium
Prepared by	UBITECH	Tested by	UBITECH
Pre-condition(s)	<ul style="list-style-type: none">• A RabbitMQ available• atos.exchange.alarms.sdnmsense.cis available on RabbidMQ• Credentials properly configured on Threat Identification Component		
Test steps			
1	Read queue with input		
2	Read empty queue		
Input data	N/A		
Result	Retrieved and printed the Json Object of the Event (see CIS-RAF interface example)		
Test Case Result	Achieved, with test RabbitMQ setup		

Test Case ID	SRAF_02	Component	Assets Identification Component
Description	Get assets from AIDB tester		
SPEC ID	SPEC-F7	Priority	Medium
Prepared by	UBITECH	Tested by	UBITECH
Pre-condition(s)	<ul style="list-style-type: none">AIDB installed and populated with test dataCredentials properly configured on Threat Identification Component		
Test steps			
1	Rest call to /assets_inventory_query		
2	Parse results		
3	Store to MongoDB		
Input data	N/A		
Result	Assert proper storage of data to the database (based on the response of Mongo)		
Test Case Result	Achieved with dummy REST call responses based on documentation, to be fully tested in integrated version of the platform		

Test Case ID	SRAF_03	Component	Assets Identification Component
Description	Get topology from AIDB tester		
SPEC ID	SPEC-F7	Priority	Medium

² <https://junit.org/>

Prepared by	UBITECH	Tested by	UBITECH
Pre-condition(s)	<ul style="list-style-type: none">AIDB installed and populated with test dataAIDB Credentials properly configured on Threat Identification Component		
Test steps			
1	Rest call to /topology_query		
2	Parse results		
3	Store to MongoDB		
4	Store to Neo4J		
Input data	N/A		
Result	Assert proper storage of data to the database (based on the response of Mongo and Neo4J)		
Test Case Result	Achieved with dummy REST call responses based on documentation, to be fully tested in integrated version of the platform		

Test Case ID	SRAF_04	Component	S-RAF Impact Analysis Component
Description	Get Vulnerabilities for asset tester		
SPEC ID	SPEC-F4	Priority	Medium
Prepared by	UBITECH	Tested by	UBITECH
Pre-condition(s)	<ul style="list-style-type: none">eVul containers runningVulnerabilities available from eVulValid assetIDeVul connection configured on Impact Identification Component		
Test steps			
1	REST call to eVul assetinfo method		
2	Parse results		
3	Store to MySql		
Input data	assetID		
Result	Assert proper storage of data to the database		
Test Case Result	Achieved		

Test Case ID	SRAF_05	Component	S-RAF Impact Analysis Component
Description	Attack Path retrieval tester		
SPEC ID	SPEC-F4	Priority	Medium
Prepared by	UBITECH	Tested by	UBITECH
Pre-condition(s)	<ul style="list-style-type: none">Assets and their connection retrieved from AIDBAsset topology stored in the Asset Identification Component databasesRisk assessment Object and corresponding ID must be created		
Test steps			
1	Call discoverattackpaths REST call with defined ID		
2	Retrieve results		
3	Assert results (based on predefined ID and topology)		

Input data	Risk assessment id
Result	Retrieved and printed the Json Object of the attack path
Test Case Result	Achieved

Test Case ID	SRAF_06	Component	S-RAF Risk Assessment Component	Level
Description	Cumulative Risk Assessment results tester			
SPEC ID	SPEC-F4	Priority	Medium	
Prepared by	UBITECH	Tested by	UBITECH	
Pre-condition(s)	<ul style="list-style-type: none">Created Risk Assessment Object (containing at least 3 assets, threats, vulnerabilities), with riskassessmentService.createCreate BusinessParter object (representing organization)All components of S-RAF up and running			
Test steps				
1	Execute riskassessmentService. countRiskassessmentsForBusinesspartner			
2	Write result			
3	Assess the test by examining if valind Long number is provided			
Input data	N/A			
Result	Valid Long number			
Test Case Result	Achieved			

Test Case ID	SRAF_07	Component	S-RAF Risk Assessment Component	Level
Description	Kafka exporter for ADAE			
SPEC ID	SPEC-F4, SPEC-F6	Priority	Medium	
Prepared by	UBITECH	Tested by	UBITECH	
Pre-condition(s)	<ul style="list-style-type: none">• Kafka up and running, topic sraf.out.adae has been created• XL-SIEM input provided• Vulnerabilities retrieved from eVul• Generation of assessment output for ADEA			
Test steps				
1	Connect to Kafka			
2	Write result			
3	Read result to assess the test			
Input data	N/A			
Result	Retrieved and printed the Json Object of the assessment results that were published in step 2.			
Test Case Result	Achieved			

Test Case ID	SRAF_08	Component	S-RAF Risk Assessment Component	Level
Description	MISP transformation			
SPEC ID	N/A	Priority	Medium	
Prepared by	UBITECH	Tested by	UBITECH	
Pre-condition(s)	<ul style="list-style-type: none">• Risk Assessment Results available• MISP format of the result available for assertion			
Test steps				
1	Prepare String input			
2	Execute Utli.transalateToMisp()			
3	Assert results			
Input data	<ul style="list-style-type: none">• Risk Assessment Result including vulnerabilities provided as text			
Result	Assertion of output text			
Test Case Result	Achieved			

Test Case ID	SRAF_09	Component	S-RAF Risk Assessment Component	Level
Description	Kafka exporter for ARIEC			
SPEC ID	SPEC-F4, SPEC-F5	Priority	Medium	
Prepared by	UBITECH	Tested by	UBITECH	
Pre-condition(s)	<ul style="list-style-type: none">• Kafka up and running, topic sraf.out.ariec has been created• XL-SIEM input provided• Vulnerabilities retrieved from eVul• Generation of assessment output with the MISP format agreed for ARIEC			
Test steps				
1	Connect to Kafka			
2	Write result			
3	Read result to assess the test			
Input data	N/A			
Result	Retrieved and printed the Json Object of the assessment results that were published in step 2.			
Test Case Result	Achieved			