



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra



SMESEC

Protecting Small and Medium-sized Enterprises digital technology through an innovative cyber-SECurity framework

D3.9 SMESEC Framework Public Report-Final Version

Document Identification			
Status	Final	Due Date	31/05/2020
Version	1.0	Submission Date	08/06/2020
Related WP	WP3	Document Reference	D3.9
Related Deliverable(s)	D3.1, D3.2, D3.3, D3.4 D3.5, D4.9, D5.2, D5.5	Dissemination Level (*)	PU
Lead Organization	Citrix	Lead Author	Christos Tselios
Contributors	All Partners	Reviewers	Omri Soceanu, IBM Ciprian Oprisa, BD

Keywords:

security, system, design, architecture, integration, WP3, requirements, stakeholder, goals, innovation, protection, defence, management, context, concept, pattern, composition, interface, rationale, sequence, response, forensics, orchestration, hub.

This document is issued within the frame and for the purpose of the SMESEC project. This project has received funding from the European Union's Horizon2020 Framework Programme H2020-DS-SC7-2016 under Grant Agreement No. 740787 and supported by Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number 17.00067. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission.

This document and its content are the property of the SMESEC Consortium. All rights relevant to this document are determined by the applicable laws. Access to this document does not grant any right or license on the document or its contents. This document or its contents are not to be used or treated in any manner inconsistent with the rights or interests of the SMESEC Consortium or the Partners detriment and are not to be disclosed externally without prior written consent from the SMESEC Partners.

Each SMESEC Partner may use this document in conformity with the SMESEC Consortium Grant Agreement provisions.

(*) Dissemination level.-**PU**: Public, fully open, e.g. web; **CO**: Confidential, restricted under conditions set out in Model Grant Agreement; **CI**: Classified, **Int** = Internal Working Document, information as referred to in Commission Decision 2001/844/EC.

Document Information

List of Contributors	
Name	Partner
Christos Tselios	Citrix
Francisco Hernandez, Olmo Rayón	WoS
Jose Francisco Ruiz, Pablo Barrientos	ATOS
Manos Athanatos	FORTH
Kostas Lampropoulos	UoP
Ciprian Oprisa	BD
Philippe Cousin	EGM
Samuel Fricker	FHNW
Omri Soceanu	IBM

Document History			
Version	Date	Change editors	Changes
0.1	10/05/2020	Citrix	Initial content
0.2	11/05/2020	Citrix	Adding final guidelines
0.8	01/06/2020	Citrix	Integrated Partner contribution
0.9	04/06/2020	Citrix	Additional Partner contribution integration
0.91	05/06/2020	Citrix	Final check
1.0	08/06/2020	ATOS	Quality check and submission to EC

Quality Control		
Role	Who (Partner short name)	Approval Date
Deliverable leader	Christos Tselios (Citrix)	05/06/2020
Technical manager	Christos Tselios (Citrix)	05/06/2020
Quality manager	Rosanna Valle Soriano (Atos)	08/06/2020
Project Manager	Jose Fran. Ruíz (Atos)	08/06/2020

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	2 of 85
Reference:	D3.9	Dissemination:	PU
	Version:	1.0	Status:
			Final

Table of Contents

Document Information	2
Table of Contents	3
List of Tables.....	6
List of Figures	7
List of Acronyms.....	9
Executive Summary	11
1 Introduction.....	12
1.1 Purpose of the document	12
1.2 Relation to other project work.....	12
1.3 Structure of the document	12
2 SMESEC Framework design	13
2.1 Roles and Use Cases.....	13
2.2 High-level architecture diagram	16
2.3 Composition View.....	18
2.4 Component View.....	21
2.5 Interface View	24
2.6 Deployment View.....	29
2.7 Communication Bus Security	30
3 SMESEC Framework User Experience	32
3.1 Personas.....	32
3.2 Functions	35
3.2.1 Overarching User Interface Design Decisions	36
3.2.2 View: SMESEC Dashboard	38
3.2.3 View: SMESEC-At-My-Company.....	41
3.2.4 View: Tool View	43
3.2.5 New View: Hub Configuration View.....	43
3.2.6 New View: My Status View.....	44
3.2.7 View: SMESEC Tools Dashboard	45
3.2.8 New View: Tools Configuration View.....	47
3.3 Navigation	47

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	3 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

4	Design of SMESEC Framework Hub	49
4.1	System Architecture	49
4.1.1	Global perspective	49
4.1.2	Solution providers: cybersecurity solutions and data ingestion	50
4.1.3	Core Module	50
4.1.4	Data Consumption	52
4.2	Interfaces and Connectivity	52
4.2.1	Input	52
4.2.2	Output	53
5	SMESEC Framework Implementation	55
5.1	Description and Objectives	55
5.2	Integrated Tools and Core Functionality	55
5.2.1	XL-SIEM	55
5.2.2	Citrix ADC and Aggregator	56
5.2.3	Gravity Zone Endpoint	60
5.2.4	EWIS	61
5.2.5	CySec	68
5.2.6	Test-as-a-Service	69
5.2.7	Virtual Patching	71
5.2.8	Testing Platform - ExpliSAT	72
5.2.9	Training Platform	73
5.2.10	Moving Target	73
5.2.11	HUB	74
5.3	Development Environment and Frameworks	77
5.4	Integration Methodology	77
5.5	Technical Infrastructure	78
5.6	Authentication and Security	78
5.7	Deployment and Configuration	79
5.7.1	Deployment and configuration of the SMESEC Framework core	79
5.7.2	Updates	81
5.8	API for external tools	81
5.9	Functionality, Characteristics and GUI Navigation	82
5.10	Framework Evaluation	82

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	4 of 85
Reference:	D3.9	Dissemination:	PU	Version:	1.0
				Status:	Final

6	Conclusions.....	83
	References	84

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	5 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

List of Tables

<i>Table 1: Overview of SMESEC Framework Components together with their Input and Output information</i>	__	21
<i>Table 2: Interfaces of SMESEC Framework components</i>	_____	25
<i>Table 3: Deployment of SMESEC components</i>	_____	30
<i>Table 4: Element Updates of the Tool View UI.</i>	_____	43
<i>Table 5: SMESEC Framework Deployment Infrastructure</i>	_____	78

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	6 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

List of Figures

<i>Figure 1: Use case diagram – administrator</i>	14
<i>Figure 2: Use case diagram – security analyst</i>	14
<i>Figure 3: Use case diagram – reporter</i>	15
<i>Figure 4: Use case diagram – auditor</i>	15
<i>Figure 5: Use case diagram – user</i>	16
<i>Figure 6: high-level architecture diagram</i>	16
<i>Figure 7: SMESEC composition view</i>	19
<i>Figure 8: Partial SMESEC setup</i>	20
<i>Figure 9: Basic SMESEC setup</i>	20
<i>Figure 10: Community layer: extended roles supported by the SMESEC Framework.</i>	33
<i>Figure 11: Frame for the SMEEC Framework Views</i>	36
<i>Figure 12: SMESEC Tools Ribbon Menu.</i>	37
<i>Figure 13: SMESEC Dashboard.</i>	38
<i>Figure 14: Answering of awareness question.</i>	39
<i>Figure 15: Alert widget.</i>	39
<i>Figure 16: Answering of the guidance question.</i>	40
<i>Figure 17: The SMESEC-At-My-Company View</i>	42
<i>Figure 18: Hub Configuration View.</i>	43
<i>Figure 19: SMESEC Tools Dashboard.</i>	46
<i>Figure 20: SMESEC Tools Configuration view.</i>	47
<i>Figure 21: Same navigation paradigm as described in D3.3.</i>	48
<i>Figure 22: General sketch of the data lifecycle of data within the SMESEC framework Hub</i>	50
<i>Figure 23: End-to-end architecture of the SMESEC Framework Hub</i>	50
<i>Figure 24: SMESEC Framework Hub Output Information</i>	53
<i>Figure 25: SMESEC Framework dashboard. XL-SIEM section</i>	56
<i>Figure 26. High-level network topology for Citrix ADC deployment in SMESEC</i>	58
<i>Figure 27: Deploying Citrix ADC Aggregator</i>	60
<i>Figure 28 - New Connections Alert: Home Page</i>	62
<i>Figure 29 - EWIS responsive menu</i>	62
<i>Figure 30 - Top IP/Port Statistics</i>	63
<i>Figure 31 - Attack Maps</i>	63
<i>Figure 32 - How complex is to install the agent of the Honeypot per Company?</i>	64
<i>Figure 33 - How complex is to uninstall/remove the agent of the Honeypot from your system?</i>	64
<i>Figure 34 - How useful were the instructions (e.g. documentation, videos) for installing/configuring the Honeypot?</i>	65
<i>Figure 35 - Did you have to prepare your system before installing the clients/agents?</i>	65
<i>Figure 36 - Did you have to update/install additional software for installing a component?</i>	66
<i>Figure 37 - Average grade per Question for all the concerned SMEs</i>	66
<i>Figure 38. EGM TaaS Architecture</i>	70
<i>Figure 39. EGM offline testing</i>	70
<i>Figure 40. TaaS Micro service Architecture</i>	71
<i>Figure 41 AngelEye solution architecture</i>	72
<i>Figure 42: Hybrid testing platform</i>	72
<i>Figure 43: Anti-ROP for binary</i>	73

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	7 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

Figure 44: Anti-ROP for source 74

Figure 45: General view of the SMESEC framework front-end. The outputs from the Hub are displayed in the upper-right part. 74

Figure 46: Setting-up Rules 75

Figure 47: Responses plans generated for Industrial Pilot (example) 76

Figure 48: Automatic emails sent by the system when key events occur 76

Figure 49 - SMESEC Framework infrastructure deployment 80

Figure 50: SMESEC Framework Core 81

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	8 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

List of Acronyms

Abbreviation / acronym	Description
ADC	Application Delivery Controller
API	Application Programming Interface
AST	Application Security Testing
AV	Anti-Virus
CEO	Chief Executive Officer
CIRT	Cybersecurity Incident Response Team
CISO	Chief Information Security Officer
CISSP	Certified Information Systems Security Professional
CVE	Common Vulnerabilities and Exposures
DDoS	Distributed Denial of Service (network attack; also seen as DDSA)
DT	Deception Technology
Dx.y	Deliverable number y belonging to WP x
DoA	Document of Action
EC	European Commission
EPP	Endpoint Protection Platform
GRC	Governance, Risk Management and Compliance
HTTP	Hyper Text Transfer Protocol
HTTPS	HTTP Secure
IaaS	Infrastructure as a Service
IC	Innovation Committee
IDSISO	Intrusion Protection System International Organization for Standardisation
IDS	Intrusion Detection System
IoT	Internet of things
IP	Internet Protocol
ISFCISSP	Information Security Forum Certified Information Systems Security Professional
ISFAM	Information Security Focus Area Maturity
ISOISF	International Organization for Standardisation Information Security Forum
IT	Information Technology
JSON	JavaScript Object Notation

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	9 of 85	
Reference:	D3.9	Dissemination:	PU	
	Version:	1.0	Status:	Final

Abbreviation / acronym	Description
KPI	Key Performance Indicator
MISP	Malware Information Sharing Platform
MiTM	Man-in-the-Middle
MQTT	Message Queuing Telemetry Transport
OWASP	Open Web Application Security Project
PaaS	Platform as a Service
PCI-DSS	Payment Card Industry Data Security Standard
SaaS	Software as a Service
SIEM	Security Information and Event Management
SME	Small Medium Enterprise
SOC	Security Operations Centre
SSL	Secure Socket Layer
SUT	System Under Test
SW	Software
SWG	Secure Web Gateways
SWG	Secure Web Gateway
TaaS	Test-as-a-Service
UI	User interface
URL	Uniform Resource Locator
USG	Unified Service Gateway
UX	User Experience
VDI	Virtual desktop infrastructure
VM	virtual machine
VPN	Virtual Private Network
WAF	Web Application Firewall
WP	Work Package
XML	Extensible Mark-up Language
XMPP	Extensible Messaging and Presence Protocol

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	10 of 85	
Reference:	D3.9	Dissemination:	PU	
	Version:	1.0	Status:	Final

Executive Summary

The scope of this deliverable is to provide a detailed overview of the final prototype of the SMESEC security framework. The document contains a brief analysis of the final design views together with the design of the user interface, describes the various tools and standalone components integrated in the prototype, explains how these entities interact with each other and analyses the operation of the SMESEC security framework prototype and benefits it introduces. In addition, the document contains a detailed implementation, evaluation and testing analysis to clearly demonstrate the holistic approach of all participants toward delivering the specific framework.

This document builds upon D3.1 “SMESEC System Design” [1], D3.2 “SMESEC Unified Architecture – First Internal Release” [2] and D3.3 “SMESEC Framework User Manual” [3] and provides a description of changes and enhancements made to the SMESEC security framework. Treated as a living organism throughout the project, the SMESEC security framework was constantly under development to meet not only requirements gathered in previous deliverables and documented in D3.1 and D3.2 but address real-world issues as well. Such issues were identified through the Pilots or reported by skilled Third-Party personnel during the highly efficient evaluation phase carried out during the Open Call. All this effort led to various iterations, resulted in the specific architecture.

As already stated, we detail in this document the architecture of the internal SMESEC components. We present the core components that deliver orchestration functionalities: SMESEC Hub and SMESEC extensions. And, we present the architecture of the SMESEC interface. In addition, the enhanced user interface, designed with special attention to user-experience and based on iterative discussions with the use-case partners, is also presented. Finally, an overview of the overall SMESEC prototype functionality, integration, deployment and evaluation process is provided, either directly or via referenced to corresponding deliverables.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	11 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

1 Introduction

1.1 Purpose of the document

The scope of this deliverable is to provide a detailed overview of the final prototype of the SMESEC security framework. The document contains a brief analysis of the final design views, together with the design of the user interface. The document describes the various tools and standalone components integrated in the prototype, explains how these entities interact with each other and analyses the operation of the SMESEC security framework prototype and benefits it introduces. In addition, the document contains a detailed implementation, evaluation and testing analysis to clearly demonstrate the holistic approach of all participants toward delivering the specific framework.

1.2 Relation to other project work

This deliverable aggregate and extends all deliverables describing results related to WP3 and intends to provide a detailed analysis of the final architecture of the SMESEC security framework prototype together with the overall implementation methodology followed for delivering this perplexed task. In addition, several references are made to deliverables from WP4 and WP5, a clear indication that the overall consortium effort which lead to the delivery of SMESEC security framework prototype was equally spread on almost all technical WPs and subtasks.

1.3 Structure of the document

This document is structured in 6 major chapters:

Chapter 1 is the introduction which describes the main objectives of this deliverable, relationship to other deliverables, and the following sections.

Chapter 2 describes the SMESEC security framework design, analysing the Composition, Component, Interface and Deployment views along with the Communication Bus Security.

Chapter 3 provides an overview of the final user interface and analyses the intended user experience.

Chapter 4 describes the design of the SMESEC Framework Hub.

Chapter 5 describes the SMESEC security framework prototype implementation process, by giving an overview of the objectives, the integrated tools, the development environment and the frameworks which were used by the partners, the integration methodology, the underlying technical infrastructure which is necessary for the prototype to operate, some deployment and configuration guidelines, the external tool API and last but not least, the evaluation of the prototype as carried out via specialized tools.

Chapter 6 draws conclusions and summarizes the deliverable.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	12 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

2 SMESEC Framework design

The architecture of SMESEC has been slightly updated in this third year in order to cover some minor requirements and needs identified from: I) the technical work done by the consortium (either in the tools or the SMESEC Framework), II) feedback of the use case partners after performing the testing and validation of the platform and III) feedback of the SMEs and red team of the open call.

The next sections contain a description of the implementations and development that was done using the feedback focusing on the architectural updates and refinements. As the main architecture was presented in D3.2, only changes to those views are described.

2.1 Roles and Use Cases

The roles used in the SMESEC Framework are the same described in D3.3: administrator, security analysis, reporter, auditor and user. For giving a very short description of each one:

- Administrator: full access
- Security analyst: access to cybersecurity tools and reports, configuration and cybersecurity dashboard
- Reporter: access only to the reports of the tools and cybersecurity status of the organization
- Auditor: same as reporter but also configuration options
- User: regular employees. Can only access the training platform a personal dashboard

In order to facilitate the understanding of how they can access the SMESEC Framework several use case diagrams for each role were created.

The first diagram shows the functionalities of the administrator.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	13 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

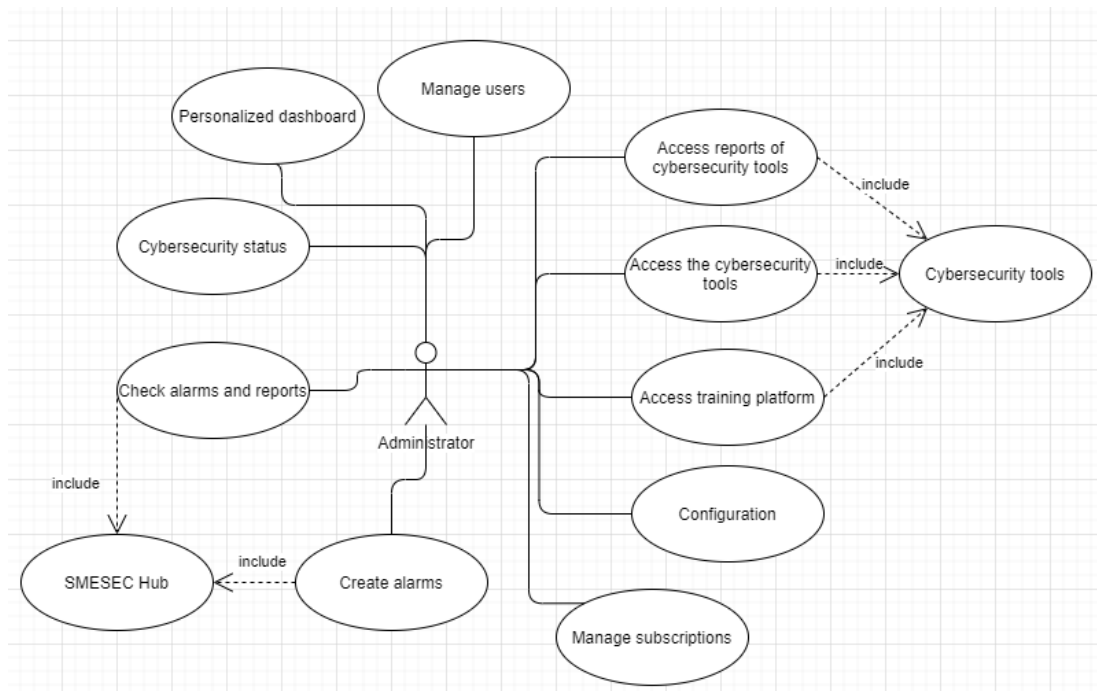


Figure 1: Use case diagram – administrator

As can be seen the administrator has access to all possible services of the SMESEC Framework. She can access the SMESEC Hub, the cybersecurity status of the organization, manage users, access to all functionalities of the cybersecurity tools and manage the subscriptions of the tools. This includes having more tools for the organization or cancelling the service of any of them.

The second role, the security analyst, is shown in in Figure 2.

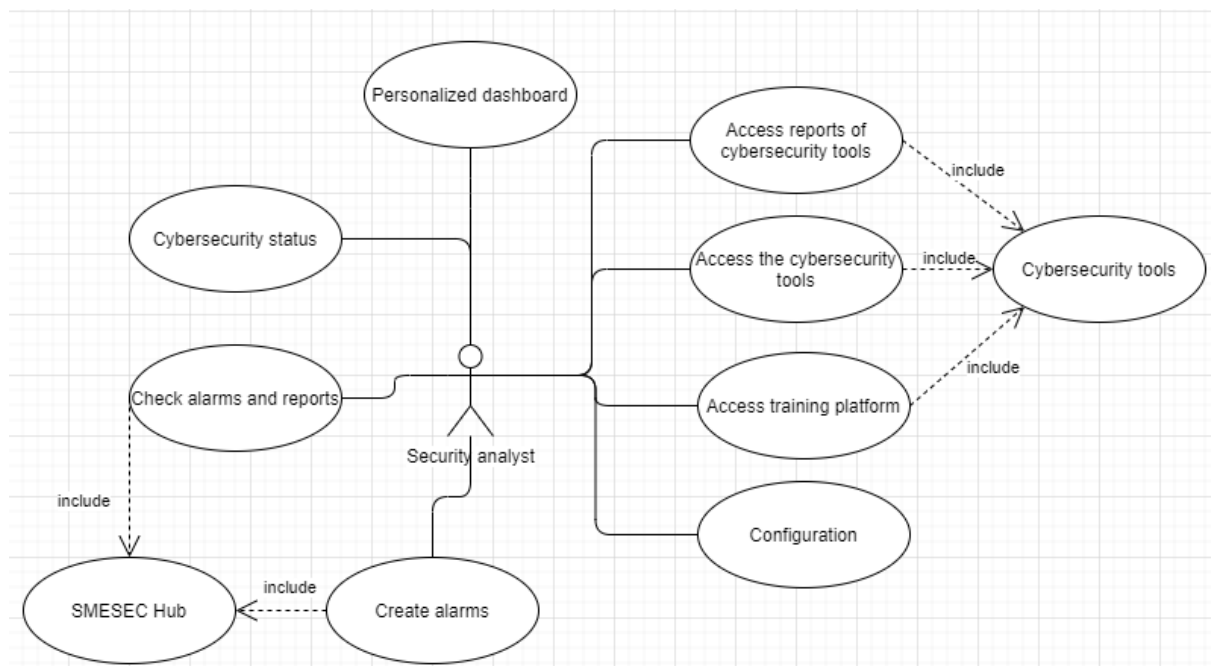


Figure 2: Use case diagram – security analyst

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	14 of 85	
Reference:	D3.9	Dissemination:	PU	
	Version:	1.0	Status:	Final

This role has access to all the cybersecurity functionalities of the SMESEC Framework but, as contrary to the administrator, cannot manage users or subscriptions of the tool. Therefore, she focuses more in the technical aspects of the organization and reporting of the status.

The next role, reporter, is shown in Figure 3.

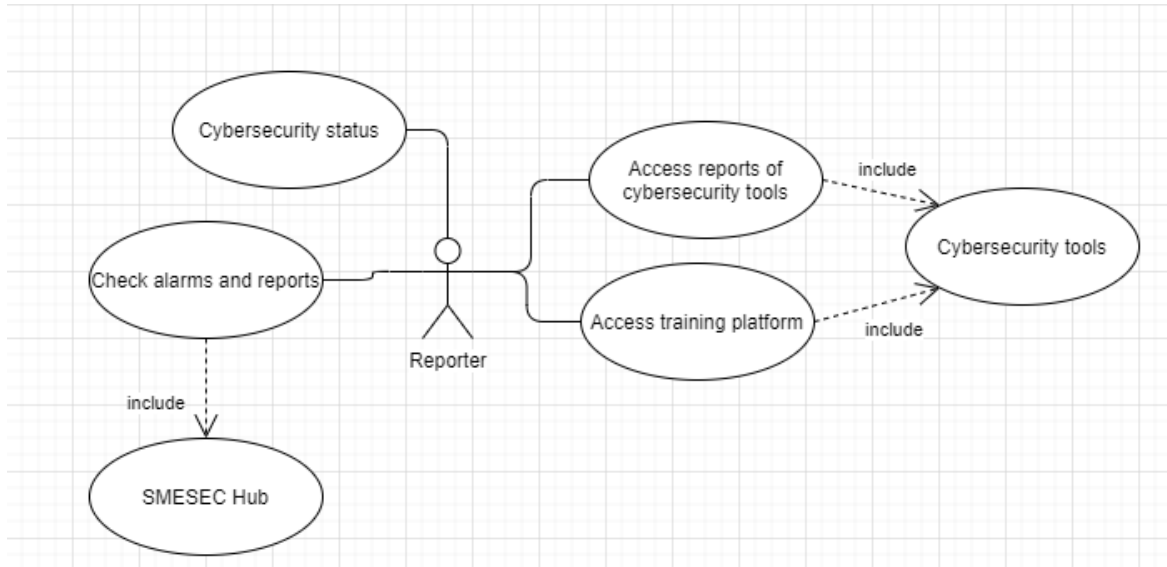


Figure 3: Use case diagram – reporter

The reporter has access to all generated report by the SMESEC Framework. This includes the cybersecurity status of the organization, the alarms and reports of the SMESEC Hub, the reports of the cybersecurity tools and the ones of the training platform.

The following role, the auditor, is similar to the reporter. Its use case diagram is shown in Figure 4.

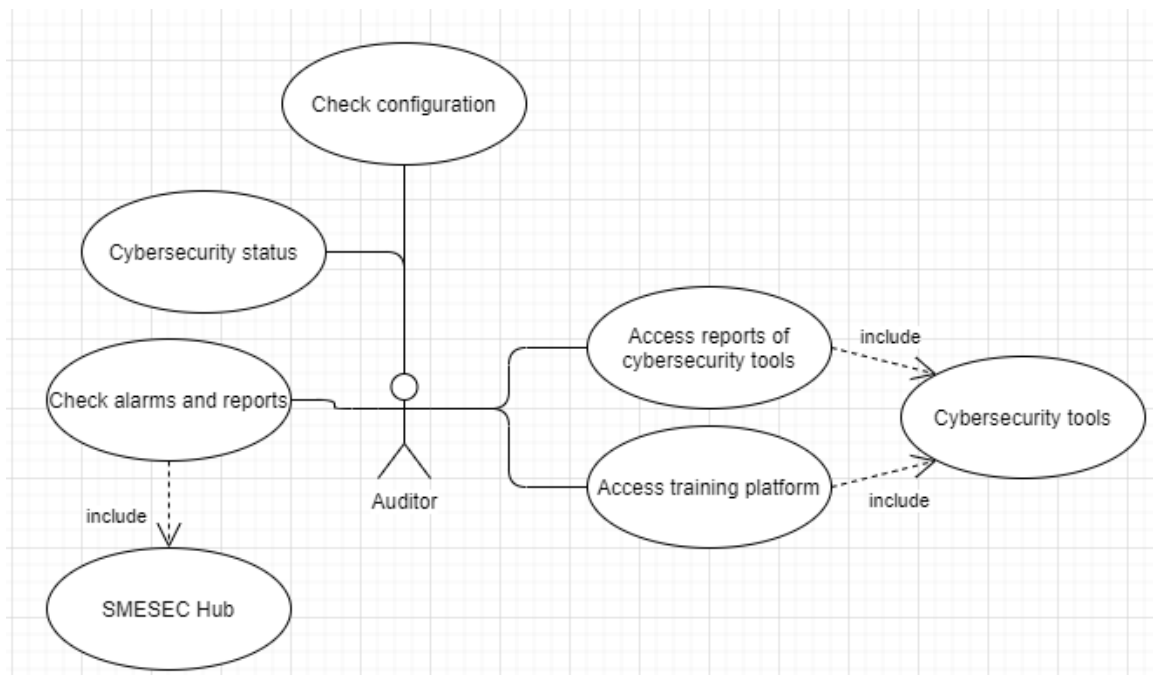


Figure 4: Use case diagram – auditor

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	15 of 85	
Reference:	D3.9	Dissemination:	PU	
	Version:	1.0	Status:	Final

As aforementioned, the auditor has the same functionalities as the reporter, but it also has access to check the configuration of the system. This is done in this way because the auditor should need to know how the system was configured in order to evaluate what happened in case of a cyberattack.

Finally, the end-user, identified as the role user, is presented in Figure 5.

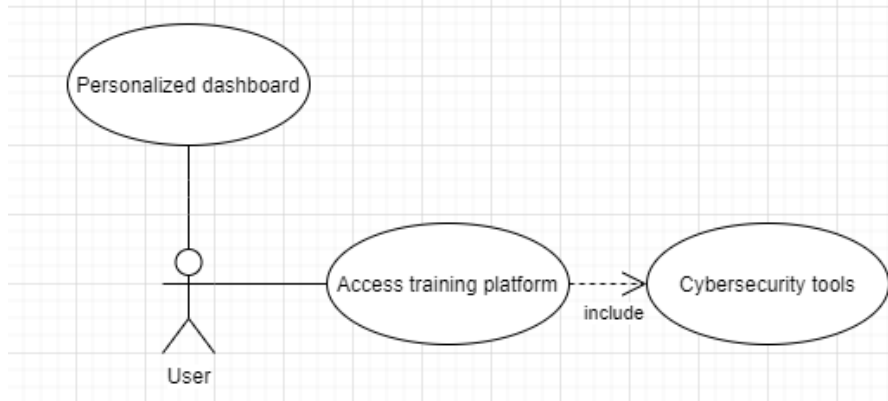


Figure 5: Use case diagram – user

The end-user of the organization does not need to access the cybersecurity status as it is responsibility only of the cybersecurity expert of the company. Therefore, the end-user can access her personalized dashboard in order to know about news that could interest her, information of the courses that are available or mandatory training and, of course, the training platform of SMESEC.

2.2 High-level architecture diagram

It is very important that SMESEC provides good services and cybersecurity functionalities. In order to do this, it was crucial to have a modular architecture and common communication infrastructure that could support this paradigm. This way, the architecture (high-level view) that was designed for SMESEC, together with interactions and external actors, and which has been improved in several iterations, is presented in Figure 6.

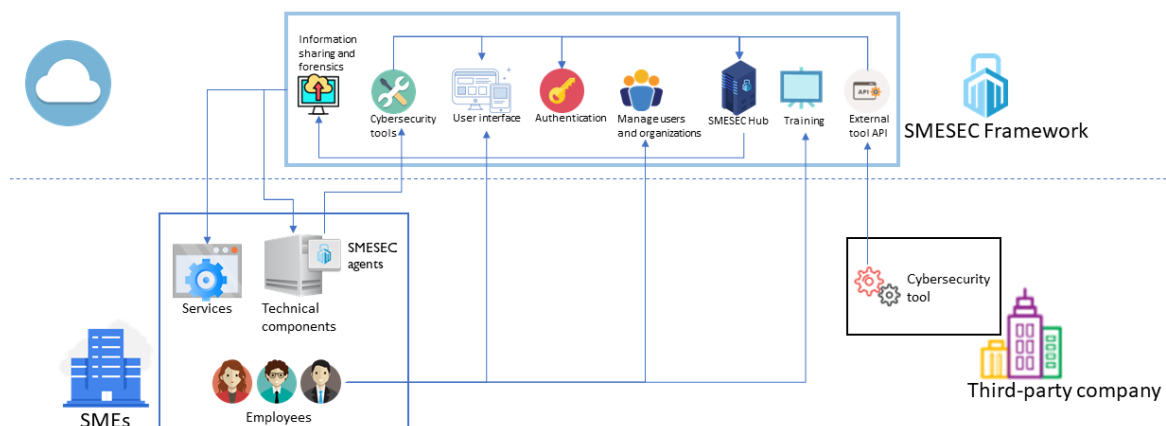


Figure 6: high-level architecture diagram

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	16 of 85
Reference:	D3.9	Dissemination:	PU
		Version:	1.0
		Status:	Final

As can be seen there are two external actors of SMESEC that interact with it. On the one hand the SME using the SMESEC Framework for having cybersecurity in its organization and, on the other hand, the third-party company that can integrate its solutions in framework in order to sell it.

The different services or components of the SMESEC Framework are very specific and also communicate between them. More in detail:

- **Cybersecurity tools:** the set of tools provided by the SMESEC Framework. It can be extended via the external tool API for third-party companies or internally by adding new tools (by the owner of the SMESEC Framework)
- **User interface:** the user interface of SMESEC that provides all the information of the organization. This is the one presented in the coming sections. Among others, it allows to see the cybersecurity status of the organization, the reports of the tools, the training platform, etc.
- **Authentication:** this component manages the authentication for the tools, external tools, users, organizations, etc. We implemented it using Keycloak as a basis.
- **Manage users and organizations:** this component, using the authentication component, allows for users and organizations to be registered and authenticated
- **SMESEC Hub:** The Hub was presented in the previous deliverable of the architecture. It serves as a “container” for data of the tools of SMESEC and external ones and can be used for creating tailored alarms in the system or to access information for creating new services or functionalities.
- **Training:** this component offers the training platform. It could also be extended with other platforms or tools that could help in training of employees. This includes also multimedia material.
- **External tool API:** this component allows for external tools to be integrated in the SMESEC Framework. This component was not foreseen at the beginning of the project, but we thought it could be useful for extending with more tools and also allowing other companies to sell their products.
- **Information sharing and forensics:** this component was mainly designed for extending the SMESEC Framework in order to provide data to an information sharing solution and for forensics. In SMESEC we allowed data to be obtained from the SMESEC Hub for this purpose although it was not implemented in the current implementation of the SMESEC Framework.

Regarding the SMEs, they interact with the SMESEC Framework in the following ways:

- Their services and technical components are protected by the cybersecurity tools in different ways. They can protect against intrusions, data thievery, illegal access, etc.
- The technical components have deployed the SMESEC agents, which compile information of events, protect them, etc. These have to be deployed in the system.
- The employees can use the SMESEC Framework for checking the cybersecurity status of the system, access the training platform, awareness solutions, manage their organization, etc. More information about the different actions each role can do is explained in the previous section.

Finally, for the external SME, it can use the SMESEC external tool API for integrating its solution and offer it to other organizations.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	17 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

2.3 Composition View

This section describes the final composition view of the SMESEC Framework. The composition view is depicted in Figure 7 and can be divided into several layers:

- The SMESEC infrastructure layer. The layer where all SMESEC infrastructure resides. This layer includes all the centralized functionalities like orchestration, authentication, configuration, and user interface.
- The SMESEC tools layers. The layer where all the tools of SMESEC reside. This layer includes all the tools provided by partners. These tools are external to the SMESEC infrastructure and are deployed at the partners' infrastructure. Also, the "external tool" resides in this layer.
- The SME infrastructure layer. The layer where all agents and endpoint security tools reside. This layer is the layer of tools integrated into the SME's infrastructure.

The architecture exposes the following user interfaces capabilities:

- Login. Supported by Keycloak [3] authorization and authentication mechanism. This is used to login into the SMESEC infrastructure and SMESEC tools layers. All components governed by Keycloak are denoted by a blue circle in the figure bellow.
- View attack chain alerts, recommendations and forensic reports. These are produced by the SMESEC Hub by orchestrating the various tools' results.
- Push notification to the user regarding alerts. These are produced by the SMESEC Hub.
- View alerts, view training, run testing and run patching. These are direct interfaces to SMESEC tool collection that are exposed to the user via the presentation interface of SMESEC.
- Edit SMESEC Hub predefined rules.
- Edit SMSEC Framework configuration, and part of the SMESEC tools' configuration (denoted by a green circle in the figure below).

The SMESEC Framework exposes the following interface categories: presentation interface and data interface. The presentation interface is used to propagate tool interfaces to the SMESEC interface, and the data interface is used for propagation of alerts and info from the tools and components into the SMESEC Hub. More details about the interface and communication module are to be found in the following sections.

The SMESEC infrastructure is composed of five main components:

- Presentation module responsible user interface interactions with underlying capabilities
- Keycloak module responsible for authorization and authentication
- Configuration module responsible for configuring the infrastructure and tools
- The SMESE-Hub responsible for orchestration of tools
- Communication interface responsible of communication to the SMESEC tools layer

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	18 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

Further details regarding the components of all three layers are to be found in the component view section.

SMESEC Framework

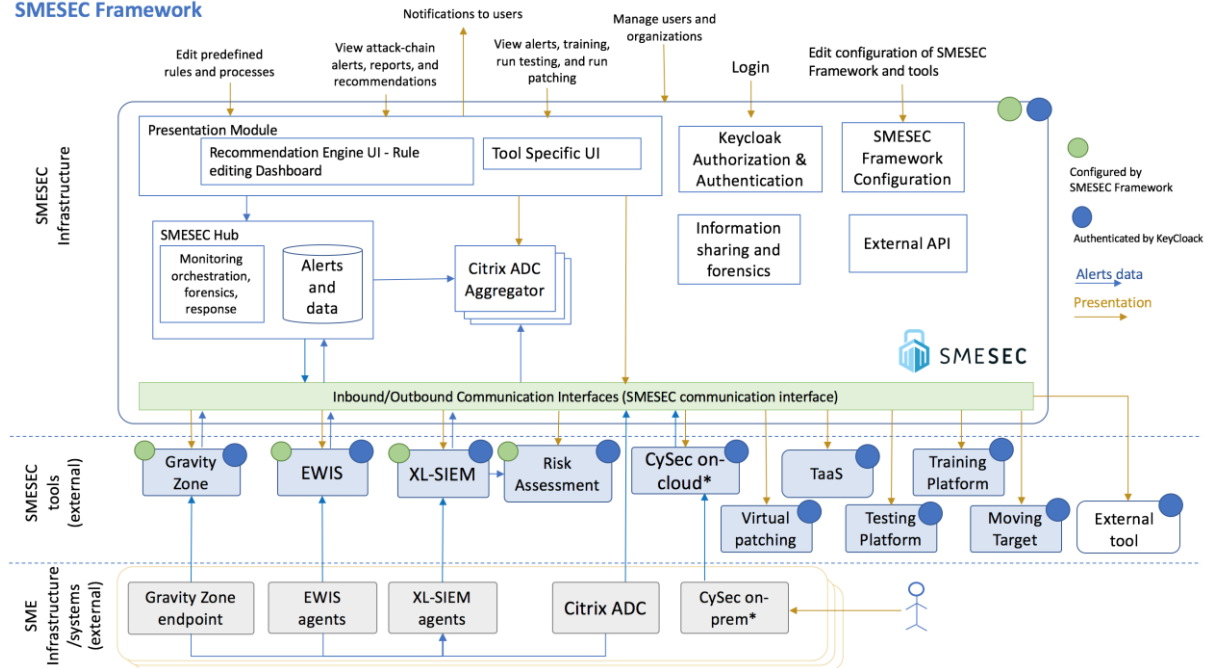


Figure 7: SMESEC composition view

In addition to the above composition view the SMESEC architecture enables two other setups in order to address specific SMESEC business needs which may require the Partial and Basic setups of the SMESEC Framework. The Partial and Basic setups are depicted in Figure 8 and Figure 9. These setups provide partial and limited capability of the overall SMESEC Framework by limiting the availability of SMESEC tools to a single user with access into this setup. The governance of setup per user is done using Keycloak authentication and is the responsibility of the tool owner.

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	19 of 85
Reference:	D3.9	Dissemination:	PU
		Version:	1.0
		Status:	Final

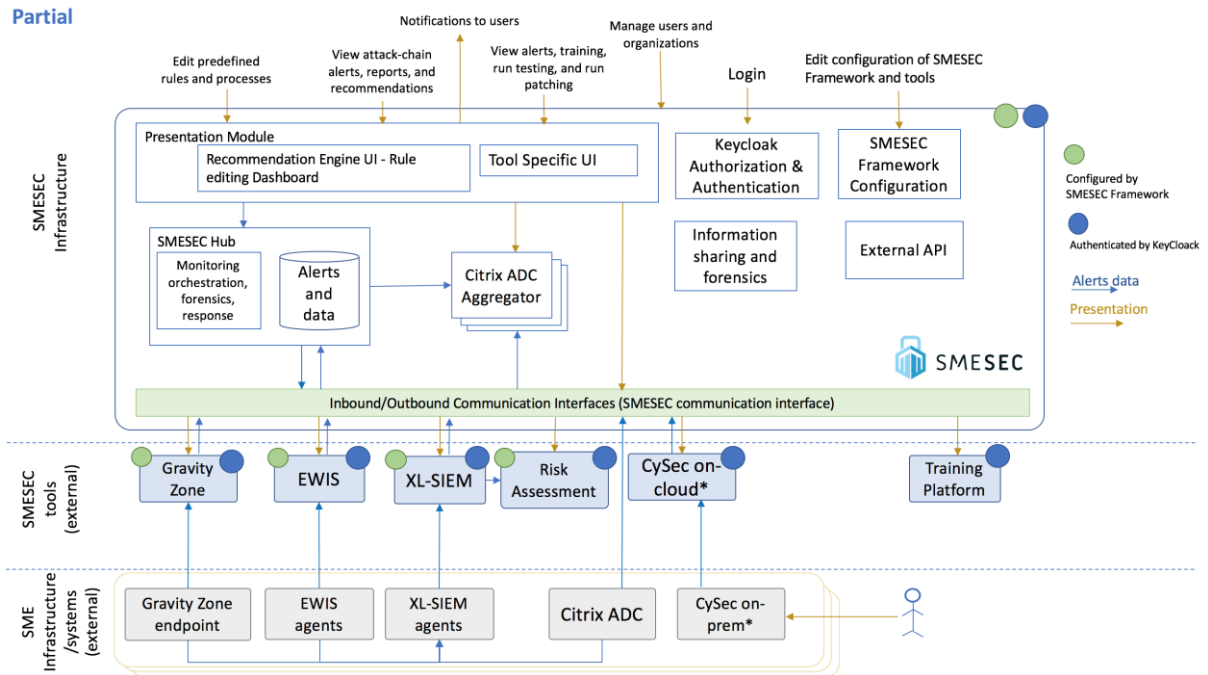


Figure 8: Partial SMESEC setup

The main difference between those two setups is the collection of tools deployed at the SME’s premise and the service available for the user. The concept behind the basic setup is that it provides basic security with monitoring, endpoint, training and orchestration. The concept behind the partial setup is to add on top of that the risk assessment, advanced network security, and security expertise assessment tools.

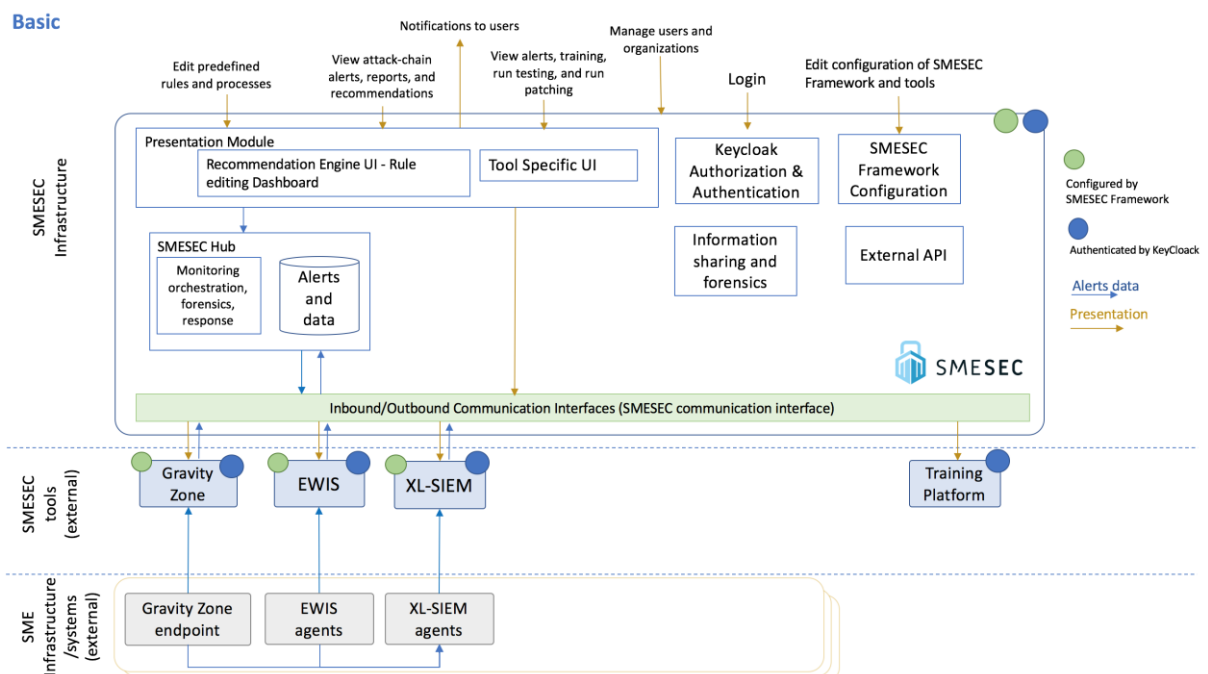


Figure 9: Basic SMESEC setup

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	20 of 85
Reference:	D3.9	Dissemination:	PU
		Version:	1.0
		Status:	Final

2.4 Component View

A detailed description of the SMESEC Framework components can be found in Table 1. For each component description, responsibility, input, and output are described. Further interface details are found in the interface view section. The components can be divided into six main categories:

- *Data collection* contains the tools, XL-SIEM agents, and EWIS agents. These components are responsible to collecting SME's data, which will be analyzed later. The SMESEC Framework supports monitoring of various sources of information, for example, network data is monitored by Citrix ADC.
- *Endpoint protection and offline tools* consists of the following tools, Citrix ADC, Gravity Zone endpoint, TaaS, Virtual patching, Testing Platform, and Moving target. These tools strive to strengthen both the infrastructures located on the SME's premises, and the security of the products developed by the SMEs.
- *Data analysis* category aggregates all data collected by the data collection and endpoint protection tools, analyses the data and prepares it for the orchestration and presentation modules. The following are aggregators in the SMESEC Framework: Citrix ADC Aggregator, Gravity Zone, XL-SIEM, EWIS.
- *Training and security assessment* tools aim to assess both the security level of the SME's infrastructures, and the awareness and knowledge in security of the employees. Furthermore, the SMESEC framework contains tools such as CySec that sets itself a target to raise awareness and give a proper security education to the SME's employees. This category contains the following tools, CySec on-prem, CySec on-Cloud, Training Platform, and Risk Assessment Engine.
- The *orchestration* contains the SMESEC Hub and extensions module. This module consists of various plugins that use hardcoded rules, alongside AI-generated patterns, to analyze all the data collected and produce alerts and recommendations.
- The *presentation module* is the interface of the whole SMESEC Framework with its users. It gives an intuitive and easy to use customizable UI that assists the user is governing over the whole framework.

The holistic table containing all components and presenting in full detail the description and responsibility of each component alongside with a high-level description of the components' input and output can be found below.

Table 1: Overview of SMESEC Framework Components together with their Input and Output information

Component	Description and responsibility	Input	Output
Citrix ADC	Intercepts network communication	Network traffic into SME's system	Information extracted from intercepted communication
Citrix ADC Aggregator	Aggregates information from	Information extracted from intercepted communication	Aggregated information into data visualization

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	21 of 85
Reference:	D3.9	Dissemination:	PU	Version:	1.0
				Status:	Final

	Citrix ADC and produces alerts		
XL-SIEM agents	Monitors on-disk log files	Log files from Citrix ADC , EWIS agents, and Gravity Zone endpoint	Information extracted from log files
XL-SIEM	Aggregates information from all XL-SIEM agents and produces alerts	Information extracted from log files from XL-SIEM agents	Alerts in either proprietary or MISP format to the HUB
Risk Assessment Engine	Correlates vulnerability posture with XL-SIEM alerts, and estimates risk possible cost in USD	XL-SIEM alerts, and vulnerability status from user	Prioritization of alerts based on vulnerability posture, and estimate security breach possible cost in USD
Gravity Zone endpoint	Malware detection and vulnerability management	Files on disk	Analysis result of malware detection sent to Gravity Zone and point and to XL-SIEM agents
Gravity Zone	Aggregates information from all Gravity Zone instances and produces alerts	Analysis result of malware detection from Gravity Zone endpoint	Aggregated alerts from all malware detection instances
EWIS agents	Honey-pot integrated into customer premises	Network traffic, files downloaded and every activity in the honeypot	Extracted information sent to XL-SIEM agents and XMPP commands sent to EWIS
EWIS	Aggregates information from all EWIS agents and produce alerts	XMPP commands from honeypot	Based on the monitored communications, syslog information of security events sent to XL-SIEM, and logs to EWIS backend database
CySec on-prem	Create recommendations for SMEs and train SMEs	User input (as answers to questions)	Logs, answers, accounts one-way replication (upon request only) to CySec-on-Cloud
CySec on-Cloud	Create recommendations for SMEs and train SMEs	User input (as answers to questions)	List of Recommendations to SMESEC HUB as MQTT-SMESEC-MISP messages to a statically configured server, and list of recommendations to user

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	22 of 85
Reference:	D3.9	Dissemination:	PU
	Version:	1.0	Status:
			Final

TaaS	Dynamic template-based testing platform	Information about connected user from Keycloak.	Test results on TaaS Front End.
Virtual patching	Create a virtual patch based on user data	Labelled samples of inputs to user application	Log scanner to be deployed at user premises
Testing Platform	Test customer's code for security vulnerabilities	Request to download the tool	Sends back the tool that was requested
Moving target	Compiler plugin	Request to download the tool	Sends back the tool that was requested
Training Platform	Provide training for SME's employees	Request to view the online training	Interactive training
HUB	Collect alerts from all online monitoring tools	Alerts in either proprietary or MISP format from XL-SIEM	Alerts sent to Citrix ADC Aggregator
SMESEC extension	Analyse alerts collection to detect possible attack-chains, provide initial forensic and response capabilities, and provide recommendations based on orchestration of alerts and CySec results	(1) alerts collected in HUB (2) Requests from the presentation module for rule editing	Attack-chain alerts, initial forensics & response, and recommendations
Presentation module	Presents results to user and receives user requests	User interaction/input	(1) present results to user (2) forward requests to system SMESEC extensions for rule editing and presentation (3) Requests to SMESEC communication interface for presentation of various tools (4) Requests Citrix ADC Aggregator for data through the available API (5) send notifications to user
Keycloak	Manages authorization and authentication of	Login request	Authentication and authorization to the following components: Gravity Zone, EWIS, XL-SIEM, Risk Assessment, CySec on-cloud,

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	23 of 85
Reference:	D3.9	Dissemination:	PU
	Version:	1.0	Status: Final

	SMESEC users		Virtual patching, TaaS, Testing Platform, Training Platform, Moving Target, External tool
Configuration	Update configuration of SMESEC framework	Configuration request	Configuration status for the following components: XL-SIEM, EWIS, Gravity Zone
Communication interface	Delegates communication between SMESEC tools and the SMESEC Framework	presentation requests and data transfer request	Presentation and data requests
External tool	TBD	TBD	TBD

2.5 Interface View

The interface view is used to specify the internal interfaces of the SMESEC Framework. The SMESEC Framework consists of the SMESEC infrastructure, SMESEC tools that run on various cloud providers, and endpoint tools that run on the SME’s premises. The diverse execution environments require a delicate approach to the design of communication between the various entities.

All inbound and outbound communication to and from the SMESEC Infrastructure goes through the SMESEC communication interface. It presents a standardized way of communication with the SMESEC Infrastructure and plays the role of the “gatekeeper” by providing a secure two-way gate to and from the infrastructure.

The communication between each endpoint tool on the SME’s premises with other tools provided by SMESEC partners is defined solely by the tool owners with the constraint of all communication to be secure to protect both the framework, and the potentially sensitive SME’s data.

The following table describes the interfaces between all components of the SMESEC Framework in detail. For each component, a list of interfacing components is provided, a description of what requests does this component initiate, a description of what requests does this component serve, and details whether this component provide a presentation interface, data interface, authentication interface, configuration interface, and encryption on-rest.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	24 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

Table 2: Interfaces of SMESEC Framework components

Component name	list of interfaces to other components	initiates	serve	presentation interface	data interface	on rest encryption	authentication interface	configuration interface
Citrix ADC	<ul style="list-style-type: none"> SMESEC communication interface 	pushes information to (1) XL-SIEM agent (2) SMESEC communication (3) Citrix ADC Aggregator through the Citrix NITRO API	none	no	yes	no	no	no
Citrix ADC Aggregator	<ul style="list-style-type: none"> SMESEC communication interface 	none	(1) consume Citrix-information routed by communication interface (2) provide data to HUB (3) provide data to presentation module	yes	yes	no	no	no
XL-SIEM agents	<ul style="list-style-type: none"> XL-SIEM (on cloud) 	push information to XL-SIEM	consume information from (1) Gravity Zone endpoint (2) EWIS agent (3) Citrix ADC	no	yes	no	no	no
XL-SIEM	<ul style="list-style-type: none"> SMESEC communication interface XL-SIEM agents 	push alerts to the (1) SMESEC communication module (2) Risk assessment engine	(1) consume information from XL-SIEM agents (2) serve presentation requests from the communication module	yes	yes	no	yes	yes

Risk Assessment Engine	<ul style="list-style-type: none"> • XL-SIEM • SMESEC communication interface 	none	(1) consume alerts from the XL-SIEM (2) serve presentation requests from the communication interface (3) consume survey answers from presentation module	yes	no	yes	yes	yes
Gravity Zone endpoint	<ul style="list-style-type: none"> • Gravity Zone (on cloud) • XL-SIEM agents 	pushes information to (1) XL-SIEM agent (2) Gravity Zone	none	no	yes	no	no	no
Gravity Zone	<ul style="list-style-type: none"> • SMESEC communication interface • Gravity Zone endpoint 	push alerts to the SMESEC communication module	serve presentation requests from the communication module	yes	yes	no	yes	yes
EWIS agents	<ul style="list-style-type: none"> • EWIS (on cloud) • XL-SIEM agents 	syslog information to XL-SIEM, Logs to EWIS backend databases	XMPP commands from EWIS backend.	no	yes	no	no	Yes
EWIS	<ul style="list-style-type: none"> • SMESEC communication interface • EWIS agents • XL-SIEM • CITRIX (Netscaler) 	push alerts to the SMESEC communication module	serve presentation requests from the communication module	yes	yes	no	yes	yes
CySec on-prem	<ul style="list-style-type: none"> • CySec on-cloud 	push status to CySec on-cloud	serve user requests via UI	yes	yes	no	no	no

CySec on-Cloud	<ul style="list-style-type: none"> SMESEC communication interface 	none	(1) serve presentation requests from the communication module (2) serve status update requests from the communication module	no	yes	no	yes	no
TaaS	<ul style="list-style-type: none"> SMESEC communication interface 	none	serve presentation requests from the communication module	yes	no	no	yes	no
Virtual patching	<ul style="list-style-type: none"> SMESEC communication interface 	none	serve presentation requests from the communication module	yes	no	no	yes	no
Testing Platform	<ul style="list-style-type: none"> SMESEC communication interface 	none	serve presentation requests from the communication module	yes	no	n/a	yes	no
moving target	<ul style="list-style-type: none"> SMESEC communication interface 	none	serve presentation requests from the communication module	yes	no	n/a	yes	no
Training Platform	<ul style="list-style-type: none"> SMESEC communication interface 	none	serve presentation requests from the communication module	yes	no	no	yes	no

HUB	<ul style="list-style-type: none"> • SMESEC extension • Citrix ADC Aggregator • SMESEC communication interface 	alert retrieval requests to Citrix ADC aggregator	(1) consume alerts from the communication module (2) serve alert fetch requests from SMESEC extensions module	no	yes	yes	no	no
SMESEC extension	<ul style="list-style-type: none"> • HUB • SMESEC communication interface • Presentation module 	status fetch request to the communication module and draws alerts from the HUB	presentation and configuration request from the presentation module	no	yes	yes	no	no
presentation module	<ul style="list-style-type: none"> • SMESEC extension • Citrix ADC Aggregator • SMESEC communication interface 	initiates notifications to users	serve user requests of: (1) rule and process editing (2) presentation of alerts, reports, recommendations (3) presentation of tool specific UI	yes	yes	n/a	no	no
Keycloak	<ul style="list-style-type: none"> • Gravity Zone • EWIS • XL-SIEM, Risk Assessment • CySec on-cloud • Virtual patching • TaaS • Testing Platform • Training Platform • Moving Target • External tool 	none	serve (1) user login requests (2) module authentication and authorization requests	yes	yes	yes	yes	no
configuration	<ul style="list-style-type: none"> • XL-SIEM • EWIS • Gravity Zone 	initiate configuration requests	serve user configuration requests via UI	yes	yes	yes	yes	n/a

communication interface	<ul style="list-style-type: none"> Gravity Zone (on cloud) EWIS (on cloud) XL-SIEM (on cloud) CySec (on cloud) AngeEye, TaaS Training Platform Moving Target External Tool Citrix ADC SMESEC extension HUB Presentation Module Citrix ADC Aggregator Keycloak authorization and authentication 	(1) presentation requests to Gravity Zone, EWIS, XL-SIEM, Risk Assessment, CySec on-cloud, Virtual patching, TaaS, Testing Platform, Training Platform, Moving target and External tools (2) status fetch requests to CySec on cloud	consume alerts and data from Gravity Zone, EWIS, XL-SIEM, Citrix ADC	yes	yes	no	no	no
External tool	<ul style="list-style-type: none"> SMESEC communication interface 	None	presentation requests from the communication module	TBD	TBD	TBD	yes	no

2.6 Deployment View

The deployment of SMESEC Framework can be categorized into three categories:

- Deployment of SMESEC infrastructure
- Deployment of SMESEC tools
- Deployment of agents and endpoint tools

The deployment of agents and end-point-security is necessary for collecting information from the SME systems; thus, these tools are always deployed in the SME's premise. In addition to those, a CySec tool deployment into the SME's is optional for SME's who are concerned about privacy.

The SMESEC tools layer includes two categories: (1) online tools that aggregate information from agents and end-point-security tools (2) offline tools that are not dependent on the agents and end-point security tools. The tools in this layer are deployed on tool-providers' premises or on the cloud. One exception for this is the Citrix ADC Aggregator that was deployed inside the SMESEC-infrastructure during the development of the prototype and is planned to become an independent deployment in the future.

The SMESEC infrastructure is includes all the components responsible for the tools' collection and orchestration. This is deployed at ATOS premises and it supports multi-tenancy of SME's.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	29 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

Follows a table describing the deployment details of the SMESEC Framework:

Table 3: Deployment of SMESEC components

Component	Deployment	Multi-tenancy
Citrix ADC	SME's infrastructure	instance per SME
Citrix ADC Aggregator	SMESEC infrastructure	instance per SME
XL-SIEM agents	SME's infrastructure	multiple instances per SME
XL-SIEM	ATOS infrastructure	yes
Risk Assessment Engine	ATOS infrastructure	yes
Gravity Zone endpoint	SME's infrastructure	multiple instances per SME
Gravity Zone	BD infrastructure	yes
EWIS agents	SME's infrastructure	multiple instances per SME
EWIS	FORTH infrastructure	yes
CySec on-prem	SME's infrastructure	instance per SME
CySec on-Cloud	FHNW	yes
TaaS	EGM infrastructure	yes
Virtual patching	IBM Cloud	yes
Testing Platform	IBM Cloud	instance per SME
moving target	IBM Cloud	instance per SME
Training Platform	UoP infrastructure	yes
HUB	SMESEC infrastructure	yes
SMESEC extension	SMESEC infrastructure	yes
presentation module	SMESEC infrastructure	yes
Keycloak	SMESEC infrastructure	yes
configuration	SMESEC infrastructure	yes
communication interface	SMESEC infrastructure	yes
External tool	TBD	TBD

2.7 Communication Bus Security

All SMESEC tools connected to the communication-bus must apply mutual (two way) Keycloak authentication. All communication between the communication bus and the SMESEC tools, regardless of the underlying protocol, must be encrypted using TLS1.2 or above.

The Security responsibilities of the SMESEC communication between SMESEC infrastructure and SMESEC tools are distributed among components as follows:

- Tool security is the tool provider's responsibility.
- It is the communication bus responsibility to apply network security.
- HUB-security: It is the HUB responsibility to validate their input against possible attacks.

The bus must support multi-tenancy and load balancing.

The bus must apply network security measures:

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	30 of 85
Reference:	D3.9	Dissemination:	PU	Version:	1.0
				Status:	Final

- Install and configure a firewall for hardening
- Input validation for security purposes (i.e. DoS attack detection, discovery and response of potential malicious activity)

All security events reported by bus security (example firewall), must be logged to a central logging service, and saved for 90 days.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	31 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

3 SMESEC Framework User Experience

Usability is a crucial requirement of the SMESEC Framework. Being aimed at organisations and users ranging from no expertise on cybersecurity to high, it is essential that the usage and understanding of the framework are as high as possible. The SMESEC Framework design offers a unified interface for all tools included in the SMESEC Framework.

To design the proper user experience (UX) and identify the target personas of the SMESEC Framework, we have collected feedback from the SMESEC use case SMEs and open call SMEs trials. We used that feedback to evolve the personas, functions, and user interface, including navigation, of the SMESEC Framework in comparison to the earlier Framework description in D3.3 [27].

This section describes the evolved user personas, gives an overview of the provided UI functions, specifies the navigation, and describes the details of the user interface views. The specification refers to D3.3 Section 3 *SMESEC Framework User Experience* and describes additions or modifications to the previously specified user experience design.

3.1 Personas

The user interface has been designed for use by specific personae in the SME. The feedback obtained from the trial SMEs has shown that the roles defined in D3.3 [27] were appropriate but needed refinement to clarify the roles' background and specific characteristics and extension to cover new, advanced roles that were necessary for managing cybersecurity in the SMEs.

We here present a new three-layered role model. The new community layer characterises tiers of SMEs that differ in characteristics and needs for SMESEC support. The previously introduced SME roles layer characterises the roles within the SME for managing cybersecurity of the SME. The new SMESEC framework role model layer used for configuring access to SMESEC tools and views.

The new **community layer** characterises types of SMEs, as shown in the following figure.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	32 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

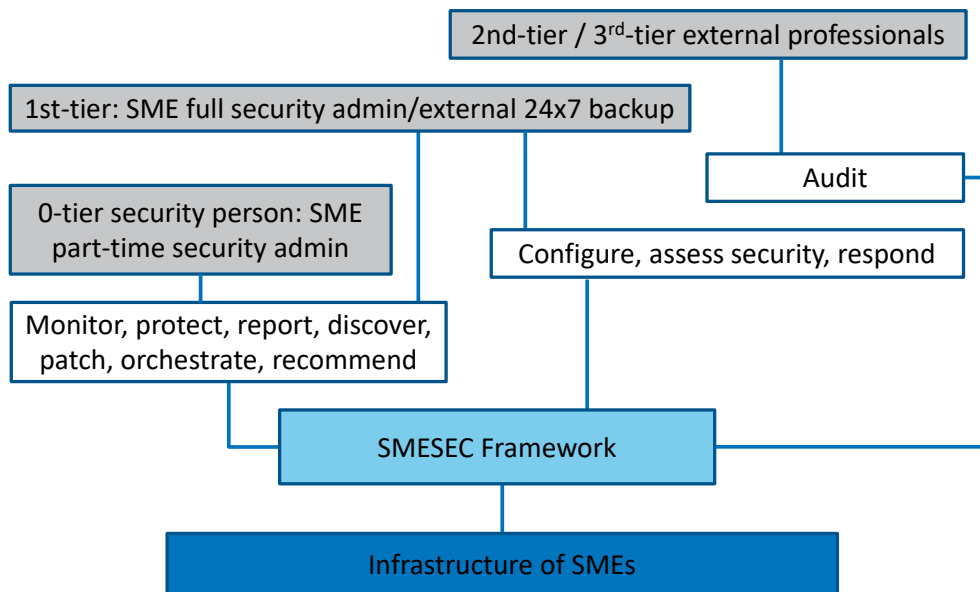


Figure 10: Community layer: extended roles supported by the SMESEC Framework.

As shown in the figure, the feedback from the trial SMEs indicated that three tiers of SMEs needed to be supported. The 0-tier matches SMEs with a business where ICT is not dominant and where cybersecurity can hardly be allocated to a full-time security administrator. The 1st tier matches SMEs with a business that depends on ICT to the extent that a full security administrator is warranted. Still, such an SME can decide to outsource the provision of ICT infrastructure to a third-party. The 2nd tier represents SMEs with an ICT-centric business, some of which providing ICT services to tier 0 and tier 1 SMEs. The 3rd tier represents SMEs with a cybersecurity-centric business, hence generate revenue with the help of cybersecurity offered by SMESEC. The following table characterises each tier and gives an overview of the support of SMESEC.

Tier	Characterisation	Primary SMESEC Functions
Tier 0	SMEs with a business where ICT is not dominant and does not allow allocating a full-time security administrator. As an example, such an SME may use Office Applications, CRM, or ERP software to manage their business.	Monitor and protect ICT and data, report events and incidents, discover vulnerabilities, patch software, orchestrate protection, and recommend improvements of cybersecurity capabilities.
Tier 1	SMEs with significant dependency on ICT, warranting the allocation of a full-time security administrator.	Tier 0 functions, and configure SMESEC, assess the SME’s cybersecurity capabilities, and respond to threats.
Tier 2	SMEs with ICT-centric business, offering ICT-based solutions and services to other companies	Tier 1 functions and security audit.
Tier 3	SMEs with cybersecurity-centric business, generating revenue with the help of cybersecurity	Tier 1 functions and security audit.

The **SME roles layer** characterises the cybersecurity role model within the SME that has been introduced in D3.3 already. The following table gives an overview of the roles while referring back to the role model described in D3.3.

Role	Characterisation	Interaction with SMESEC
CISO (Nicolas)	This primary role is responsible for cybersecurity in the SME. This role is the main user of SMESEC. Relevant for tier 0 upwards. The CISO is the security administrator or, in the case of higher-level tiers, may be assisted by a team of security administrators.	Main user of SMESEC. He receives guidance for the personal learning of cybersecurity and how to address cybersecurity with the SMESEC framework. Minimal effort to obtain and maintain overview and awareness of cybersecurity in the SME, manage controls, and report about the security of the SME.
CEO (Philippe)	Chief Executive Officer (CEO) leading the strategy and operations of the SME and being legally responsible for its overall welfare. He understands the importance of cybersecurity but is too busy to manage it sustainably. Relevant for tier 0 upwards.	The CEO, assisted by the CISO, receives information about how secure his company is and what should be done to improve the security. The CEO decides about the use of SMESEC and the various tools that are offered as components.
Employee (Claudia)	Employee of the SME and expected to be aware of cyber threats and adhere to safe practices that help to protect the SME from these cyber threats. She wants to do her work well and expects that others are helping her. Relevant for tier 0 upwards.	The employees are provided access to training offered by the SMESEC tool Securityaware.me and polls generated from CYSEC. The CISO coordinates the interaction of the employees with SMESEC online and offline.
Careless Employee (Julien)	Employee of the SME with a careless attitude and potentially malicious intentions that might hurt the security of the SME. Relevant for tier 0 upwards.	Nicolas works with Julien the same way he works with Claudia. In addition, Nicolas activates and configures monitoring tools of the SMESEC framework, such as GravityZone, the EWIS honeypot, and ADC to detect insider attacks and uses tools like IBM AntiROP and TaaS to prevent potential backdoors in the SME's products and services.
Cybersecurity expert (Martin)	Cybersecurity expert or consultant offering personalised help and advice for SMEs. The expert's business is cybersecurity, and he brings in-depth practical experience as a CISO and member of cybersecurity incident response teams (CIRT).	Martin offers CISOs specialised advice beyond what the SMESEC framework provides and support for responding to cybersecurity incidents. To assist Martin, a CISO shares the company profile, maturity information, event logs collected and with the SMESEC framework.
Cybersecurity innovator (Jose)	Cybersecurity reference person and community manager interacting with stakeholders and advancing	Some CISOs understand that creating industry-wide awareness and advancing cybersecurity technology depends on their

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	34 of 85	
Reference:	D3.9	Dissemination:	PU	
	Version:	1.0	Status:	Final

	cybersecurity for SME in Europe.	company. For that reason, such a CISO opts into sharing anonymous data about events and capability improvements the cybersecurity innovator.
External cybersecurity auditor (Christos)	External cybersecurity auditor responsible for verifying that the SME is compliant with regulations and standards.	Christos collaborates with the CISO for compliance auditing upon request. The CISO uses the SMESEC framework to implement the controls, practices, and training that Christos suggests.

The new **SMESEC framework role model layer** is used for configuring access to tools and views within the SMESEC framework. These roles are supported by the Keycloak component used for managing identities and authorisations in the SMESEC framework. The following table gives an overview and describes how each framework role is recommended to map to the SME roles.

Role	Rights	Recommended Mapping
SMESEC Framework Administrator	The SMESEC Framework administrator is maintaining the SMESEC framework and offers support for its use. It is a service provider role towards the SME. The primary rights are associated with licensing, deployment, and configuration management of the framework use.	The SMESEC framework administrator should be mapped to the cybersecurity innovator role, including his selected assistant supporters and developers.
SME Administrator	The SME administrator has full access to the SMESEC functions, user management and rights definition for the users within his SME or the SMEs he is supporting.	The SME administrator maps to the CISO. A CISO may work together with the cybersecurity expert for configuring SMESEC and the external cybersecurity auditor to inspect configuration and collected data. However, the CISO should not give away the SME administrator rights.
Employee of the SME	The employee role has limited access to tools and data as configured by the SME administrator. Note that also the SME Administrator may be an Employee.	The employee role maps to the CEO, employee, and careless employee. The SME administrator may also decide to give employee-level access to the cybersecurity experts and external cybersecurity auditors of his choice.

3.2 Functions

In deliverables D3.2 [2] and D3.3 [27], a SMESEC Framework user interface (UI) was proposed that primarily consists of a launcher and static information about the SMESEC tools that can be accessed through the launcher and a one-stop dashboard for the Chief Information Security Officer (CISO) of the SME.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	35 of 85
Reference:	D3.9	Dissemination:	PU	Version:	1.0
				Status:	Final

To account for the feedback obtained in the SME trials, the dashboard has been extended with additional views as shown by the following table:

Framework Role	Views
SME Administrator	SMESEC Dashboard, SMESEC-at-my-Company, Tool View, Hub Configuration, SMESEC Tools Dashboard, Tools Configuration
Employee	My Status

The dashboard was developed in collaboration with the use case SMEs members of the SMESEC consortium. To co-design helped to elicit latent tacit needs that would not have been discovered otherwise. It also allowed taking advantage of the SMESEC tool provider's expertise and testing of ideas of how an effective workplace can be designed that is usable and useful for the SME Administrator and employees.

The SMESEC framework UI offers a comprehensive overview of indicators and events that reflect the status of the SME, provides recommendations for actions that may be useful in the SME's situation and provides access to the SMESEC tools.

The tables offer traceability with the list of functions defined in D3.2 and D3.3 through consistent use of identifiers and motivate the modifications, respectively extensions of these earlier defined functions.

3.2.1 Overarching User Interface Design Decisions

The idea of the one-stop dashboard for the SME CISO implied restructuring of the user interface. An enhanced header and footer have been designed and an evolved navigation paradigm defined.

The following figure shows the frame, consisting of a header and footer, used for embedding the views offered to the SME Administrator and the Employees.



Figure 11: Frame for the SMEEC Framework Views

The following figure shows the ribbon menu that allows accessing each SMESEC tool with quick links. The Ribbon Menu can be opened by clicking on the Accordeon icon at the top-left of the Header.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	36 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

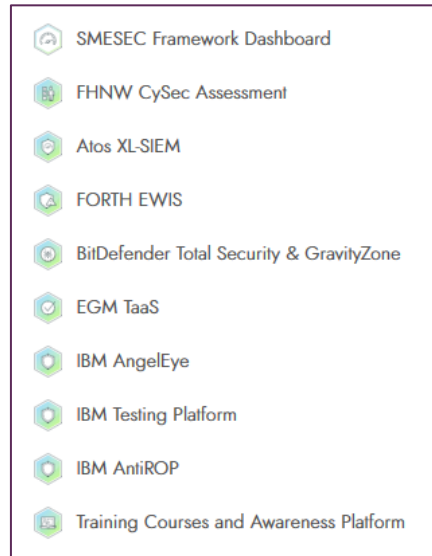


Figure 12: SMESEC Tools Ribbon Menu.

The following table shows the common UI elements of the SMESEC framework user interface.

Functions	Targeted Benefits	Implementation
FWUI-UC01: Single Sign-on (unchanged)	Allow access to all SMESEC-protected information and tools with one effort.	KeyRock-based authentication and authorisation.
FWUI-P01.1: Quick Links (unchanged)	Support exploration of tools. Support visual inspection and correlation of tools' settings and outputs.	Integrated tool display with header indicating chosen tool, tool display, and accordion with compact tool launcher.
FWUI-P01.4: Header Bar (unchanged)	The human end-user knows he is using the SMESEC framework. The human end-user can navigate across the views: a personal view with favourite indicators, the security status overview of the SME, the status of the SMESEC tools, the introduction and selection of the SMESEC tools, a selection of the framework plugins, and the security configuration of the SME.	HTML always shown on top of the screen.
FWUI-P01.5: Footer (unchanged)	The human end-user knows that the SMSEC framework is delivered by trustworthy parties.	HTML with logo and disclaimers at the bottom of the page.

The following subsections describe the views of the SMESEC framework than can be integrated into the overarching user interface.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	37 of 85
Reference:	D3.9	Dissemination:	PU	Version:	1.0
				Status:	Final

3.2.2 View: SMESEC Dashboard

The design of the SMESEC dashboard has not changed. The trials have confirmed that it is important that the SME can answer the awareness question “*how secure am I?*” and the guidance question “*how can I improve my security?*”

The following figure shows the SMESEC Dashboard:



Figure 13: SMESEC Dashboard.

The SMESEC dashboard answers the awareness question with an indicator showing the SME’s level of security, an overview of the most recent attacks, and the history of latest security events. The Security Information and Event Manager XL-SIEM collects attack and event history, the EWIS tool offers detection of recent attacks, and the GravityZone and ADC tools offer detection of security events.

The following figure shows how the awareness question is answered.

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	38 of 85
Reference:	D3.9	Dissemination:	PU
		Version:	1.0
		Status:	Final

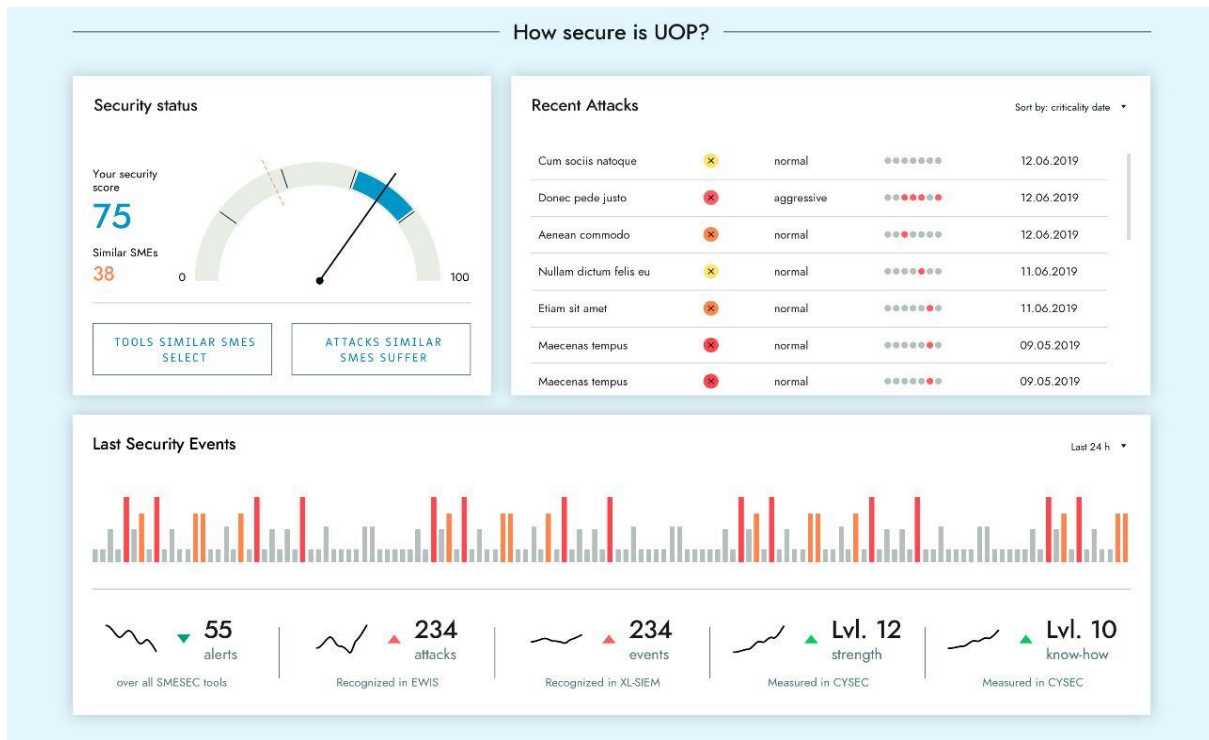


Figure 14: Answering of awareness question.

An alerts section has been added at the top of the SMESEC Dashboard, allowing to highlight information about recent critical events.

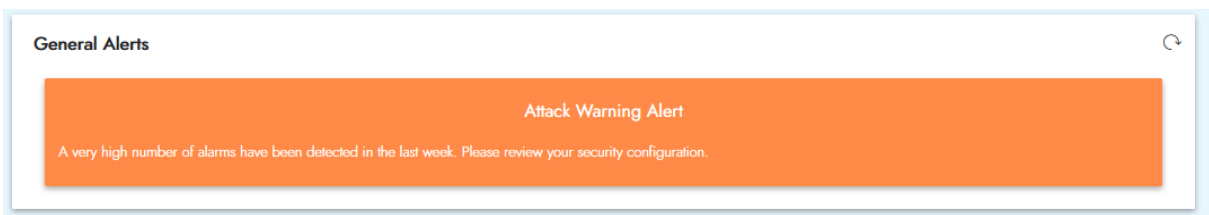


Figure 15: Alert widget.

The SMESEC dashboard also answers the guidance question with the following widgets. An overview of the SME's most critical cybersecurity capabilities is offered with the focus areas progress widget. Recommendations for tools to be installed and training to be provided to employees are offered with recommender widget showing the currently relevant top recommendations. The maturity indicators widget shows bottom-line indicators about the company's protection strengths, cybersecurity know-how, and cybersecurity fitness, which is calculated based on how consistently the company is working on cybersecurity with the help of the SMESEC framework. The CYSEC tool is used to generate recommendations and calculate the indicators.

The following figure shows how the guidance question is answered.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	39 of 85
Reference:	D3.9	Dissemination:	PU	Version:	1.0
				Status:	Final

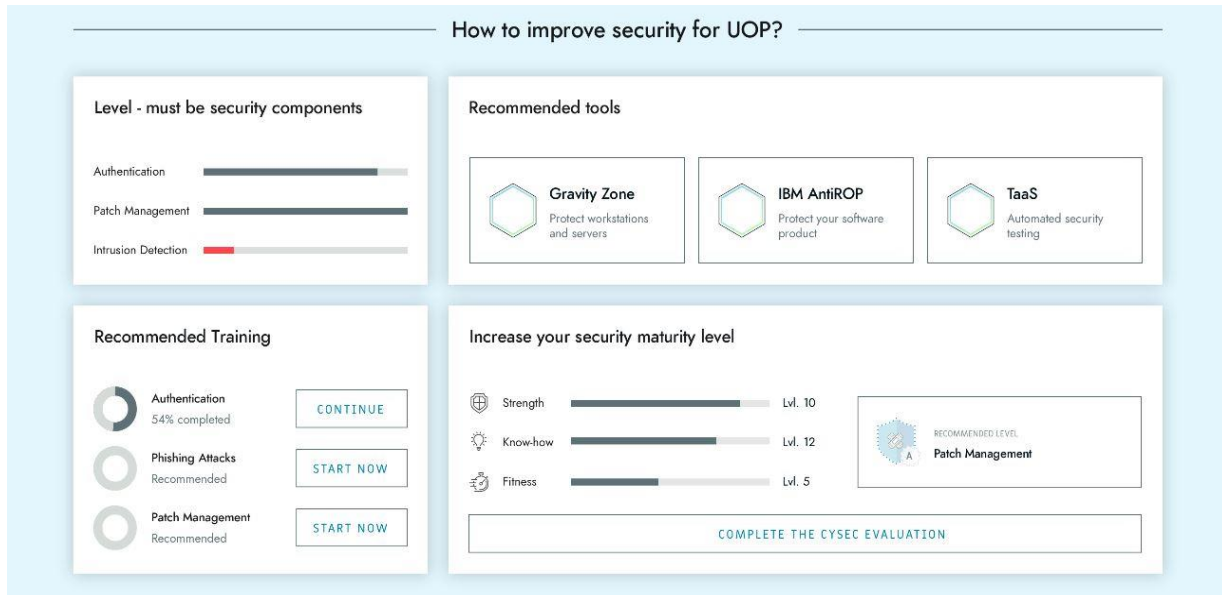


Figure 16: Answering of the guidance question.

The following table shows

Functions	Targeted Benefits	Implementation
FWUI-UC03-V3: Display cybersecurity KPI and alarms for the SME (changed to separate the individual views clearly)	Awareness of current threats and protection status, and guidance of the CISO with little expertise with recommended actions. The SMESEC tools report the following information: real-time security-related events, discovered vulnerabilities, the SME's security maturity, alerts, and trends. Flexibility for the consortium to add and remove SMESEC tools	Mashup of UI controls rendered by the various SMESEC tools.
FWUI-P03.1: Dashboard (unchanged)	The human end-user is aware of the threat exposure and protection of the SME and know recommended actions for improving the SME's security.	Integration of plugin-rendered HTML.
FWUI-P03.2: Tool-Launching Recommendations (unchanged)	The human end-user knows recommended actions and can launch Securityaware.me, respectively CYSEC with the right context to implement the action.	Integration of tool-rendered HTML and links to the matching tool context.
FWUI-P03.3: Alert Display (unchanged)	The human end-user is aware of alerts.	Integration of plugin-rendered HTML and link to the matching tool for resolving the alert.

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	40 of 85	
Reference:	D3.9	Dissemination:	PU	
	Version:	1.0	Status:	Final

3.2.3 View: SMESEC-At-My-Company

The View SMESEC-At-My-Company offers a detailed overview of important data processed by each of the SMESEC tools that are activated by the company. This view offers one integrated overview produced by all of the SMESEC tools in use by the SME on one page. An interaction with a tool leads to the tool's user interface.

The following table shows the overview of tools that are part of the SMESEC framework and the primary functions they provide.

SMESEC Tool	Primary Function
SMESEC Hub	One-stop-shop for SMESEC cybersecurity tools
XL-SIEM	Security information and event management
Gravity Zone End-Point	End-point security
Citrix ADC, EWIS	Network security
CYSEC Coaches	Recommendations and guidance for CISO
Securityaware.me	Employee training
EGM TaaS, IBM Testing Platform	Software and device testing
IBM Angel Eye	Customised virtual patching
IBM Anti-ROP	Moving target capability for software development

The following figure shows the design of the view.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	41 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

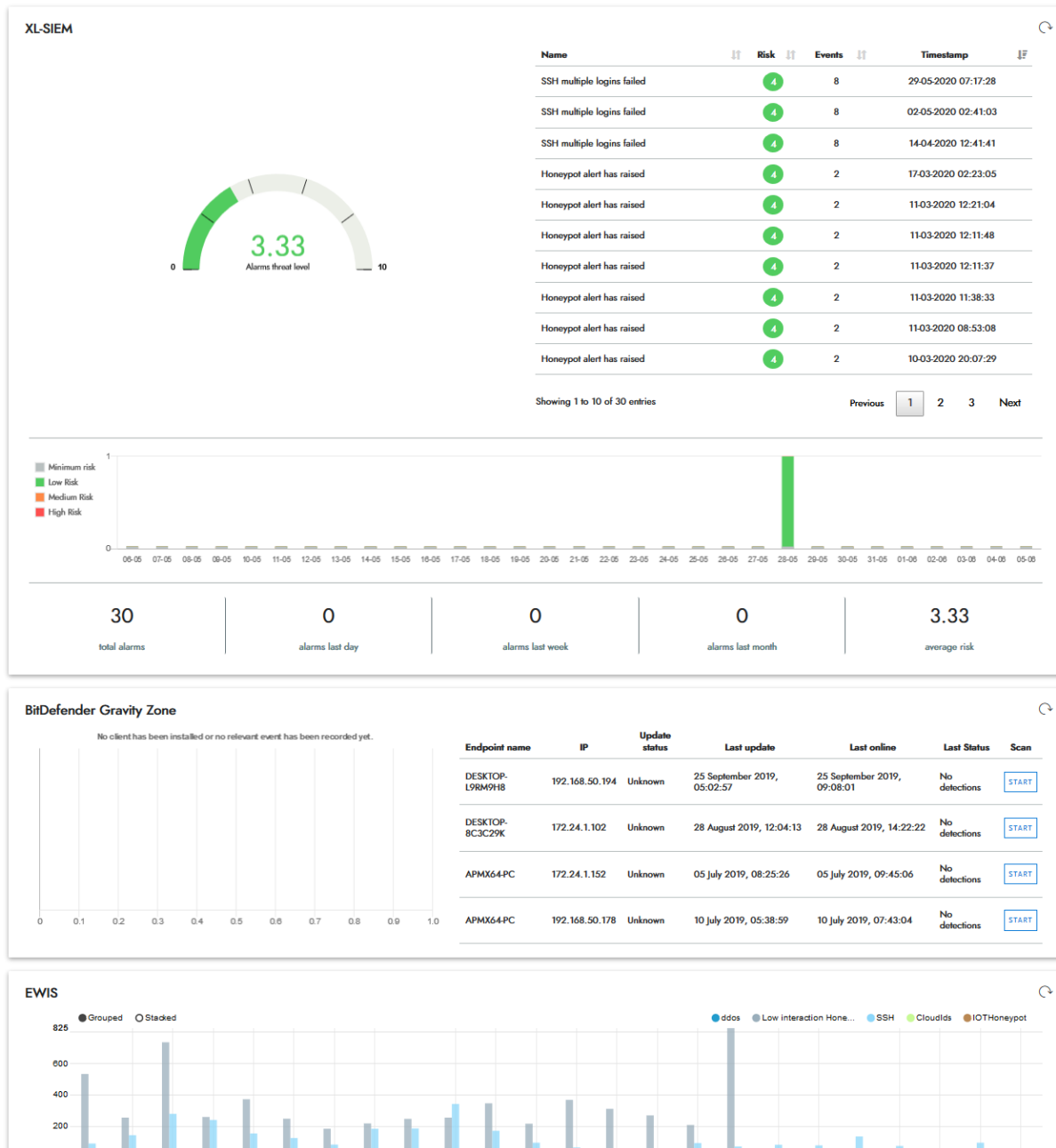


Figure 17: The SMESEC-At-My-Company View

The following table shows the elements of the SMESEC-At-My-Company view.

Function	Targeted Benefits	Implementation
FWUI-P04.1: Dashboard (unchanged)	The human end-user is aware of cybersecurity status according to the activated SMESEC tools.	Integration of plugin-rendered HTML.
FWUI-P03.2: Tool-Launching Recommendations (unchanged)	The human end-user knows recommended actions and can launch any SMESEC tool through the respective widget used for information display.	Integration of tool-rendered HTML and links to the matching tool.

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	42 of 85
Reference:	D3.9	Dissemination:	PU
	Version:	1.0	Status: Final

FWUI-P03.3: Alert Display (removed from this view)	The human end-user is able to activate a tool to be considered in the Dashboards or to deactivate it.	Removed from this view and integrated into the SMESEC Dashboard view.
----------------------------------------------------	-------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------

3.2.4 View: Tool View

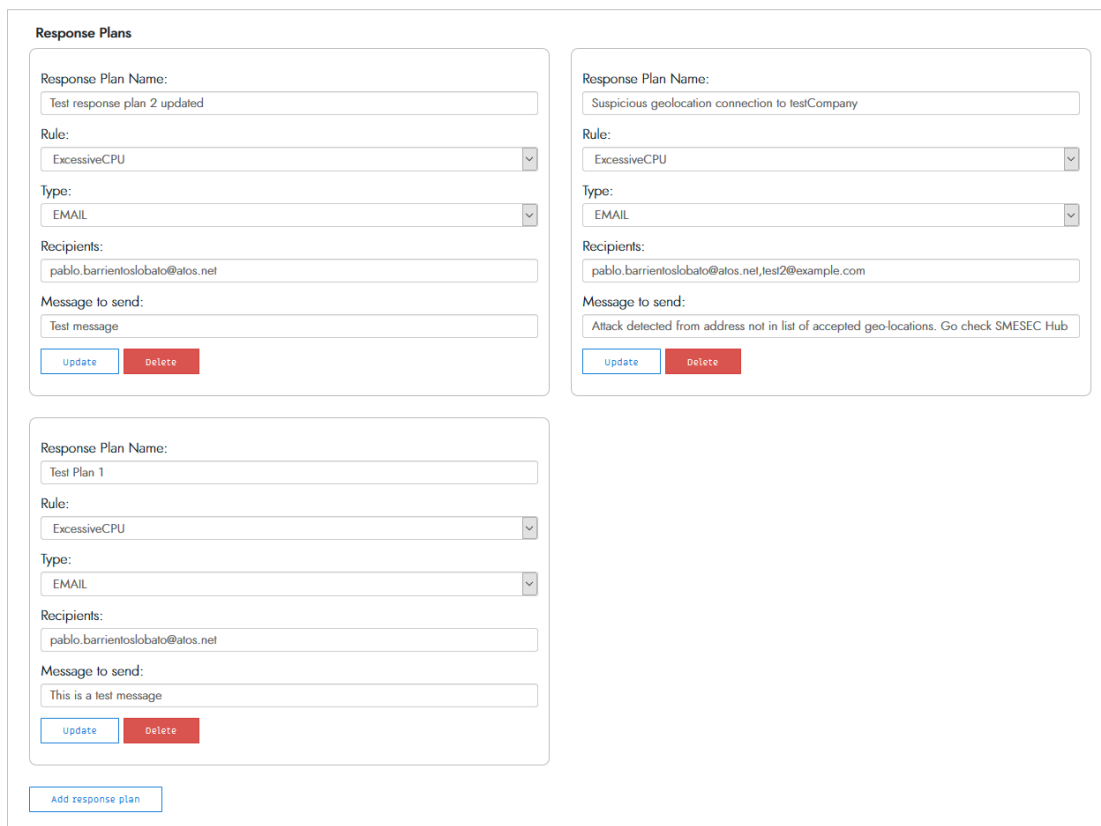
The Tool View FWUI-P02 is an iFrame used for integrating the tool’s user interface. The function catalogue is updated as specified in the following table.

Table 4: Element Updates of the Tool View UI.

Function	Targeted Benefits	Implementation
FWUI-P02.1: Tool UI (unchanged)	The human end-user uses the launched SMESEC tool without distracting cluttering.	iFrame integration of tool front-end.

3.2.5 New View: Hub Configuration View

The SMESEC Hub Configuration view provides the SME Administrator with the ability to specify rules that modify the behaviour of the SMESEC Framework. These rules allow the SMESEC framework to automatically respond to events, such as honeypot detection alerts, e.g. by sending a mail or an SMS to a designated person. The following figure shows the user interface.



The screenshot displays the 'Response Plans' configuration interface. It features three distinct configuration cards, each with the following fields:

- Response Plan Name:** A text input field.
- Rule:** A dropdown menu with 'ExcessiveCPU' selected.
- Type:** A dropdown menu with 'EMAIL' selected.
- Recipients:** A text input field containing an email address.
- Message to send:** A text input field for the alert message.

Each card includes 'Update' and 'Delete' buttons. At the bottom of the interface, there is an 'Add response plan' button.

Figure 18: Hub Configuration View.

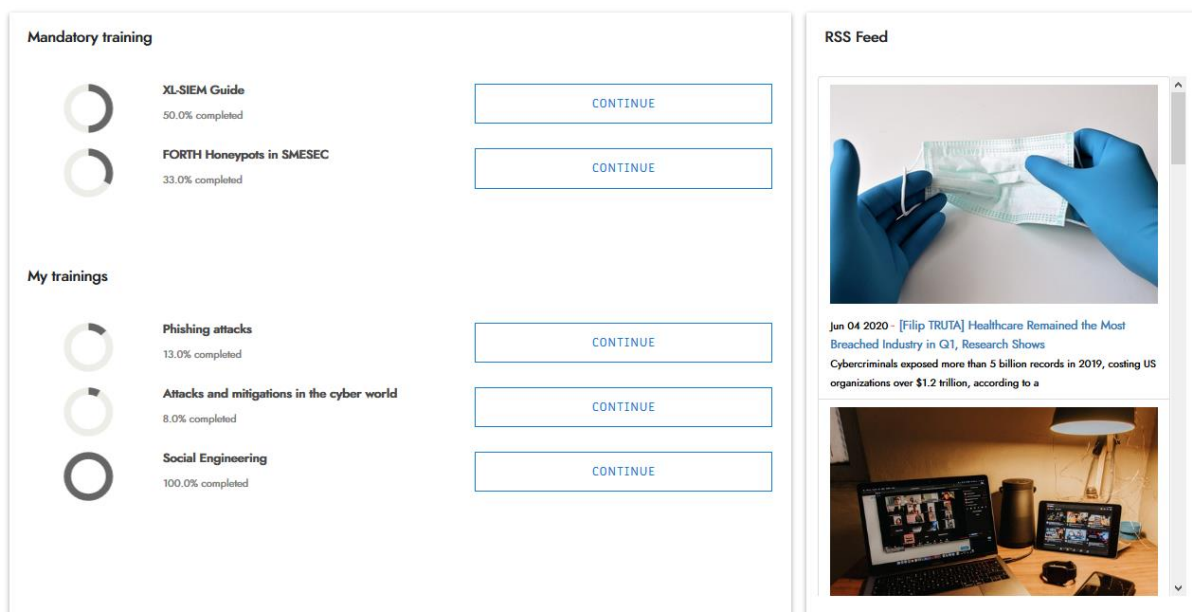
Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	43 of 85
Reference:	D3.9	Dissemination:	PU	Version:	1.0
				Status:	Final

The following table shows the elements of the SMESEC Hub Configuration view.

Function	Targeted Benefits	Implementation
FWUI-P11.1: Configure Rule (new)	Automated notifications can be configured, allowing a fast reaction to important events captured with the SMESEC tools.	Rule configuration.

3.2.6 New View: My Status View

The SMESEC framework was extended with a view for the employee that is logged in. The view provides the employee with the ability to see his training status and access the training that is either required or interesting to learn for the employee. The view also provides the employee with the ability to see a feed of important news around cybersecurity.



The following table shows the elements of the SMESEC Tools Dashboard view.

Function	Targeted Benefits	Implementation
FWUI-P10.1: Training Status (new)	The employee is aware of his or her training needs and progress.	HTML with cross-page links and dynamically generated status indicator.
FWUI-P10.2: News Feed (new)	The employee is aware of recent threats and insights about cybersecurity for SMEs. The employee is encouraged to return to the view as the news may be perceived as interesting and relevant to personal work and behaviour.	RSS feed embedded in an iFrame.


Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	44 of 85
Reference:	D3.9	Dissemination:	PU
	Version:	1.0	Status: Final

3.2.7 View: SMESEC Tools Dashboard

The SMESEC Tools Dashboard provides the SME Administrator with the ability to obtain, download, and activate tools offered in the SMESEC Framework.

This configuration of tooling for the SME is supported with a simple interface. It shows the installation status for available tools and absence of tools that are not available to the SME, e.g. due to license restrictions. The interface further allows to download a tool on premise, access the tool provided as a service, or access a tutorial for the tool. It also provides access to license management where, e.g., a license can be upgraded to access more tools.

The following figure shows the SMESEC Tools Dashboard.




XL-SIEM
Type: Monitoring
Status: Running

XL-SIEM is a tool able to receive monitoring events coming from a variety of sources (through some agent software) and after identifying and analysing the data, by possibly correlating with other sources, it can react and try to mitigate the effects of a cyber-attack.

Download XL-SIEM agent

Go to course!




EWIS
Type: Monitoring
Status: Running

EWIS (Early Warning Intrusion Detection System) is a honeypot-based solution where the so-called sensors VMs can be deployed in an infrastructure and attract potential attacks by capturing the malicious user's actions and transferring that information to a central database in real-time. The system consists of two main parts: the honeypot VMs and a central control panel that is used for management and visualization purposes. EWIS can detect DDoS attacks and provide the appropriate alerts, with the accuracy of the produced results being proportional to the amount of the dark IP address space monitored and the amount of honeypot VM instances deployed.

Download EWIS

Go to course!




GravityZone
Type: End Point Protection
Status: Running

GravityZone provides high quality safety to business against evolving threats by protecting the endpoints and providing meaningful insights on their use. Using advanced behaviour-based technologies, Bitdefender detected 99% of unknown threats in independent trials run by reputed independent testing organization like AV Comparatives. Bitdefender also has two additional anti-ransomware defence layers - a blacklist of 2.8 million samples and rising, and a vaccine that can immunize devices against the encryption process.

Download agent

Go to course!




ADC
Type: Monitoring
Status: Not Available

ADC is an application delivery controller that performs application-specific traffic analysis to intelligently distribute, optimize, and secure Layer 4-Layer 7 (L4 - L7) network traffic for web applications, provides flexible delivery services for traditional, containerized and microservice applications and delivers enhanced cybersecurity features. Its feature set consists of switching features, security and protection features, and server-farm optimization features. Under the auspices of SMESEC, only part of the overall Citrix ADC functionality will be utilized, mostly related to enforcing company security/compliance policies, obtaining user behavior insights, and eliminating encryption-related blind spots potentially are exploited by attackers to evade security controls.

Go to course!

Upgrade SMESEC now!



Risk Assessment Engine
Type: Risk Management
Status: Disabled

Risk Assessment Engine is a tool that runs in near real-time a set of risk assessment algorithms and define a set of actions to be enforced. It also allows for qualitative and quantitative analysis of results and allows managers to understand the long-term cyber-risk exposure and help them plan for their cybersecurity strategy.

Go to course!

Activate it now!

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	45 of 85
Reference:	D3.9	Dissemination:	PU	Version:	1.0
				Status:	Final

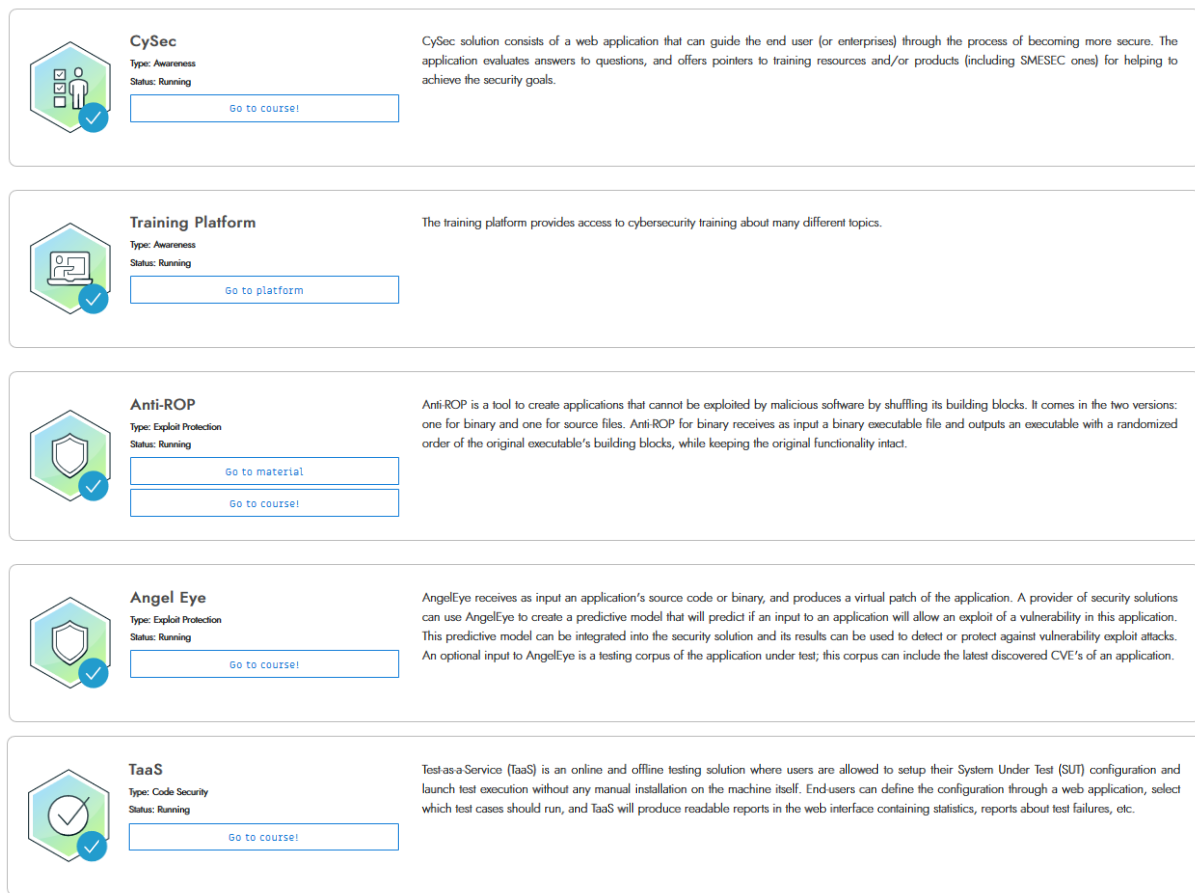


Figure 19: SMESEC Tools Dashboard.

The following table shows the elements of the SMESEC Tools Dashboard view.

Function	Targeted Benefits	Implementation
FWUI-P02.1: Tool Launcher (changed)	The human end-user gets introduced into the topic of cybersecurity through the presentation of SMESEC tools. The human end-user can activate a tool with a full understanding of the tool's scope.	HTML with cross-page links.
FWUI-P01.6: Activation (changed)	The human end-user is able to activate a tool to be considered in the Dashboards or to deactivate it.	Status information and buttons to change status.

SMESEC also provides the framework administrator with the ability to integrate third-party tools into the SMESEC framework, hence allows the evolution of the SMESEC offering. A third-party API has been defined for that purpose. Integrated tools will also appear in the SMESEC Tools Dashboard.

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	46 of 85
Reference:	D3.9	Dissemination:	PU
	Version:	1.0	Status: Final

3.2.8 New View: Tools Configuration View

The Tools Configuration view provides the SME Administrator with the ability to specify settings for the SMESEC tools. The following figure shows the definition of CITRIX ADC, CYSEC, and GravityZone settings.

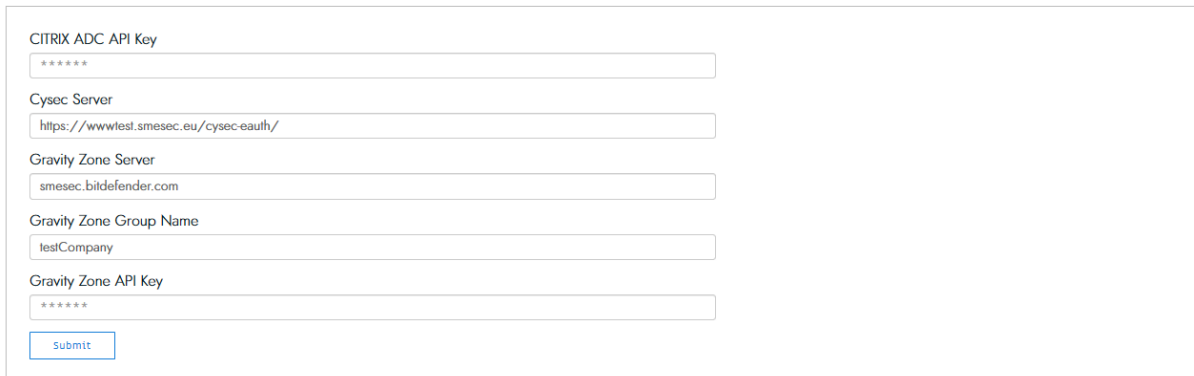


Figure 20: SMESEC Tools Configuration view.

The following table shows the elements of the Tools Configuration view.

Function	Targeted Benefits	Implementation
FWUI-P12.1: Tool Settings	Provides the SME Administrator with the ability to adapt the SMESEC tools to the SME.	HTML interface for settings variables

3.3 Navigation

To account for the complexity of cybersecurity monitoring and management, the SMESEC framework UI offers a simple navigation approach based on two paradigms: a) menu bar, b) rich information displays, and c) integrated information display and launcher. The menu bar provides the user with the ability to switch among views. The rich information displays offer information for answering important end-user questions. The launcher allows running a tool from the specific context provided by the widget in the view's mashup.

In comparison to D3.3, the navigation paradigm remained unchanged. Refined is the menu bar that has benefitted from the extended number of views: Hub Configuration View, My Status View, and Tool Configuration View.

The following figure shows the screens and navigation pathways.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	47 of 85
Reference:	D3.9	Dissemination:	PU	Version:	1.0
				Status:	Final

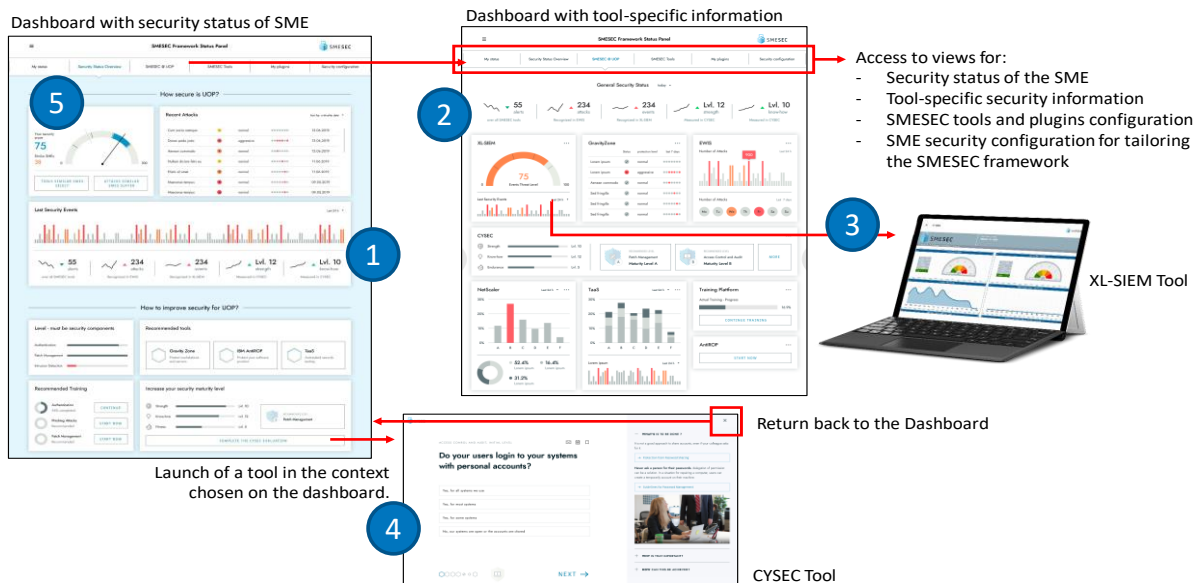


Figure 21: Same navigation paradigm as described in D3.3.

The starting point of the user journey, after login, is the same as previously: the SMESEC dashboard. It provides the user with the ability to get a one-page overview of the cybersecurity status of their SME (1). That status includes the answers to the questions of how secure is your SME and how to improve the security of your SME. The first question is answered with a score of the SME's security, an overview of recent attacks, and the timeline of recent security events detected by the SMESEC framework. The second question is answered with the current status of self-assessment, capability improvement, and training provision and recommendations of next steps.

The user is offered the choice through a top-level menu bar to switch to the SMESEC tools and drill down into the detailed statuses reported by the SMESEC tools and to see how each of the tools has contributed to the security status assessment (SMESEC-At-My-Company view, 2). The view (2) also allows inspecting the status of tools that did not report their measurements or results for the aggregate overview of (1). For example, TaaS is a tool used to manage cybersecurity as a quality aspect for product or service development and does not require the immediate reaction, e.g. of a cybersecurity incident response team (CIRT), to resolve observed problems.

The view (2) can also be used to launch any of the SMESEC tools. Shown as an illustrative example in a previous figure is the launch of the XL-SIEM tool (3). The view (1) can also be used for launching tools but is restricted to specific training actions with Securityaware.me or self-assessment and capability improvement actions with the cybersecurity coach CYSEC (4). Every of the SMESEC tools runs standalone from the end-user's perspective and can be opened and closed in parallel to the SMESEC dashboard. Tools that represent plugins into other frameworks, such as the IBM AntiROP that is used as a compiler plugin, offer download instructions and how to use guidelines.

As a final option, the user can use the menu bar for accessing the remaining views (5).

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	48 of 85	
Reference:	D3.9	Dissemination:	PU	
	Version:	1.0	Status:	Final

4 Design of SMESEC Framework Hub

The functionality to aggregate and correlate alarms coming from diverse cybersecurity solution is implicit in the conception of the SMESEC framework: an easy-to-understand overview of the status of the monitored infrastructure, as well as the recommendation of mitigation plans for non-cybersecurity experts in the event a particular threat realizes, is a feature that makes the difference compared to the existing solutions in the market. Thus, the SMESEC framework aims to offer this additional functionality, which goes beyond the mere sum of the individual parts. To meet this demand, the development of the so-called SMESEC framework Hub was envisaged from the very beginning of the project.

From a design point of view, the Hub results in a data aggregator module that gathers in a single point the inputs from the deployed cybersecurity solutions and triggers configurable alarms and recommendations for end-users. This middle point within the SMESEC framework architecture checks and validates the quality of the data by crossing various sources and ultimately, improves the experience of the user, independently from the previous technical and cybersecurity knowledge. The Hub performance materializes in a user-friendly output displayed in the front end (*see section 5.2.12*).

4.1 System Architecture

This section overviews how the different technologies, modules and connectors that integrate the SMESEC framework Hub enable the abovementioned functionalities.

As said before, the system collects inputs from different sources, processes them and produce a refined output to be displayed at the framework front-end. A high-level and simplified overview of the architecture is shown in Figure 23. Before getting on the details of each constituent block, it is important to mention that their design was made in such a way that they are highly adaptable (not only the core module, as explained in *Section 4.1.3*, but the entire system). This allows ingesting heterogeneous inputs, provide diverse functionalities and result in tailor-made outputs. Nevertheless, and before commissioning, the SMESEC Framework Hub requires ad-hoc consulting efforts to adapt the technology to the specific requirements of the end-user.

4.1.1 Global perspective

The operation of the SMESEC Hub system is composed of four sequential steps: (i) the data collection from the cybersecurity solutions which are deployed at the monitored infrastructure, (ii) the format and transfer of these data that will enable their orchestration within the SMESEC Hub, (iii) the core module where all the data are processed, sent and eventually stored, and finally (iv) the consumption of the outputs to be done from the front-end and an API. This latter step allows real-time analysis and the study of historical data for audit purposes.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	49 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

GLOBAL VIEW: Information goes from the use cases the data will be captured by the solution providers, and added to the system.

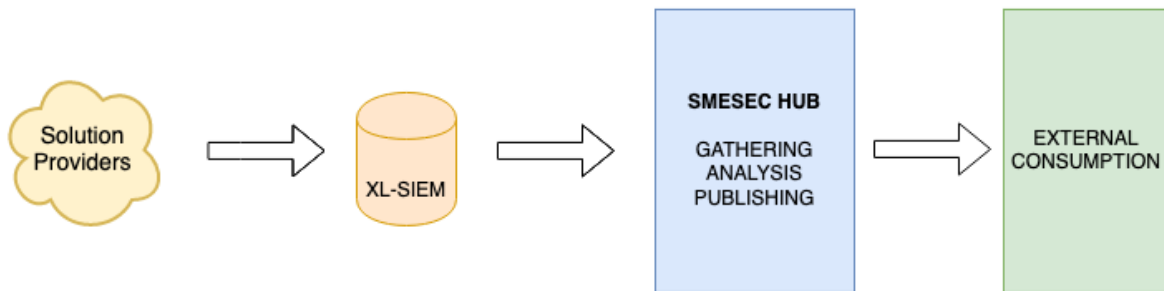


Figure 22: General sketch of the data lifecycle of data within the SMESEC framework Hub

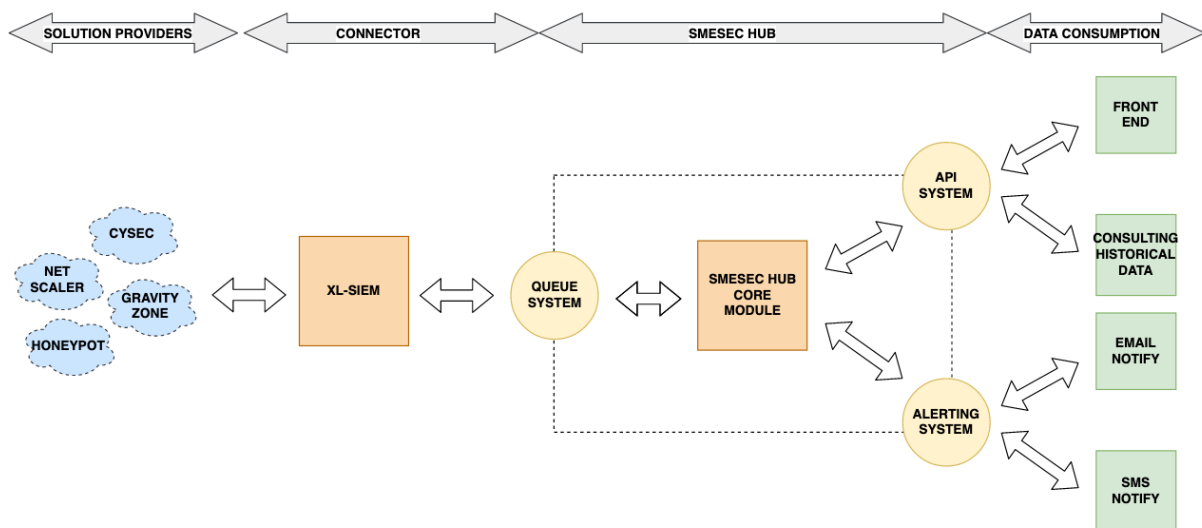


Figure 23: End-to-end architecture of the SMESEC Framework Hub

4.1.2 Solution providers: cybersecurity solutions and data ingestion

All the cybersecurity solution deployed at each particular instance of the SMESEC framework export key data from events and alarms by using their own functionalities. To be correctly processed by the Hub, these data must be properly sent to the system queues by using a standard format (see description below).

4.1.3 Core Module

Once the incoming data are acquired by the Hub connector, the information is then processed by the core module, which is the keystone of the whole system. There, different business rules process the information coming from the solution providers to raise alarms and trigger recommendations to the SMESEC framework user.

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	50 of 85	
Reference:	D3.9	Dissemination:	PU	
	Version:	1.0	Status:	Final

Moreover, and as commented before, it is worth emphasizing the core module is designed in such a way that it is agnostic to the logic implemented behind each particular case. This means that will be adaptable to almost any input and it will be able to provide many different outputs, after an initial set-up period.

4.1.3.1 Business rules

Business rules are the main added value item of the SMESEC Hub. These pieces of software with internal logic are responsible for processing the information received (incoming data), releasing an output intended to help the end-user in managing the system monitored by the SMESEC framework.

Depending on the functionality offered and the involved data sources, business rules are classified as follows:

- **Simple**: they will perform basic operations by using a single data source.
- **Enhanced**: they aggregate multiple data sources and the operations require more complex calculations.

In the table below, some of the implemented business rules per pilot are listed. The main objective of these first proof-of-concept exercises was to demonstrate the versatility of the solution to different needs. By way of example, the full details of the three of them are also provided in this deliverable.

ID	PILOT	NAME	DETAILS / COMMENT
PT_1	Smart City	Alert Major Events	<i>An alert is raised if a major event is received</i>
PT_2	Smart City	Report Threshold Attacks to Endpoints	<i>Every day, Patras University receives a report sent by mail with the details of the attempted attacks.</i>
GP_1	Smart Grid	Rise awareness and recommend measures	-
GP_2	Smart Grid	Non-VPN IP Alert	An alert is raised of a login attempt without using an VPN
SY_1	Smart Voting	Rise awareness and recommend measures	EXPLAINED BELOW
SY_2	Smart Voting	Citrix Detection Alert	Correlation of diverse events detected by CITRIX tool
WS_1	Industrial	Unwanted geolocation filtering	EXPLAINED BELOW
WS_2	Industrial	CPU and processes understanding	EXPLAINED BELOW

WS_1: Unwanted geolocation filtering	
Objective:	Identify connections from non-allowed locations.
Partners involved:	Atos, FHNW, WS
Schema:	XL-SIEM → SMESEC Hub (IP list and locations) → Front end
Input:	Alarm over suspicious IP address.
Processing:	The system has a list of allowed IPs. The system also calculates the location of the IP that is being sent from the firewall.
Output:	If the algorithm detects any location different from the allowed ones there will be an alert launched to the front end to notify the end user (front-end, SMS, email).

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	51 of 85				
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

WS_2: CPU and processes understanding.	
Objective:	Identify over working from our systems and malicious processes running on them.
Partners involved:	All cybersecurity tools
Schema:	Any Provider → SMESEC Hub → Front End
Input:	Form all the servers that are being monitored by the cybersecurity tools, we obtain information of the CPU usage and running process. Data are periodically sent to the Hub.
Processing:	There are two different events that could raise the trigger: if the CPU usage is going over a predefined threshold or if any of the running processes is matching any of the prestored ones labelled as malicious.
Output:	If any of this happens an alert will be sent to the front end, SMES or email.

SY_1: Rise awareness and recommend measures.	
Objective:	Capture all the information from the attacks that the honeypot is collecting to provide to the end user with details and recommended actions to mitigate it.
Partners involved:	Forth, Atos, FHNW
Schema:	Honeypot → XL-SIEM → SMESEC Hub → Front End
Input:	An attack that has reached the honeypot is sent to the Hub.
Processing:	From a prestored attacks database, there will be a search launched to gather all the data available regarding the detected event by the honeypot to transfer this information to the end-user.
Output:	The front-end will display all the available information with details of actions to mitigate the attack.

4.1.4 Data Consumption

After the computation step of business rules, the output information (alarm / recommendation) is sent to the SMESEC framework front-end, where end-users can take meaningful decisions. The methodology to consume these data, both from the perspective of their real-time visualization and the analysis of historical values is done through an API service published by the Hub.

4.2 Interfaces and Connectivity

The key to the success of the SMESEC Hub relies on how raw data coming from the cybersecurity solutions are ingested in the module and the output is delivered to the users. For the sake of simplicity, one preferential single point has been designed and implemented for each step.

4.2.1 Input

The entity selected to exchange information with the cybersecurity solutions is a queues system. In this point, solution providers periodically post the data coming from the pilots (monitored infrastructures) by using MISP-modified messages (JSON format). Details about the exact format were given in previous deliverables.

In principle and to avoid unnecessary messages, the XL-SIEM collects most of the data used by the SMESEC Hub for regular operation; this tool has agents distributed in the SMEs assets, enjoying a privileged position to gather different inputs and deliver them to the SMESEC Hub in a coherent manner. There are however some exceptions, where either other tools directly connect or the SMESEC Hub autonomously interacts with the pilots for extracting very specific pieces of

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	52 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

information. Actually, this is done by remote access to the systems, preserving the confidentiality, integrity and availability of the data. The access is done from the SMESEC Hub due to its limited exposure to the external world in comparison to the XL-SIEM.

In the following table, the most relevant details of the input messages are shown. The agreed protocol among the partners has further elements which are more technical ones. Full details are available upon request.

INPUT FORMAT DETAILS	
FIELD	EXAMPLE
Source of the data	Event → Attribute → PluginID and PluginSID
Timestamp	Event → Date
Attacker (IP, port, host name, ...)	Event → Attribute → Source IP
Victim (IP, port, host name, ...)	Event → Attribute → Destination IP and Port
Severity/reliability/risk numeric indicator	Event → Attribute → Risk value
Additional info (e.g for CPU usage business rule, the list of processes running in the machine and their corresponding %CPU)	Event → Attribute → User data

4.2.2 Output

The system provides two ways of consuming the output information from the SMESEC framework Hub. It can be seen below a schematic of the two options and the results that can be used by the end-users: (i) direct interface through the framework and (ii) alerting system that provides both SMS and email notifications.

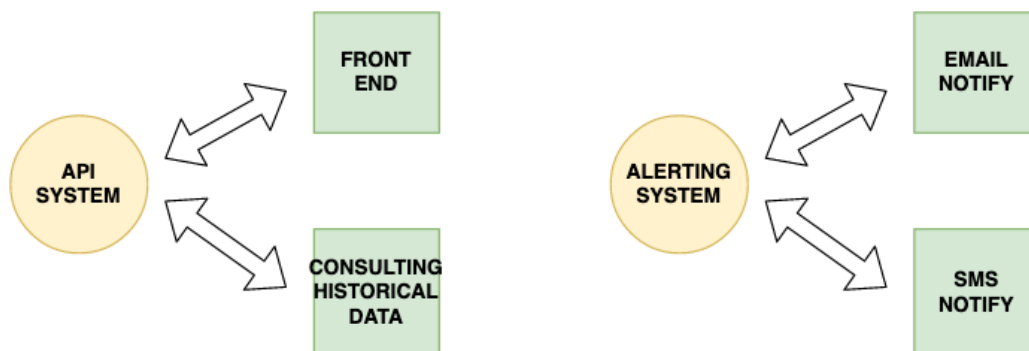


Figure 24: SMESEC Framework Hub Output Information

4.2.2.1 API System

The primary use of the SMESEC Hub is mainly done through the SMESEC framework. There, all the information is displayed in the front end in a friendly way for a standard end-user. On the other hand, the API is also accessible but properly protected with standard encryption and authentication methods. This allows consuming the historical data without the simple user interface, facilitating the interconnection of the SMESEC solution with third-party tools for more advanced applications, such as auditing purposes.

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	53 of 85	
Reference:	D3.9	Dissemination:	PU	
	Version:	1.0	Status:	Final

The following tables show some of the details that are used by the API to make feasible these connections. Depending on the use case and the type of event the codes will obviously differ.

ID	ALERT DETAILS
400	High severity alert received from SIEM
500	Honeypot not running
600	Failed SSH from outside of allowed range
700	Successful SSH from outside of allowed range
800	Number of failed SSH attempts exceeds threshold
900	CPU exceeded threshold
1000	RAM exceeded threshold
ID	ORIGIN OF THE ALERT
7	Gridpocket pilot
12	Syctl pilot
14	Patras University pilot
15	Worldsensing pilot

4.2.2.2 Alerting systems

Standard SMEs but in particular micro and small enterprises suffer from a lack of manpower. This makes difficult to block an employee for continuously monitoring the cybersecurity status of the company's assets. To circumvent this unfavourable scenario, the SMESEC Hub implements an alerting system through email and SMS so that selected people within the company (i.e. CTO & CEO) can receive instantaneous messages in case of a threat happens together with very oriented recommendations. Both the recipients and the action plans are easily configurable, and the idea is to contribute keeping the awareness level regarding cybersecurity high among the key players of new companies.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	54 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

5 SMESEC Framework Implementation

5.1 Description and Objectives

This section describes the prototype of the SMESEC Framework implemented throughout the duration of the Project, having its latest bug fixes integrated on M36. The first implemented version, which was achieved at M12 was only an initial one where all tools were accessible in a unified portal. This was complemented by a very initial version of the authentication module.

This updated version of the SMESEC Framework has been improved from the last one focusing in the dashboard and internal components. We followed the refined version of the architecture in order to create better and improved communication, storage of information, data processing, security, etc. Also, we created a new dashboard as the initial entry point of the SMESEC Framework where we show quick-access information about the cybersecurity status of the system. This was possible thanks to all the information compiled from the tools and the extremely useful feedback of the use cases. After checking the initial version they highlighted how the first thing they wanted for access was “how is my system” and not a long list of tools that they have to directly access for information.

Additionally, we worked in the development and refinement of internal components that provide storage, authentication, etc. The authentication system was integrated in all tools, the framework, the training platform, etc. following the list of roles identified previously.

Look & feel and user-experience is very important for us. SMESEC aims to provide a specialized and unified cybersecurity solution for SMEs. Therefore, and bearing in mind the low-level expertise of most of the employees of these organizations, we had to go through many iterations for refining the usability of the SMESEC Framework. Also, it was important to provide the information in the easier and more accessible way.

Finally, we are working in providing a third-party API for external providers of cybersecurity solutions so they can integrate their solutions into our framework, making it a “cybersecurity market” where SMEs can promote their applications, do business and take advantage of the information compiled from the tools for creating plugins.

5.2 Integrated Tools and Core Functionality

5.2.1 XL-SIEM

As already mentioned in several deliverables, SMESEC framework uses the ATOS XL-SIEM as incident detector, correlating events received from different monitoring agents and generating the corresponding alerts when incidents are detected. Different panels are available at the XL-SIEM to visualize the status of the system, all being integrated into the SMESEC framework prototype. As this tool gathers the data from the rest of tools of the SMESEC Framework, it gives a general overview of what is happening in the system, as it can be seen at Figure 25.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	55 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

This figure contains a table with the alarms raised by the XL-SIEM ordered by date. Also, a velocimeter graph showing how risky are the alerts detected, and a distribution graph showing the evolution of alerts during the last month.

This section is very similar to the first block of the security status overview tab. This is because the XL-SIEM is the tool in charge of the system monitoring and collects alerts from the rest of the tools that compose SMESEC. However, this section does not offer all the alerts shown in the security status overview, as the latter one contains alerts coming from the SMESEC HUB.



Figure 25: SMESEC Framework dashboard. XL-SIEM section

5.2.2 Citrix ADC and Aggregator

Integrating Citrix ADC to the SMESEC Framework

Citrix ADC (formerly NetScaler ADC) is an application delivery controller that performs application-specific traffic analysis to intelligently distribute, optimize, and secure Layer 4-Layer 7 (L4 - L7) network traffic for web applications, provides flexible delivery services for traditional, containerized and microservice applications and delivers enhanced cybersecurity features. Its feature set consists of switching features, security and protection features, and server-farm optimization features. Under the auspices of SMESEC, only part of the overall Citrix ADC functionality was utilized.

The full spectrum of Citrix ADC security and protection features efficiently protects web applications from Application Layer attacks. An ADC appliance allows legitimate client requests and can block malicious requests. It provides built-in defences against denial-of-service (DoS) attacks and supports features that protect against legitimate surges in application traffic that would otherwise overwhelm

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	56 of 85
Reference:	D3.9	Dissemination:	PU
		Version:	1.0
		Status:	Final

the servers. An available built-in firewall protects web applications from Application Layer attacks, including buffer overflow exploits, SQL injection attempts, and cross-site scripting attacks.

Citrix ADC appliance resides between the clients and the servers, so that client requests and server responses pass through it. In a typical installation, virtual servers configured on the appliance provide connection points that clients use to access the applications behind the appliance. In this case, the appliance owns public IP addresses that are associated with its virtual servers, while the real servers are isolated in a private network. It is also possible to operate the appliance in a transparent mode as an L2 bridge or L3 router, or even to combine aspects of these and other modes. As shown in the diagram of the SMESEC Framework Architecture in Figure 7, Citrix ADC is positioned together with the infrastructure it protects and more specific in the SME premises. After being configured, the node can provide advanced secure services as demonstrated in the Pilots. Detailed information on how Citrix ADC is deployed and configured can be found in D4.2 Final Integration Report on e-Voting.

The overall Citrix ADC functionality is based on the notion of **virtual server**, an internal Citrix ADC entity that clients can use to access applications hosted on the servers. It is represented by an alphanumeric name, virtual IP (VIP) address, port, and protocol. The name of the virtual server is of only local significance and is designed to make the virtual server easier to identify. Virtual servers are points for delivering features. Most features, like compression, caching, and SSL offload, are normally enabled on a dedicated virtual server. When the Citrix ADC appliance receives a request at a VIP address, it chooses the appropriate virtual server by the port on which the request was received and its protocol. The appliance then processes the request as appropriate for the features configured on the virtual server. There are several types of virtual servers however for the auspices of SMESEC only the following will be utilized:

Load Balancing (LB) virtual server

Receives and redirects requests to an appropriate server. Server selection is mostly based on the preferred load balancing methods defined by the user during the initial configuration.

Content Switching (CS) virtual server

Directs traffic to a server based on the content that the client has requested. Content switching virtual servers often work in conjunction with load balancing virtual servers.

SSL virtual server

Receives and decrypts SSL traffic, and then redirects to an appropriate server. The appropriate server selection process has many similarities to choosing a load balancing virtual server.

Delivering extended CyberSecurity services in the freemium subscription

Under the auspices of SMESEC project, Citrix had pledged providing thousands of USD worth of licenses for some of their most popular services delivered through Citrix ADC. However, as it turned out, small SMEs are reluctant or even incapable of paying such a premium, regardless of the significant cybersecurity boost it provides. To tackle this situation, we have focused on providing similar services through selective and meticulously deployed LB, CS and SSL virtual servers that efficiently protect the SME servers from most application layer attacks, while always **remain in the**

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	57 of 85	
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status: Final

Free-Tier offering of Citrix ADC. In the unlikely case that the cybersecurity requirements of an SME exceed the provided solution, a license which unlocks additional features must be purchased. The deployed solution is based on Citrix ADC Express and its only functional limitations are 20Mbps throughput and 250 concurrent SSL connections. Virtual server functionality is not compromised or affected; therefore cybersecurity protection resembles to the one of the full-blown Citrix ADC packet.

Pilot Adaptation and generic deployment blueprints

Citrix ADC configuration is rather challenging, given its complex nature not only as a standalone node, but also in conjunction with the overall cloud environment in which it is deployed into. Most SMEs do not have the skilled or effectively trained personnel to modify their cloud infrastructure accordingly or rely on inappropriate cloud solutions for deploying Citrix ADC and fully exploit the functionality it provides. We have tried to address this issue, by preparing de-facto set of detailed deployment instructions as well as material which properly positions Citrix ADC inside most popular Cloud provider infrastructure topology and tries to answer potential deployment questions through visual examples. This set of generic deployment blueprints was evaluated by partner SMEs participating in the Pilots of SMESEC, while a second evaluation round was carried out during the Open Call trial phase.

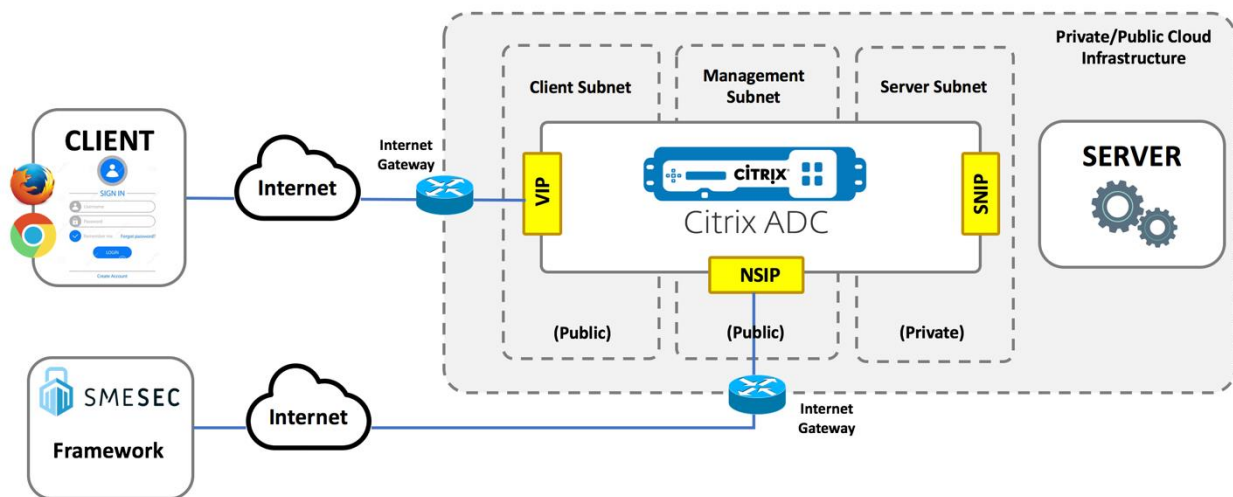


Figure 26. High-level network topology for Citrix ADC deployment in SMESEC

Integration of Citrix ADC and XL-SIEM using SYSLOG message exchange

For enabling proper integration of Citrix ADC with the XL-SIEM residing in the SMESEC Framework it was necessary to slightly modify the internal node configuration which dictates that all Citrix ADC logs must be stored in a proprietary, yet SYSLOG compatible format called NSLOG. The overall process allowed us to (i) obtain logs in SYSLOG format from all types of internal Citrix ADC processes and events, (ii) forward these logs to an external SYSLOG server using the dedicated management interface of the deployment. The only limitation is that the external SYSLOG server which receives SYSLOG messages must either have (i) a public IP or (ii) an IP in the same subnet as the Citrix ADC management one.

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	58 of 85
Reference:	D3.9	Dissemination:	PU
	Version:	1.0	Status: Final

Citrix ADC Aggregator

In proper large-scale TELCO deployments, Citrix ADC is deployed in parallel with a dedicated node the Citrix ADC Management and Analytics System (MAS), which provides centralized network management, analytics, automation, and orchestration to support applications deployed across hybrid cloud and containerized infrastructures. MAS gives admins a single dashboard from which they are able to view, automate, and manage network services across their entire infrastructure. However, also deploying MAS in the SMESEC Framework wasn't an option of many reasons, we therefore opted developing a dedicated node called Citrix ADC Aggregator, specifically for the needs of the project.

Design

Citrix ADC Aggregator exploits the integrated NITRO API of Citrix ADC to issue GET notifications and retrieve specific data regarding the overall functionality of the affiliated Citrix ADC node. The design assumed that only one Citrix ADC node will be deployed per SME and that no other entity of SMESEC framework will be allowed to use the NITRO API.

Citrix ADC Aggregator consist of two (2) different Docker containers namely the (i) Server App container and the (ii) Database container. Server App container is responsible for making the NITRO API calls to the Citrix ADC node through the management interface and most importantly exposes a different, dedicated API to all other interconnected entities. This approach renders Server App as the communication interface between Citrix ADC and the SMESEC framework. The Server App issues GET requests using the NITRO API and obtains data related to the nodes' functionality every 10 seconds which are stored in the database of the Database container. In case the SMESEC Framework makes a request related to historical data using the API, the Server App makes a query in the database and issues the response.

It is obvious that Citrix ADC Aggregator follows a microservice-based architecture which enhances flexibility and efficiency. The Server App and the Database are containerized, fully isolated and communicate via API calls. Moreover, this approach also tackles possible multitenancy issues, since each microservice operates independently while each user accessing the SMESEC Framework only makes API requests to the associated containers.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	59 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

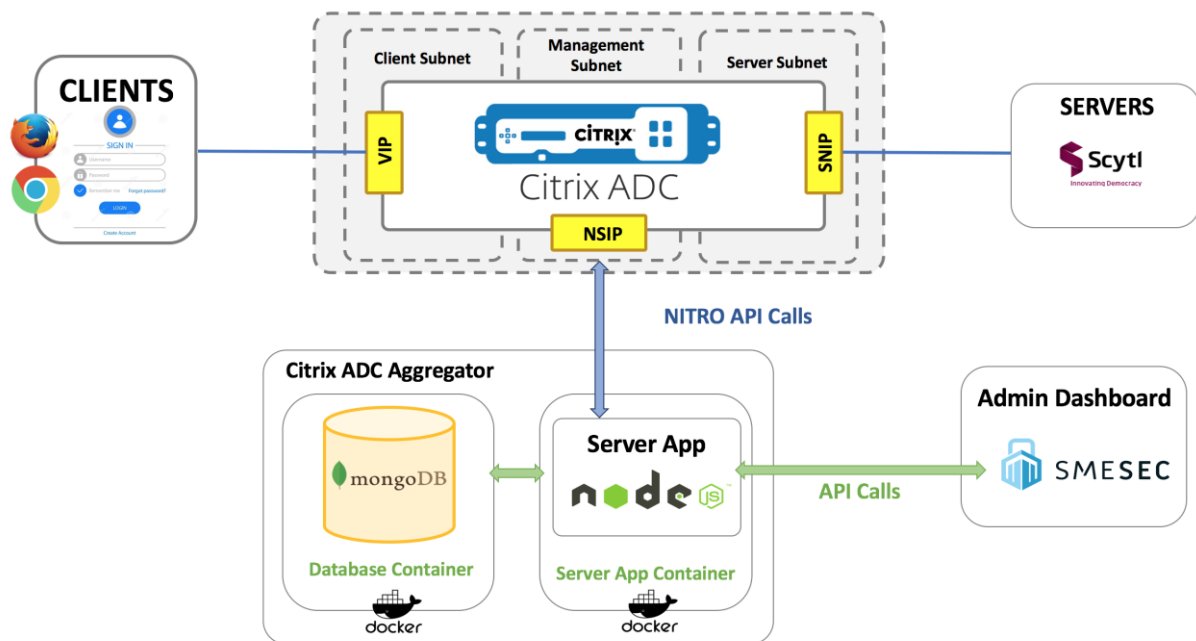


Figure 27: Deploying Citrix ADC Aggregator

5.2.3 Gravity Zone Endpoint

Bitdefender GravityZone is the antimalware solution from the SMESEC framework and consists of the GravityZone console (deployed on premises or on cloud) and the Endpoint Security, installed on each endpoint in the SME.

For integrating in the SMESEC framework, the following aspects were considered:

- integration with other SMESEC tools
- integration in the global dashboard

For the first aspect, we leveraged GravityZone existing capability of sending syslog events. Bitdefender collaborated with Atos for providing security information events to the XL-SIEM component. The integration process and the sent information are described in the section 3.2.1 from Deliverable 3.4.

For the second aspect, we worked on extracting relevant information from GravityZone and present them in the unified dashboard, while also provide minimalistic orchestration capabilities.

The dashboard of the GravityZone console is a comprehensive interface, offering a lot of information for advanced users. In the SMESEC framework context, however, we wanted to provide a glimpse of the antimalware security status. This simplified interface offers the most important information, while the user can navigate to the GravityZone dashboard for more advanced tasks.

First of all, the list of endpoints in the SME network is displayed as a table, with the following fields:

- Endpoint Name
- IP
- Update Status
- Last Update

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	60 of 85
Reference:	D3.9	Dissemination:	PU	Version:	1.0
				Status:	Final

- Last Online
- Status

The Endpoint Security solution runs as usual on each endpoint in the list, and the system administrator will see the most recent detection in the “Status” column. The update status and the time of the last update is also displayed. A Scan Task can be automatically created directly from this table, triggering a full scan on the selected endpoint. A full list of the API calls and more technical details can be found in the Bitdefender Control Center API Guide [39].

5.2.4 EWIS

EWIS system in SMESEC Unified Framework includes the following tools that can be enabled/disabled based on the systems that exist in the SMEs’ IT environment:

- **DIONAEA** [38] is a low interaction malware-capturing honeypot initially developed under The HoneyNet Project’s 2009 Google Summer of Code (GSoC). Dionaea’s main goal is to trap malware exploiting vulnerabilities exposed by services offered over a network, and ultimately obtain a copy of the malware. We performed a patch to Dionaea in order to avoid detection by nmap. It supports IPv6 and TLS and uses Python as scripting language to simulate many popular services such as SMB, FTP, MySQL and others, and libemu to detect shellcodes.
- **KIPPO** [34] is a medium-interaction SSH honeypot. Kippo is used to log brute force attacks and the entire shell interaction performed by an attacker. It includes a modified version of SSH service and is written in Python language. It emulates a Debian filesystem by providing content for key files than an attacker is more likely to access. Whenever a malicious user connects to the Kippo all data are captured and stored for further analysis. It records all commands executed by the attacker as well as all files that are downloaded. Currently and based on the information captured the events the Kippo honeypot can create and share those data through a custom-made syslog wrapper to the SMESEC framework.
- **DDoS honeypot.** Our DDoS solution is based on the ideas proposed in the “AmpPot: Monitoring and Defending Against Amplification DDoS Attacks” presented in RAID 2015 [35]. It can detect amplification DDoS attacks and report the events to the central control system like XL-SIEM. After the first implementations, we fine-tuned it to produce alerts that are in line with the format of XL-SIEM. Also, events based on the alerts were created and represented in the XL-SIEM during the various validation tests.
- **Cloud-IDS** The cloud-based solution can detect possible attacks that take place within a host running many VMs. Virtual hosts hosted under the same Hypervisor can produce orders of magnitude more network throughput than conventional communication over the internet. This happens as VMs are using the internal CPU BUS to communicate, which can lead to throughput over 30 GB/s, and thus an infected VM could produce massive DoS or other attacks against other co-hosted Vms. The detection system that we have devised, is based on a well-known IDS (SNORT [36]) which is deployed within the host OS of the server hosting the VMs. The system includes a database, a log-processing engine and a web based interface to visually present the results. The hypervisor of the system is configured to centrally monitor and log all the “malicious” activity related to the VMs of the specific machine and provide results through a web interface or in the form of raw data. Thus, the solution is able to identify intra-VM attacks and Inter-VM attacks, as well as, attacks originating from wherever in the internet, that cannot be identified by and IDS monitoring the uplink of a cloud infrastructure. The solutions need the installation of a special hypervisor and an intrusion detection system on top of it. The solution can be deployed either to the Cloud or locally. Currently, the feasibility

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	61 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

of our solution has been tested using the XEN hypervisor which we have used and tested for our current implementation. Additionally, events and alerts generated by our system are reported to our dashboard and to the XL-SIEM through syslog.

- IOT Honeypot.** Cowrie [37] is a medium interaction honeypot. It can emulate the SSH service and can log and monitor the shell interaction of the attacker as well as any binaries he might download. We emulate the same services run by the IOT gateway and we have set up cowrie to run on port 22 for SSH in the localhost to log any attempts to attack. In addition, it has Snort installed to listen for DoS attacks in the local network. Thus, if an attacker manages to bypass the router or in the possibility of an inside attack, the attacker will choose to attack our honeypot as it will appear as valid and easy target. Logs of the attack will be reported to our dashboard.
- EWIS Dashboard** From all the above honeypots’ sensors all security information and incidents are sent both to the XL-SIEM agents and via XMPP commands are sent to EWIS backend. All this information is visualized through the Honeypot Panel. In the “Home” page the user can see the data from each honeypot enabled, either Grouped or Stacked for the last 24 hours, for the past 7 days per day. Additionally, when new alerts arrive at the system, while the user is browsing through our dashboards, a pop-up, informs the user of the new alerts as depicted in Figure 28.

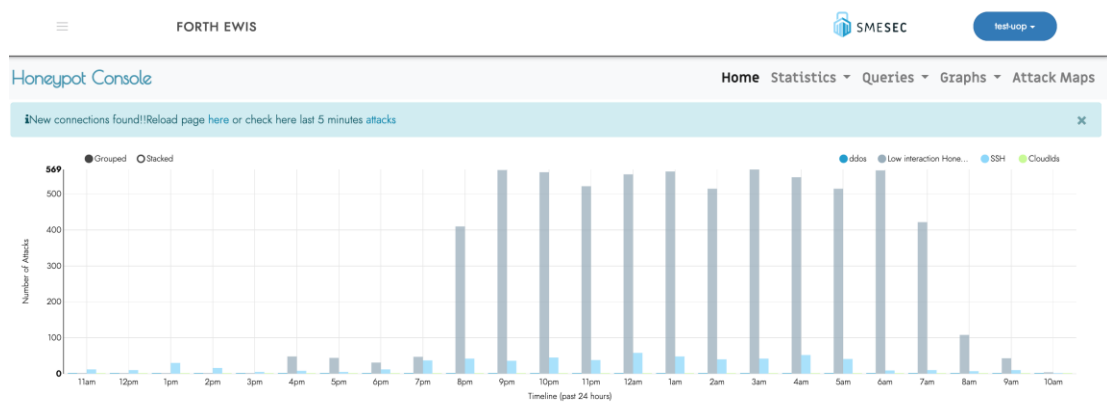


Figure 28 - New Connections Alert: Home Page

Per which honeypots’ sensors are installed in each SME different screens appear in the menu. When DDOS, SSH, Low Interaction Honeypot are enabled the menu looks like Figure 29.



Figure 29 - EWIS responsive menu

In “Top IP/Port statistics” page (Figure 30), there is a list of the top 10 Attacker IPs as well as of the top Ports attacked derived from the databases captured from EWIS Honeypot (DDOS, SSH, Low Interaction Honeypot).

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	62 of 85	
Reference:	D3.9	Dissemination:	PU	
	Version:	1.0	Status:	Final

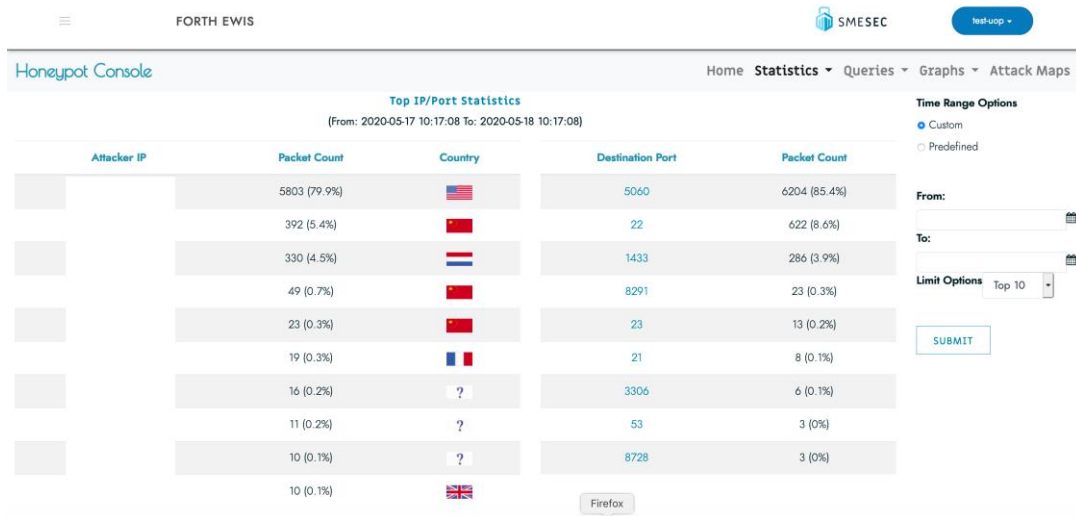


Figure 30 - Top IP/Port Statistics

In the “IP LookUp” page, the user can search for a specific IP for a time range captured from DDOS, SSH and Low Interaction Honeypot. In the “Port Lookup” page, the user can get information about the attacks in specific port for the selected time range captured from DDOS, SSH and Low Interaction Honeypot (Dionaea). In the “Traffic Breakdown Statistics” there are basic figures from data exported from the Low Interaction Honeypot. In “DDOS Traffic” page, data extracted from the DDoS honeypot are presented. In the “SSH Statistics” data captured from the SSH honeypot are presented such as Top Passwords, Top Usernames etc. In “Cloud Based Intrusion Detection System Statistics” all data extracted from Cloud IDS Honeypot are demonstrated in figures such as Attacks per Signature, Attacks per Severity, Attacks per IP etc. In “Summary attacks” basic plots are concentrated for the last 5 minutes, per which honeypots are enabled. Finally, in “Attack Maps” page (Figure 31) there is a world map, where the distribution of Attackers around the world is shown as well as the distribution of Packet Source IP’s. The data for these maps is extracted from EWIS Honeypot.

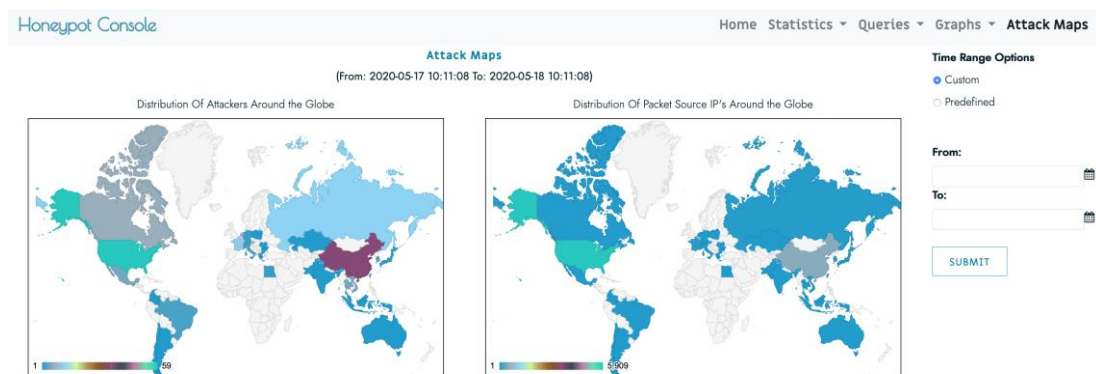


Figure 31 - Attack Maps

EWIS Refinements

During the Open Call, the SMEs were asked to fill in a questionnaire for the installation process and the over usability complexity of installing and using all tool they chose to install in their premises. We focused on the replies concerning the EWIS solution. The feedback received from that process is depicted in the following graphs. The scale for all answers is from 1 (minimum) to 5 (maximum).

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	63 of 85	
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status: Final

Open Call Category 2a Only - Honeypot

How complex is to install the agent of the Honeypot?

3 responses

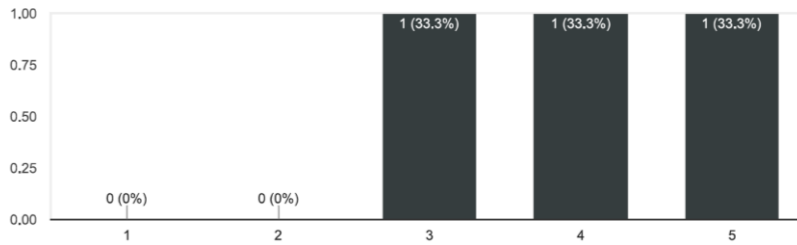


Figure 32 - How complex is to install the agent of the Honeypot per Company?

Figure 32 and Figure 33 present the responses of the 3 SMEs that integrated and tested the EWIS on the complexity of the installation/configuration process as well as the maturity of the installation guide. The responses received from the external SMEs denoted that the EWIS' complexity varies from average to too complex to install/configure. This variation of responses denotes that the system needs a medium to strong IT background to understand its concept and move forward with the installation process. Figure 34 along with the feedback and the interactions with SMEs during the integration process made apparent the need of the need of updating the installation manual that we had initially provided to the SMEs.

How complex is to uninstall/remove the agent of the Honeypot from your system?

3 responses

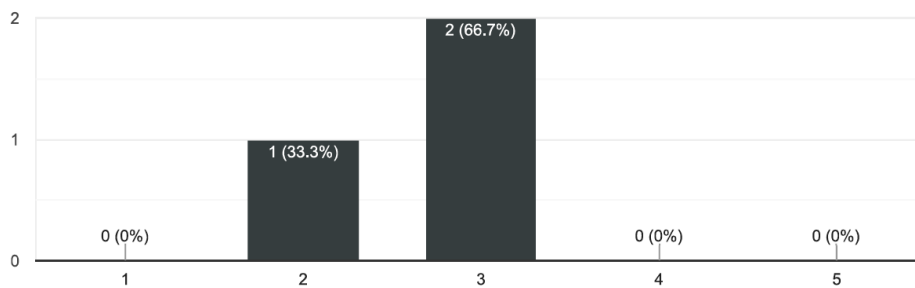


Figure 33 - How complex is to uninstall/remove the agent of the Honeypot from your system?

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	64 of 85
Reference:	D3.9	Dissemination:	PU	Version:	1.0
				Status:	Final

How useful were the instructions (e.g. documentation, videos) for installing/configuring the Honeypot?

3 responses

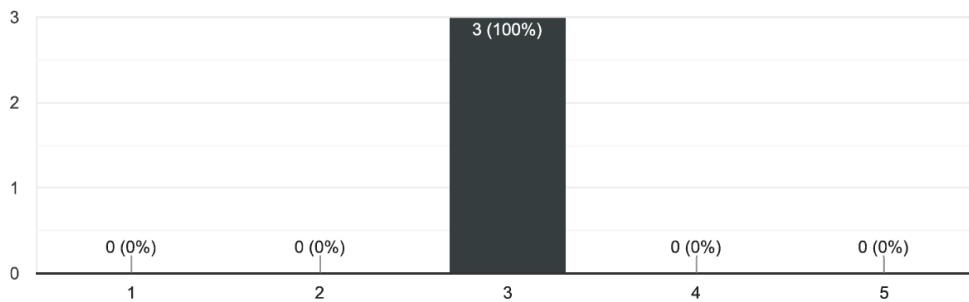


Figure 34 - How useful were the instructions (e.g. documentation, videos) for installing/configuring the Honeypot?

Figure 35 and Figure 36 show that EWIS can work and be integrated with an existing IT system without the need of changes to that system, in a parallel and seamless way.

Did you have to prepare your system before installing the clients/agents?

3 responses

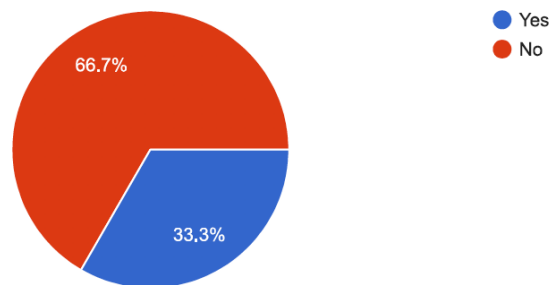


Figure 35 - Did you have to prepare your system before installing the clients/agents?

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	65 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

Did you have to update/install additional software for installing a component?

3 responses

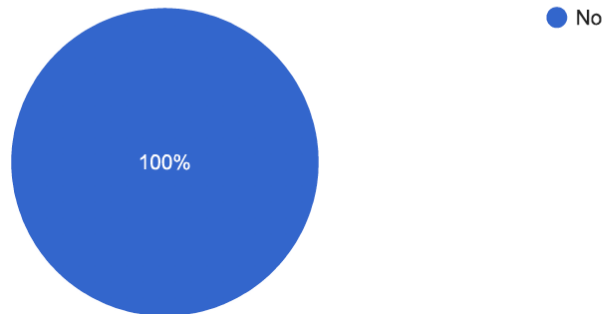


Figure 36 - Did you have to update/install additional software for installing a component?

Figure 37 summarizes the average grade for each question from the responses that were received from the Open Call SMEs. It seems apparent that the required technical expertise to install EWIS was not present in the selected SMEs resulting in delays and extra clarifications to be given to them, regarding the installation and use of EWIS.

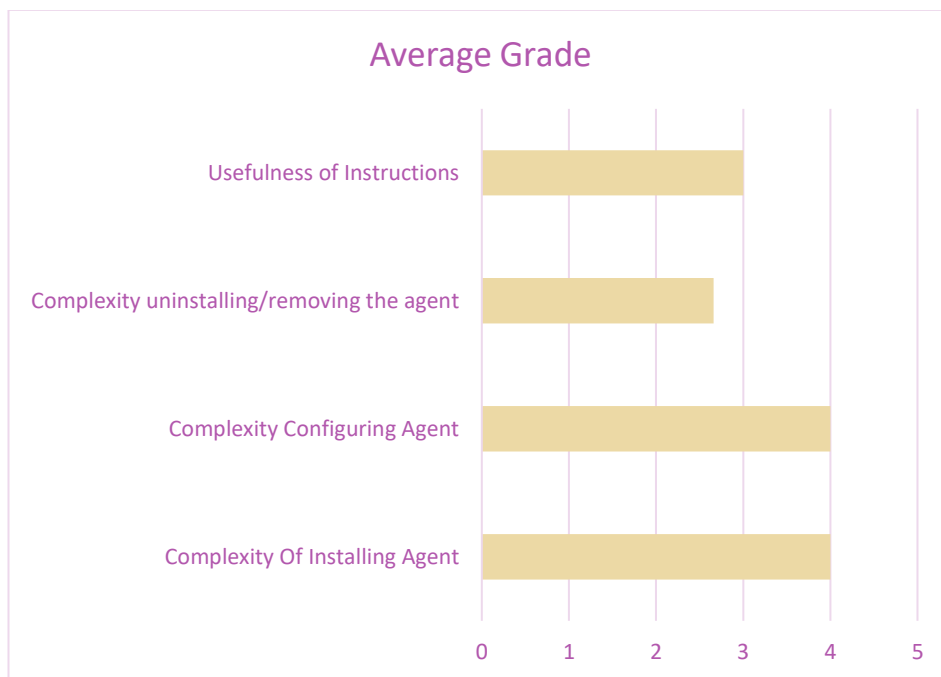


Figure 37 - Average grade per Question for all the concerned SMEs

A summary of the recommendations/comments received during both the Pilot and Open Call evaluation along with the mitigation actions is the following paragraphs:

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	66 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

- *Clarifications were requested from various SME's, about the installation process of our tools, denoting that there was a need for a revised version of the installation guide. Finally, from the whole process we realized that EWIS system is a SOTA deeply technical system that will an installation facilitator and on-call support for the installation and the use of the EWIS system.*
- *EWIS bridged mode poses limitations. Even though we spent time examining this matter, unfortunately is something that cannot be circumvented as is it a low-level core component of the system.*
- *Provide localized version supporting multiple languages.*
- *It was recommended that the update of the sensors to be an automatic procedure*
- *Some performance issues on the retrieval of attack data through our dashboard were observed and reported.*

Based on the above the following refinements were made to the tools.

- During the Open Call, we received certain questions from various SME's for the installation of our tool, making apparent that it needed more details to be included. For that reason, we made the following changes to our installation guide. Firstly, we added a section in the possibility that the SME cannot get a dedicated public IP address for EWIS due to their network topology and restrictions. Our guide now provides instructions on how to correctly set up the port forwarding and changes that must be performed on the certificate generation process in order our tool to work with a local IP address. In addition, added text on how to change our tool's networking interface name, in the case that one on use is not the correct one, was added.
- FORTH, also provided the Greek translation for the localisation of the global SMESEC dashboard.
- FORTH performed minor updates to the EWIS dashboard and enhanced its performance by refining numerous DB queries and resolving a time-consuming bug.
- FORTH provided a new version of the EWIS bundle that includes an auto-update feature and every time a new version of any of the sensors included in EWIS instance is available, it's automatically downloaded and installed to all registered instances.
- In the initial design of the EWIS system, a honeypot specifically for the IoT domain was not included. After closely examining the Industrial IoT pilot we concluded that an IoT-like honeypot should be present in our proposed solution. Thus, we included the IOTHoneypot, which is built on top of Cowrie [37], to our EWIS bundle. It is a medium interaction honeypot capable of emulating the SSH service and can log and monitor the shell interaction of the attacker as well as any binaries he might download. We examined and created an emulated version of the IOT gateway of our industrial IoT pilot emulating the same services, in order to lure attackers and provide early warning alerts based on the events gathered. We have set up cowrie to run on port 22 for SSH, as it is the dominant port/protocol used for the development of IoT applications, in the local network to log any attempts to attack. In addition, we enhanced the IOTHoneypot sensor capabilities by setting up Snort to monitor the traffic inside the network and report any suspicious behaviour or DOS attacks attempts. Thus, if an attacker manages to by-pass/compromise the router or in the case of an inside attack, the attacker will most probably select to attack our honeypot, since it appears as valid and easily exploitable target. If an attacker successfully connects, we will capture his interactions with the honeypot (commands executed and binaries he down-loaded) and all attack related information and an alert will be created/produced. Logs of the attack will be reported to our backend and displayed in our dashboard and via syslog to the deployed XL-SIEM from ATOS.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	67 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

5.2.5 CySec

CYSEC is an integrated tool for the continuous improvement of an SME. It provides the SME Administrator to benefit from interactive expert advice. The advice is generated dynamically based on questions and answers encoded in so-called coaches and selected by the human end-user. This mechanism of integrating self-assessment and learning from the advice allows dynamic adaptation of cybersecurity to the SME based on the information offered by the SME.

Unlike other tools, it has no statically pre-plotted path. Instead, it is possible to collect data about infrastructure based on the previously given input. Depending on the data, additional information is collected. Furthermore, questions may be revisited and revised at any time. Based on the data collected, the tool issues recommendations and orders them by importance.

The tool offers not only Information collection. Half of the screen is dedicated to providing knowledge about the topic currently being covered by the coach. This knowledge interface provides the user with the ability to improve knowledge and skills. Furthermore, to reward the user for achievements, the coach may award batches.

The CYSEC tool is available via the SMESEC framework. It may be installed standalone on the premises either by running a docker or installing it via an APT-Repository on Linux. For Windows, a generic WAR container for a Tomcat9 server is provided. FHNW released CYSEC as open-source software hosted on Github and released under the Mozilla Public License V2 (MPL2).

Within the SMESEC framework, all the information collected is available to other applications via an API. For example, The SMESEC framework receives all CYSEC recommendations and displays them in a combined view in terms of a dashboard to the user. Other applications may collect data from CYSEC, such as answers or specific scores via the API.

On-premise installations have the option to replicate parts of the information entered on-premise into the cloud, allowing them to get custom feedback from a central infrastructure.

5.2.5.1 CYSEC Coaches

Right now, the following Coaches are available within the SMESEC Framework:

- A company coach
This coach collects general data about a company to improve the quality of answers. Furthermore, this coach provides some very basic, generic guidelines about GDPR and Data protection without going into detail.
- A backup coach
This coach offers advice on the choice of backups, backup types as well as helping in the process of integrating an adequate backup into a complex environment.
- A patch management coach
This coach determines weaknesses in the patch management of an SME and offers advice on how to fill the gaps.
- A user training coach
This coach offers advice on how to train users and helps the SMEs to stay alert of any missing trainings they might have.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	68 of 85	
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status: Final

- A malware scan coach

This coach is analyzing the situation regarding malware weaknesses within the SMEs infrastructure.

5.2.5.2 Standard Coach Language (SCL)

A specification language has been developed to provide cybersecurity experts with the ability of flexibly specifying coaches for CYSEC and adapting existing coaches, e.g. for supporting a new language. Java elements of the language allow utilizing the CYSC platform API.

To minimize the complexity of specifying a coach, we developed a generic coach library allowing to formulate sophisticated expert know-how in XML. This generic, XML capable language was named „Standard Coach Language” (SCL).

SCL simplified the process of writing coaches tremendously. By using the SCL library, we may use an indefinite number of sub-scores allowing us to formulate fuzzy conditions for branching within a coach as well as complex scoring. Adapted recommendations do no longer require code for the coaches to be written. Instead, they may be written in XML like the rest of the coach.

The following excerpt gives an example of a coach handler:

```

<metadata key="_cysec.logic">
  <mvalue key="default">
    <stringValue>
      TRUE : default : {
        addScore("knowhowMax", 1);
      };
      isAnswered("company-q20") : q20 : {
        addScore("knowhow",1);
        createSubcoach("lib-access-control", "default");
      };
      not(isAnswered("company-q20")) : q20not : {
        addScore("knowhow",-1);
      };
    </stringValue>
  </mvalue>
</metadata>

```

5.2.6 Test-as-a-Service

EGM Test-as-a-Service (TaaS) is an online and offline testing solution where users are allowed to setup their System Under Test (SUT) configuration and launch test execution without any manual installation on the machine itself. It is possible for end-users to configure the tool through a web application, select which test cases should run, and TaaS will produce readable reports in the web interface containing statistics, reports about test failures.

EGM TaaS is primarily based on Model-based testing for generating realistic test cases. The solution is available at two levels, with the first being launched online, as a web service having a client connected to the services and execute some tests, while the second is hardware-bound. Figure 38 and Figure 39 show the internal architecture of EGM TaaS and the key interactions with the users and the SUT device, for the online and the offline test execution respectively.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	69 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

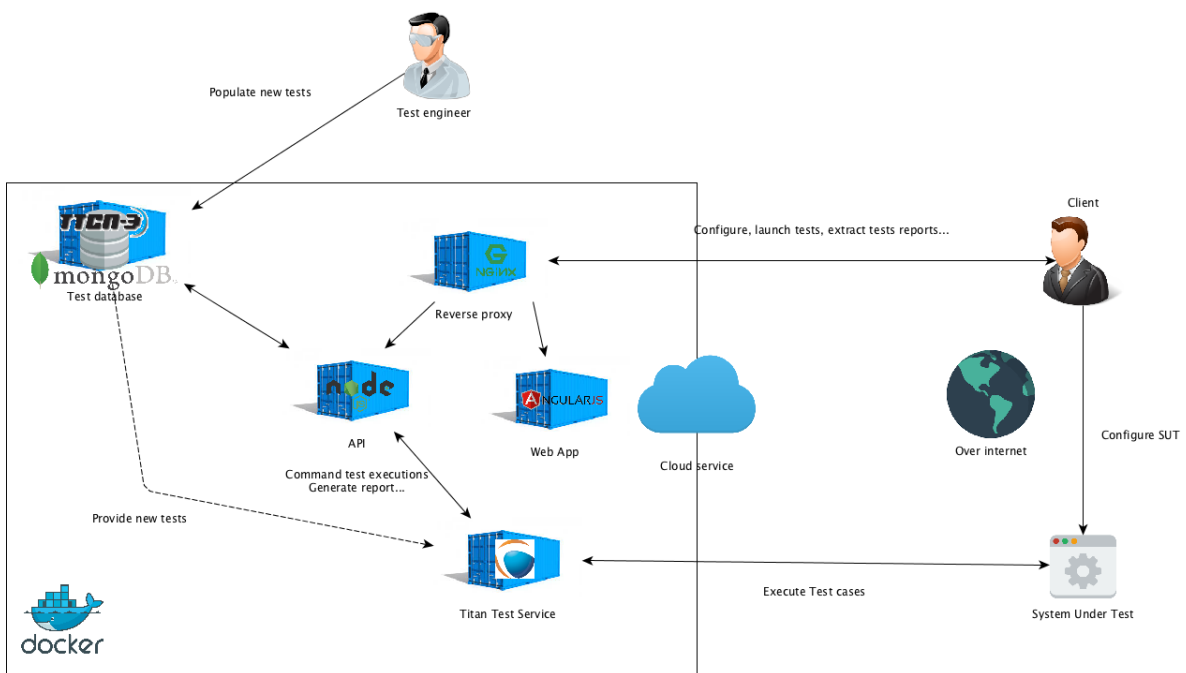


Figure 38. EGM TaaS Architecture

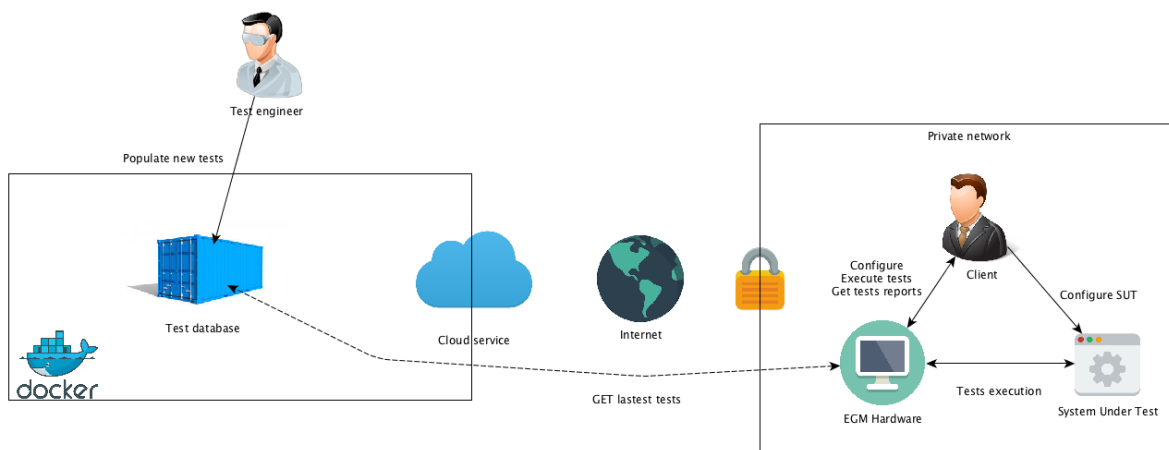


Figure 39. EGM offline testing

TaaS integration with SMESEC project

To fully meet the technical requirements of the SMESEC Project, as described in D2.1 [33], additional configuration (which involves modification and adaptation of specific architecture blocks) is required:

- Externalise the authorization and authentication functionalities to a third part provider, the SMESEC project requires to have a unified security solution (Keycloak server hosted by ATOS) and it should not be handled by the tool itself (The full description of the Keycloak integration is described on D3.4 [29]).
- Change the TaaS frontend style with the SMESEC project CSS.

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	70 of 85	
Reference:	D3.9	Dissemination:	PU	
	Version:	1.0	Status:	Final

- Add two test services (API and LoRa) to meet the pilot’s requirements: The TaaS, as described in the previous section, implements a micro-service architecture, where each test service (oneM2M, API, semantic validation ...) is represented as a micro-service in the global architecture. A test coordinator service is also implemented, which play the role of the orchestrator between the different services, if any interaction is needed (for example, any test service with the reporting service). Figure 40 shows what we have described. Both test services (API and Lora) has been described in the deliverable D3.4 [29], while the overall integration description can also be found in D4.9 [28].

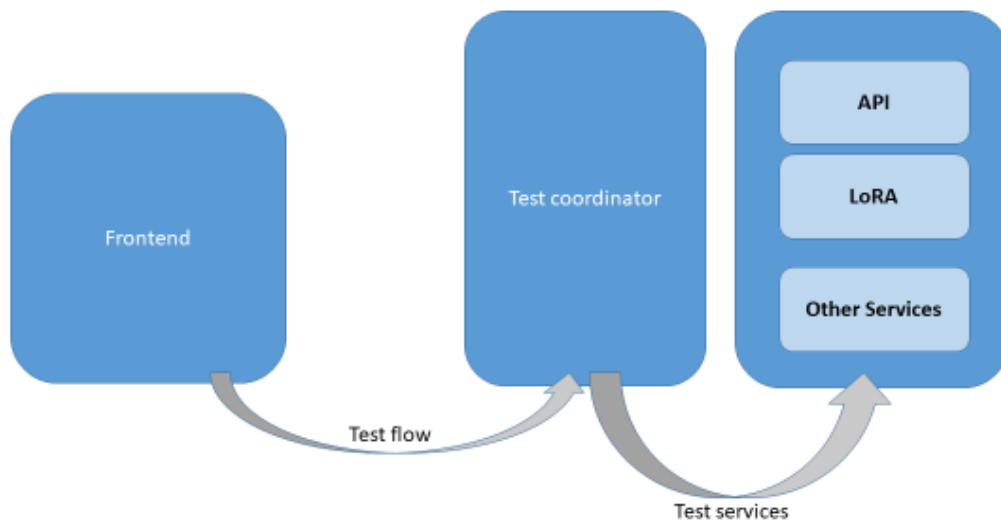


Figure 40. TaaS Micro service Architecture

5.2.7 Virtual Patching

Virtual patching is tackled in SMESEC using IBM’s AngelEye tool. AngelEye receives as input an application’s source code or binary and produces a virtual patch of the application. A provider of security solutions can use AngelEye to create a predictive model that will predict if an input to an application will allow an exploit of a vulnerability in this application. This predictive model can be integrated into the security solution and its results can be used to detect or protect against vulnerability exploit attacks. An optional input to AngelEye is a testing corpus of the application under test; this corpus can include the latest discovered CVE’s of an application.

Figure 41 shows the overall AngelEye architecture.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	71 of 85
Reference:	D3.9	Dissemination:	PU	Version:	1.0
				Status:	Final

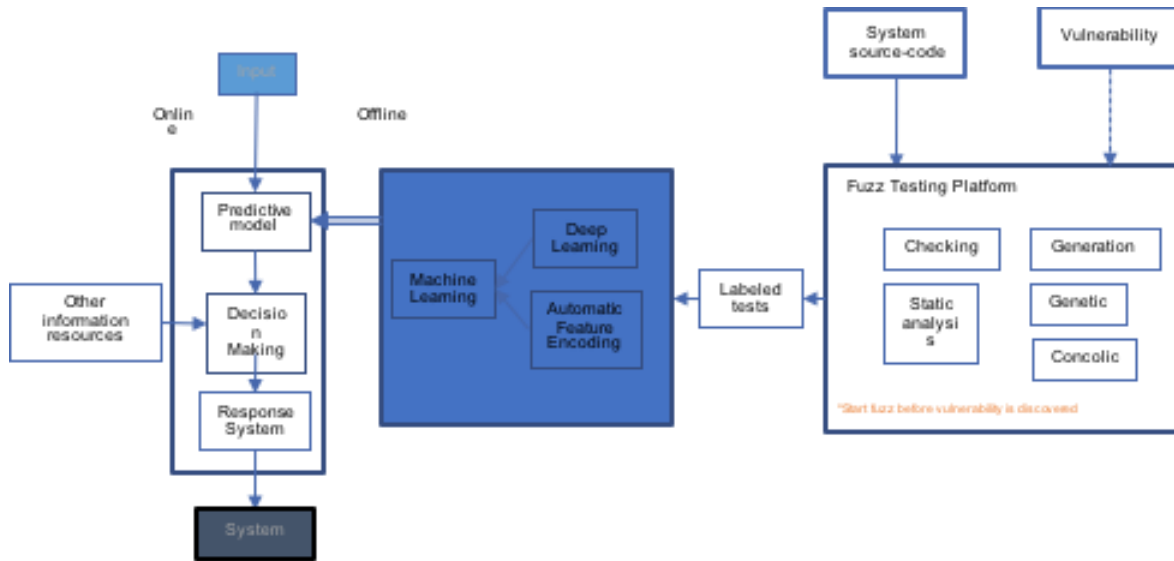


Figure 41 AngelEye solution architecture

5.2.8 Testing Platform - ExpliSAT

ExpliSAT is integrated into AngelEye as a testing platform and acts as another fuzzing engine. ExpliSAT receives source code and a test as input and produces a number of new tests that can execute run-time paths adjunct to the run time path of the given test. Figure 42 shows the architecture of the interaction of ExpliSAT (symbolic interpreter) and genetic fuzz testing.

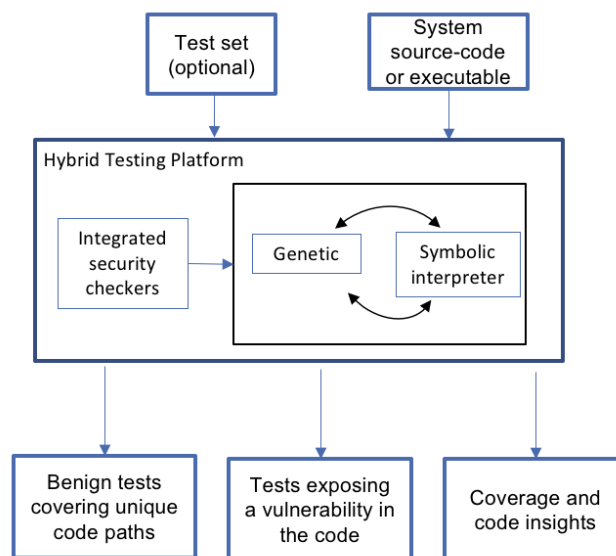


Figure 42: Hybrid testing platform

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	72 of 85
Reference:	D3.9	Dissemination:	PU
		Version:	1.0
		Status:	Final

5.2.9 Training Platform

SMESEC project created and published a complete set of online courses to increase security awareness and also train its users on how to configure and operate the SMESEC framework. To provide these courses, the project adopted the free e-learning platform designed and operated by one of its partners (UoP). This platform is called SecurityAware.me and can be found at the <https://www.securityaware.me> website.

SecurityAware.me is a platform which allows users to create and manage "interactive" online courses using real infrastructures and testbeds (servers, computers, networks etc.) across Europe. Contrary to various e-learning platforms SecurityAware.me focuses solely on cybersecurity. Experts from security companies and institutes around Europe are invited to create courses and contribute training material for various security topics and levels of complexity. A detailed overview of how SecurityAware platform is integrated in the SMESEC Framework can be found in D3.6 Section 4 [32].

5.2.10 Moving Target

Anti-ROP is a tool to create applications that cannot be exploited by malicious software by shuffling its building blocks. It comes in the two versions: one for binary and one for source files.

Anti-ROP for binary receives as input a binary executable file and outputs an executable with a randomized order of the original executable's building blocks, while keeping the original functionality intact. A user can use the Anti-ROP solution to randomize an executable running in the system, and effectively protect this executable from any vulnerability exploit attack. This architecture is depicted in Figure 43

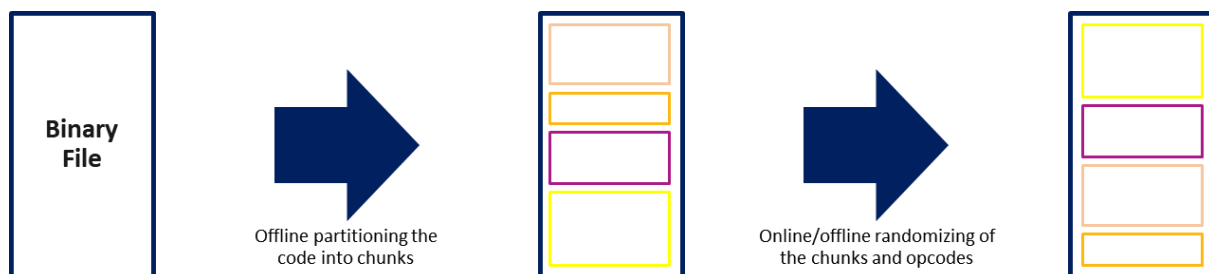


Figure 43: Anti-ROP for binary

In Anti-ROP for source, the input is a source code of a file or number of files, and a randomization seed. The compiler runs and the Anti-ROP plugin is invoked to randomize the order of the blocks. The output is a binary file which has the same functionality and blocks as compiling without Anti-ROP plugin, but with different order of blocks. Anti-ROP for source can be used for creating many unique copies of the same functionality and effectively protecting against exploitation of vulnerabilities. This architecture is depicted in Figure 44.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	73 of 85
Reference:	D3.9	Dissemination:	PU	Version:	1.0
				Status:	Final

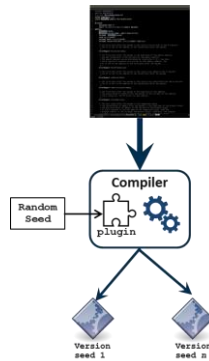


Figure 44: Anti-ROP for source

5.2.11 HUB

As explained in Section 4, the main goal of the Hub is to provide a simple but useful tool that can be integrated into the day-to-day of SMEs, in particular those with limited knowledge on cybersecurity. In this sense, the interface with the module has been made as simple as possible, keeping a central role in the SMESEC framework front-end.

Thus, by defining which business rules are sensible to be used in a specific deployment and after activating them, the front-end clearly displays in real-time the outputs (attacks and threats), indicating the recurrence, criticality and timestamp of each one (Figure 46). In this way, the end-user will have a first and clear view of what is happening in the company. This information can be later complemented with the full details provided by the specific cybersecurity tool covering the attack (i.e. XL-SIEM or honeypot).

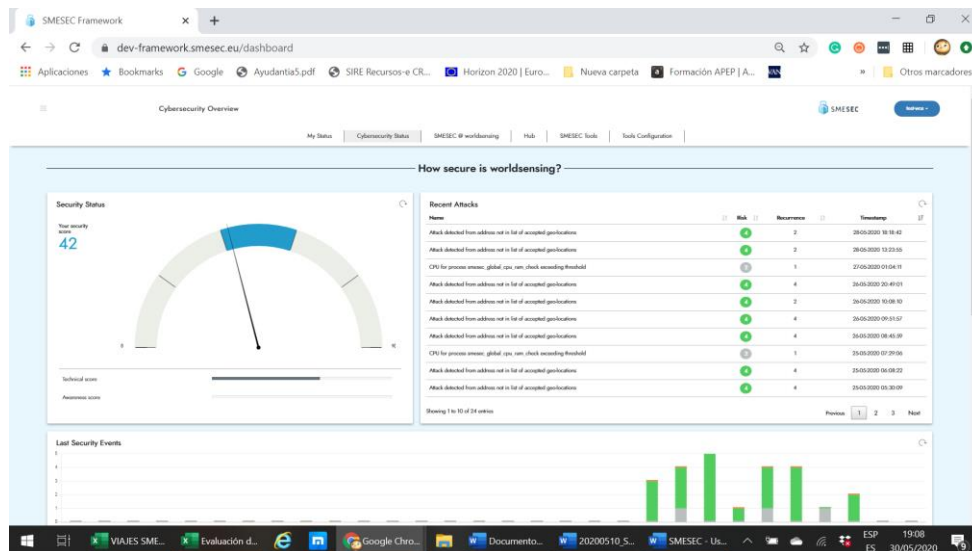


Figure 45: General view of the SMESEC framework front-end. The outputs from the Hub are displayed in the upper-right part.

Nevertheless, the full potential of the Hub is the definition of “response plans” in an easy way. As shown below, through an intuitive interface, the system allows setting-up alarms, and

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	74 of 85
Reference:	D3.9	Dissemination:	PU
		Version:	1.0
		Status:	Final

recommendations to be received by selected people within the company so that mitigation actions can be immediately triggered.

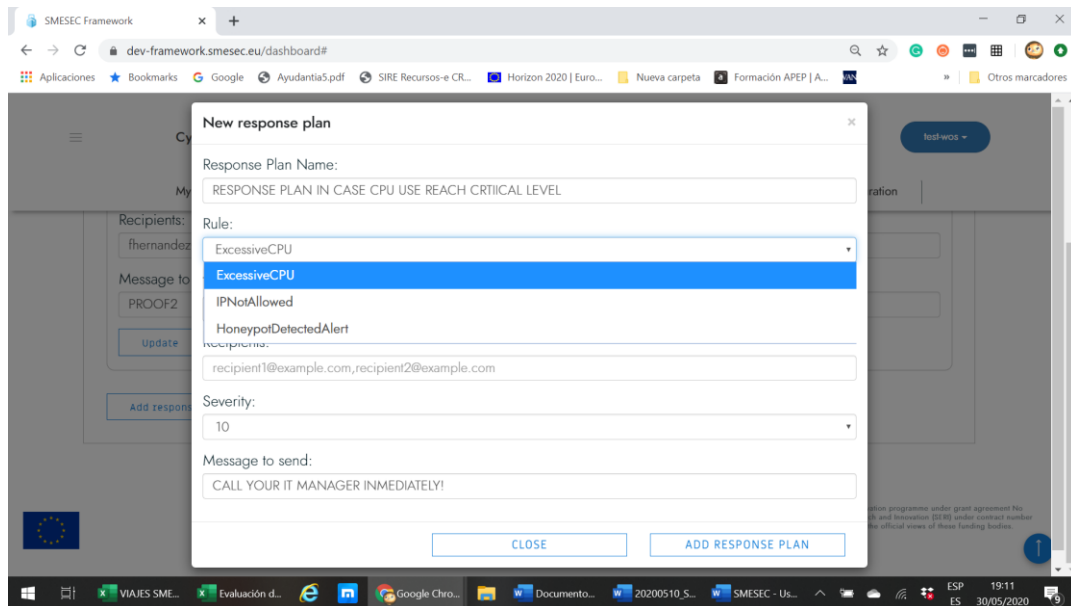


Figure 46: Setting-up Rules

Hence, a complete set of responses plans can be activated for each business rule, through ad-hoc and separated messages (SMS & email) to different people with specific profiles in a SME (see images below). In this way, the final objective is to get everyone in micro- and small companies to participate in the tasks related to cybersecurity, from a technical to a business perspective.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	75 of 85	
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status: Final

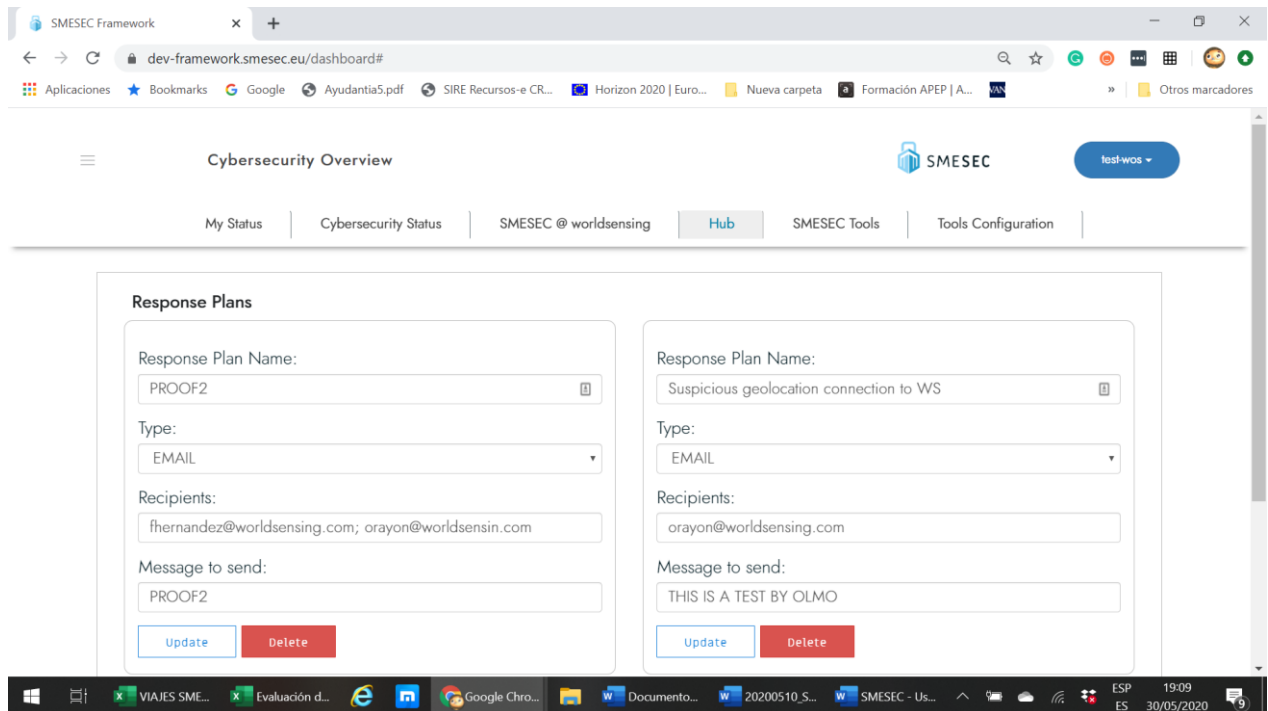


Figure 47: Responses plans generated for Industrial Pilot (example)

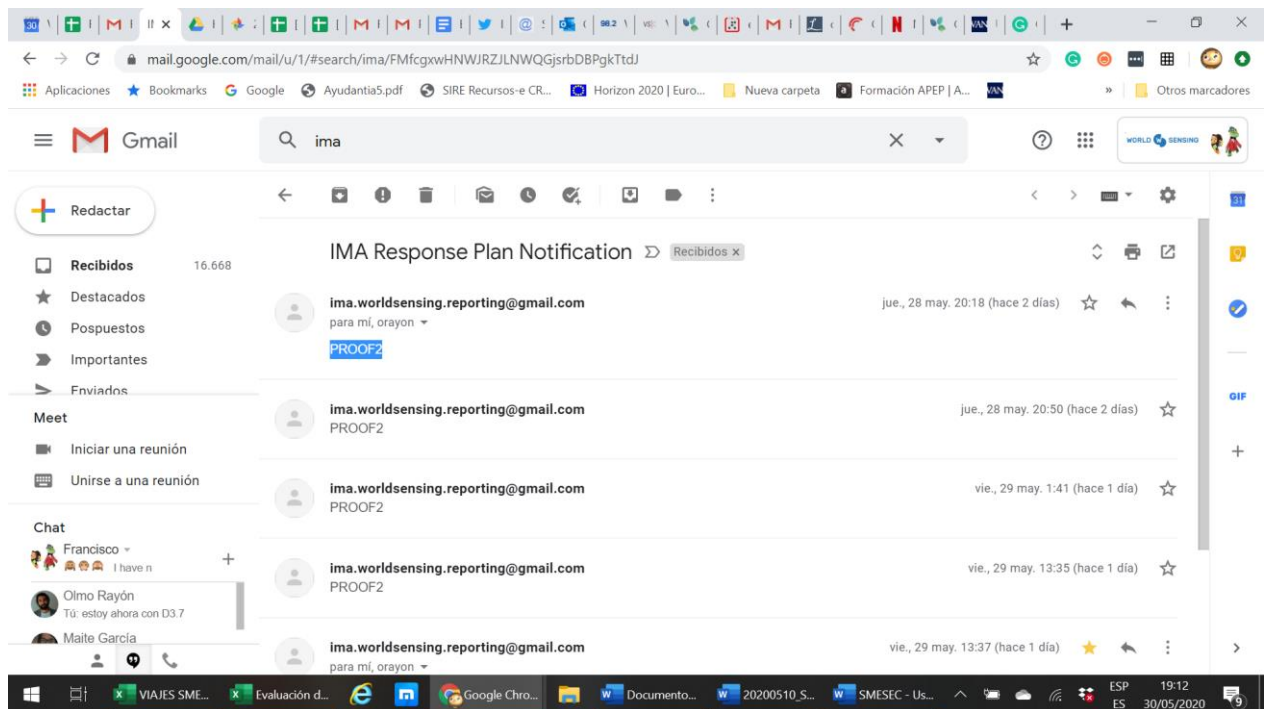


Figure 48: Automatic emails sent by the system when key events occur

In short, the SMESEC Hub is up and running. It has been integrated between the cybersecurity solutions and the presentation module of the framework to fill the gap necessary to mainly raise the awareness of non-experts' profiles in start-ups and similar companies through a dynamic alarm system.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	76 of 85	
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status: Final

5.3 Development Environment and Frameworks

For the development of the SMESEC Framework, it has been decided to use the Java programming language as main technology, because of its flexibility and level of support. The SMESEC Framework uses as a core Spring Boot, an open source framework sponsored by Pivotal [11].

For the visualization part, the Thymeleaf template engine [12] is used, in conjunction with open source CSS and Javascript frameworks, such as jQuery, Chart.js and Bootstrap [13].

In order to support the development, a continuous integration environment was deployed. This environment is composed of a continuous integration server, using Jenkins [14]. This server automates the necessary tasks to compile the code, perform the tests, analyze code for bugs and possible vulnerabilities in third-party dependencies (described below), creating the docker image and deploy it to a container, so a test instance is always up and running with the latest changes ready to perform integration tests.

More information about the integration environment can be found below, under Section 5.5.

For performing these tasks described above, we use Maven [15] as build system, known for its stability and available plugins for extending the functionality.

Besides this infrastructure, a Nexus Repository Server [16] is deployed to store the different snapshots and versions for both the SMESEC Framework compiled code and the Docker images used for deploying it.

5.4 Integration Methodology

The integration of the different tools composing SMESEC in the SMESEC Framework have been done in two different ways, depending on the existing capabilities of each tools.

For the XL-SIEM, GravityZone, and EWIS, the tool's own dashboards are showed in the SMESEC Framework as iframes. This is done due to the impossibility to recreate the complete functionality of the tool with API calls. The approach taken here comes with the downside of showing many different tools, each with its own look and feel, in the same website. This has been overtaken updating each tool style, so they adapted to the general SMESEC look and feel.

For the rest of the tools, dashboards have been created from scratch using API calls, and displaying the required information to the user.

A special case is the "Security Status Overview" dashboard. In this part information coming from all the tools have been combined with the goal of providing intelligent insights to the SMESEC customers. This integration is provided in both the XL-SIEM and the SMESEC HUB. These tools expose an API, from which the data is retrieved and displayed.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	77 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

5.5 Technical Infrastructure

To support the development and integration environment, four different virtual machines have been allocated at ATOS premises.

These virtual machines provide support for the following functionalities:

- **Authentication.** This contains the Keycloak server, along with a PostgreSQL database and a LDAP server that serves as backend for user storage.
- **Monitoring.** A Zabbix [17] instance in charge of collects data from all the agents deployed in the rest of the servers of the infrastructure. This tool is able to warn about possible problems before they cause an outage of any of the services. The Zabbix server is also configured to monitor the SMESEC tool's availability.
- **Artifact storage.** A Nexus Repository Server configured with a Maven repository and a Docker registry. This server is in charge of storing a copy of the jar file containing the SMESEC Framework code, along with the Docker image used as a base for the running container for each version.
- **Continuous integration.** The CI server is composed of a Jenkins instance, a Sonarqube instance, and a Docker CE installation, that serve to continuously test, build, analyse and deploy the code of the SMESEC Framework.

The technical description of the hardware used for supporting the infrastructure can be found below:

Table 5: SMESEC Framework Deployment Infrastructure

SERVER	CONTENT	vCPU	RAM (GB)	Disk (GB)	OS
Authentication server	Keycloak, PostgreSQL, LDAP Server	2	16	70	CentOS7
Monitoring	Zabbix, OpenVAS	4	8	30	CentOS7
Artifact storage	Nexus Repository	2	8	200	CentOS7
Continuous integration	Jenkins, Sonarqube, Docker	2	32	100	CentOS7

5.6 Authentication and Security

The authentication of the SMESEC Framework is provided by Keycloak [3], using the OpenID protocol [18] for both authentication and authorization.

For each request, the access token of the user is checked against the Keycloak server for its validity. It also checks if the user has the necessary permissions to perform the request. For these actions, we use the official Spring Boot adapter [19], provided by Keycloak. The roles we defined for accessing the

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	78 of 85
Reference:	D3.9	Dissemination:	PU	Version:	1.0
				Status:	Final

SMESEC Framework are the ones defined in the previous deliverable. Also, what can be accessed in the SMESEC Framework is described in the previous section.

The Keycloak Spring Boot adapter uses the Spring Security [20] framework under the hood, which is also used to assist in the protection against XSS or CSRF attacks.

Nevertheless, the protection against XSS attacks that Spring Security provides relies in the browser capability to understand the X-XSS-Protection header [21], so every input of the SMESEC Framework needs to be sanitized. At this moment this is not implemented, since the SMESEC framework does not expect any user input. In order to prepare for providing this security measure, the SMESEC Framework is making use of the OWASP HTML Sanitizer Project [22], which is already configured and ready to use.

Also, Content Security Policy [23] is planned to be implemented so only trusted sources are allowed to execute scripts in the SMESEC Framework. This security measure will help us preventing clickjacking attacks.

Also, to ensure that the code of the SMESEC Framework is free of vulnerabilities, we run static code analysis with SonarQube [24], using the FindBugs Security Audit [25] profile. Besides this analysis, and given that we are using many third-party dependencies, OWASP Dependency Checker [26] is being used to analyze possible vulnerabilities in the dependencies used, so we are able to upgrade those dependencies as soon as possible.

Finally, we plan to have a red-team (thanks to the open call of the project) for checking the resilience and security of the framework. The idea is that they perform several exercises and in each iteration, give us feedback for improving the system. Having different tools in a unified framework means the communication and data storage is critical so this will be one of the main points of action.

5.7 Deployment and Configuration

5.7.1 Deployment and configuration of the SMESEC Framework core

The current deployment diagram of the SMESEC Framework is depicted at Figure 49:

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	79 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

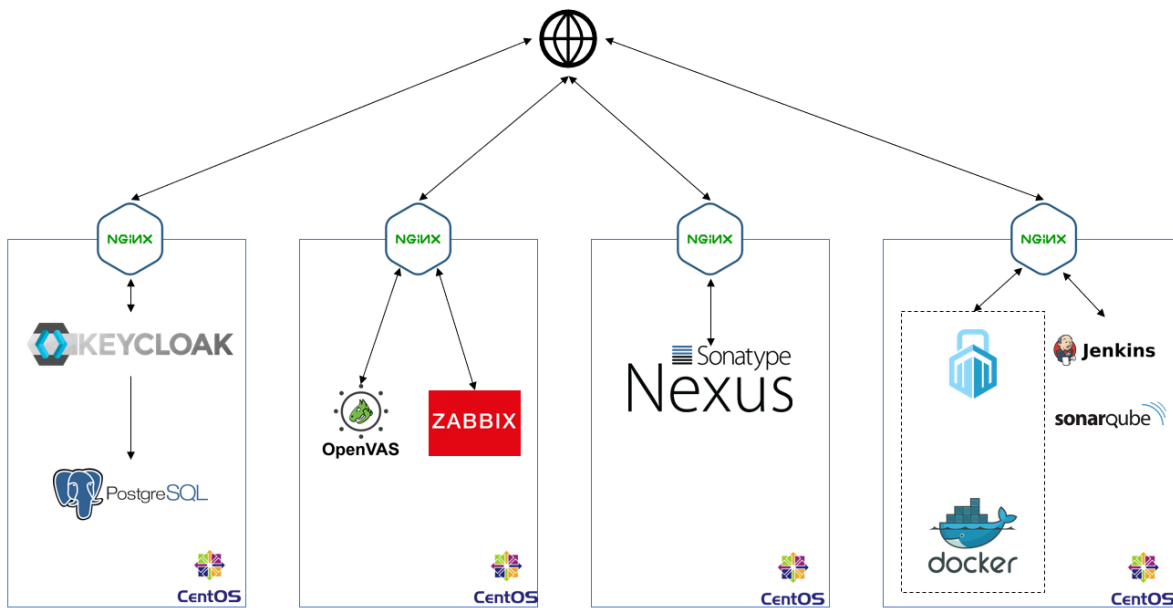


Figure 49 - SMESEC Framework infrastructure deployment

In the diagram above, we can see the different components of the SMESEC Framework core module. On one hand, we have the Keycloak server, deployed on an independent VM. This instance provides a central management for all the users of the SMESEC Framework, grouped by company. This includes authentication and authorization.

In another VM, showed more in detail in Figure 50 below, we have all the components of the SMESEC Framework core itself. It consists on a set of Docker images, managed by a Docker Compose file, and behind a NGINX web server that acts as a reverse proxy, providing TLS termination to the SMESEC Framework. The SMESEC Framework deployment is composed of a Spring Boot application, containerized into a Docker image, and a MongoDB docker image.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	80 of 85
Reference:	D3.9	Dissemination:	PU	Version:	1.0
				Status:	Final

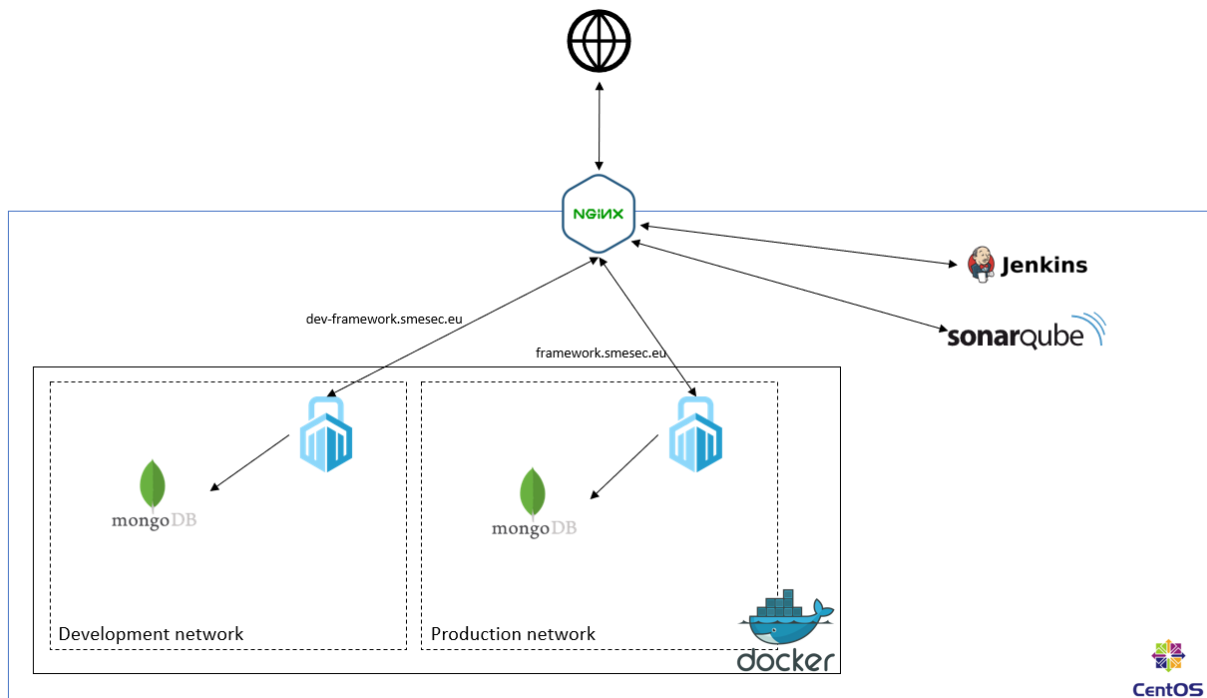


Figure 50: SMESEC Framework Core

At this moment, a single container of both the SMESEC Framework application and the MongoDB database are deployed in a production environment, with appropriate measures to have these containers always available to the users. However, both of these applications can scale-out easily, making the SMESEC Framework ready to grow and provide high availability and fault tolerance in the future, when the customer base is increased.

5.7.2 Updates

Updates of the SMESEC Framework comes in two different ways. For the SaaS deployment option, updates are automatically rolled-out with no necessary interaction of the end user. This is the preferred method, and the easiest one for the user.

For on-premise deployments, upgrades are done via Docker images. There is a public repository that provide a way to download the Docker image for every release of the SMESEC Framework. Changing the running container with the upgraded images will be enough for upgrading the SMESEC Framework to the newest version. Data retention is provided by Docker volumes, that will ensure that data is persisted across container restarts.

In case of a breaking change, or other possible risks while upgrading the SMESEC Framework, special guides will be rolled out to provide support for the upgrade.

5.8 API for external tools

Starting on the second year, we realize that providing a way to include tools that could already exist in SMEs infrastructure into the SMESEC Framework could lead to an important competitive advantage,

Document name:	D3.9 SMESEC Framework Public Report- Final Version	Page:	81 of 85	
Reference:	D3.9	Dissemination:	PU	
	Version:	1.0	Status:	Final

since the security management of the SME would be easier and we would have more data to provide more accurate insights to the SMEs, thus increasing the value of the SMESEC Framework.

In order to provide a way for including content coming from third-party tools without big modifications into our system, and to provide a common way to a wide variety of different tools, each with their own formats, we decided a shared-responsibility model, in which the SMESEC consortium provides a set of tools (from now on, the external API) that eases the process for transforming the data of the tool to the format the SMESEC Framework comprehends. This tool shall be deployed by the tool that is integrated owner. The external API provides many ways to send the data into it, via integrating it directly into your tool, or sending the data to a REST API. The communication is secured with x509 certificates, and the external API provides a simple way to modify the parsing of the input data to transform it to what the SMESEC Framework expects. More detailed information can be seen in the external API documentation [40].

5.9 Functionality, Characteristics and GUI Navigation

All the functionality is described in the D3.3 [27]

5.10 Framework Evaluation

This section will report all activities and results of the security evaluation process as performed by a Third-Party organization in the context of “SMESEC Open Call”. The evaluation phase discovered certain security vulnerabilities and thoroughly assessed the resilience of (i) the SMESEC framework when operating in a standalone manner and (ii) a specific pilot (E-Voting) including additional nodes for providing the overall service functionality. In both cases, the frameworks’ robustness against specific attacks was tested and security recommendation were provided by experts to mitigate the potential impact.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	82 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

6 Conclusions

The scope of this deliverable was to provide a detailed overview of the final prototype of the SMESEC security framework. The document contained a brief analysis of the final design views, together with the design of the user interface, described the various tools and standalone components integrated in the prototype, explained how these entities interact with each other and analysed the operation of the SMESEC security framework prototype together with the specific benefits it introduces. Moreover, the document contained a detailed implementation, evaluation and testing analysis to clearly demonstrate the holistic approach of all participants toward delivering the specific framework.

This document builds upon D3.1 “SMESEC System Design” [1], D3.2 “SMESEC Unified Architecture – First Internal Release” [2] and D3.3 “SMESEC Framework User Manual” [27] and provides a description of changes and enhancements made to the original SMESEC security framework. Treated as a living organism throughout the project, the SMESEC security framework was constantly under development to meet not only requirements gathered in previous deliverables and documented in D3.1 and D3.2 but address real-world issues as well. Such issues were identified through the Pilots or reported by skilled Third-Party personnel during the highly efficient evaluation phase carried out during the Open Call. All this effort led to various iterations, resulted in the specific architecture.

Detailed in this document is the architecture of internal SMESEC component. We presented the core components that deliver orchestration functionalities: SMESEC Hub and SMESEC extensions. And, we presented the architecture of the SMESEC interface. In addition, the enhanced user interface, designed with special attention to user- experience and based on iterative discussions with the use-case partners, is also presented. Finally, an overview of the overall SMESEC prototype functionality, integration, deployment and evaluation process was provided, either directly or via references to corresponding deliverables.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	83 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

References

- [1] SMESEC deliverable 3.1 “SMESEC System Design”, Fady Copty, 2018.
- [2] SMESEC deliverable 3.2 “SMESEC Unified Architecture – First Internal Release”, Fady Copty, 2018.
- [3] Keycloak, <https://www.keycloak.org/>, Last visited: June 1st 2020.
- [4] Firefox, <https://www.mozilla.org/en-US/firefox/>, Last visited: June 1st 2020.
- [5] Chrome, <https://www.google.com/chrome/>, Last visited: June 1st 2020.
- [6] Windows, <https://www.microsoft.com/en-us/windows>, Last visited: June 1st 2020.
- [7] Safari, <https://www.apple.com/lae/safari/>, Last visited: June 1st 2020.
- [8] MacOS, <https://www.apple.com/macos/>, Last visited: June 1st 2020.
- [9] SMESEC Grant Agreement no. 740787 – Annex I Description of the Action (Part B), April 2017.
- [10] MISP data models – MISP core format, MISP taxonomies, <https://www.misp-project.org/datamodels>, Last visited: June 1st 2020.
- [11] Pivotal, <https://pivotal.io/>, Last visited: June 1st 2020.
- [12] Thymeleaf, <https://www.thymeleaf.org/>, Last visited: June 1st 2020.
- [13] Bootstrap, <https://getbootstrap.com/>, Last visited: June 1st 2020.
- [14] Jenkins, <https://jenkins.io/>, Last visited: June 1st 2020.
- [15] Maven, <https://maven.apache.org/>, Last visited: June 1st 2020.
- [16] Nexus Repository, <https://www.sonatype.com/nexus-repository-sonatype>, Last visited: June 1st 2020.
- [17] Zabbix, <https://www.zabbix.com/> Last visited: June 1st 2020.
- [18] OpenID, <https://openid.net/>, Last visited: June 1st 2020.
- [19] Spring-Boot, <https://spring.io/projects/spring-boot>, Last visited: June 1st 2020.
- [20] Spring-Security, <https://spring.io/projects/spring-security>, Last visited: June 1st 2020.
- [21] XSS header protection, <https://docs.spring.io/spring-security/site/docs/5.0.x/reference/html/headers.html#headers-xss-protection>, Last visited: June 1st 2020.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	84 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final

- [22] Java sanitizer, https://www.owasp.org/index.php/OWASP_Java_HTML_Sanitizer_Project, Last visited: June 1st 2020.
- [23] Content security policy, <https://content-security-policy.com/>, Last visited: June 1st 2020.
- [24] Sonarqube, <https://www.sonarqube.org/>, Last visited: June 1st 2020.
- [25] Find security bugs, <https://find-sec-bugs.github.io/>, Last visited: June 1st 2020.
- [26] OWASP dependency check, https://www.owasp.org/index.php/OWASP_Dependency_Check, Last visited: June 1st 2020.
- [27] SMESEC Deliverable 3.3 “SMESEC Framework User Manual”, Benny Zeltser, 2019.
- [28] SMESEC Deliverable 4.9 “Overall Pilot alignment and integration process report”, Jose Francisco Ruiz, 2019.
- [29] SMESEC Deliverable 3.4 “SMESEC products integration on the Unified Architecture”, Ciprian Oprisa, 2019.
- [30] SMESEC Deliverable 5.5 “Open Call Design, Implementation and Results Report”, Manos Athanatos, 2020.
- [31] SMESEC Deliverable 5.2 “System readiness for validation activities”, Noemi Folch, 2019.
- [32] SMESEC Deliverable 3.6 “Final SMESEC Security Awareness and Training Report”, Philippe Cousin, 2020.
- [33] SMESEC Deliverable 2.1 “SMESEC security characteristics description, security and market analysis report”, George Oikonomou, 2017.
- [34] Kippo, <https://en.wikipedia.org/wiki/Kippo>, Last visited: June 1st 2020.
- [35] Krämer, Lukas, et al. "Ampot: Monitoring and defending against amplification ddos attacks." International Symposium on Recent Advances in Intrusion Detection. Springer, Cham, 2015
- [36] Snort, <https://www.snort.org/>, Last visited: June 1st 2020.
- [37] Cowrie, <https://github.com/cowrie/cowrie>, Last visited: June 1st 2020.
- [38] Dioanea, <https://github.com/DinoTools/dioanea>, Last visited: June 1st 2020.
- [39] Bitdefender API, https://download.bitdefender.com/business/API/Bitdefender_GravityZone_On-Premises_APIGuide_enUS.pdf, Last visited: June 1st 2020.
- [40] SMESEC, <https://docs-adapter-tools.smesec.eu/>, Last visited: June 1st 2020.

Document name:	D3.9 SMESEC Framework Public Report- Final Version			Page:	85 of 85		
Reference:	D3.9	Dissemination:	PU	Version:	1.0	Status:	Final