Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

SMESEC

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

D3.1 SMESEC System Design

Document Identification			
Status	Final	**Due Date**	28/02/2018
Version	1.0	**Submission Date**	28/02/2018

Related WP	WP3	**Document Reference**	D3.1
Related Deliverable(s)	----	**Dissemination Level (*)**	PU
Lead Organization	IBM	**Lead Author**	Fady Copty
Contributors	ATOS, BD, WoS, Citrix	**Reviewers**	George Oikonomou, CITRIX Ciprian Oprisa, BitDefender

Keywords:

security, system, design, architecture, integration, WP3, requirements, stakeholder, goals, innovation, use case, protection, defence, management, context, concept, pattern, composition, interface, rationale, sequence.

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.

(*) 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
Fady Copty	IBM
Rodrigo Diaz Rodriguez	ATOS
Ciprian Oprisa	BD
Francisco Hernandez	WoS
George Oikonomou	Citrix

Document History			
Version	Date	Change editors	Changes
0.1	12/01/2018	Fady Copty (IBM)	Initial version of table of contents and basic draft of design views
0.2	19/01/2018	Fady Copty (IBM)	First draft sections 3 and 5 (missing 5.1.5, 5.3)
	19/01/2018	Rodrigo Diaz Rodriguez (ATOS)	First draft section 1.
0.3	22/01/2018	Ciprian Oprisa (BD)	First draft of section 6.2
	22/01/2018	Francisco Hernandez (WoS)	First draft of section 4
	22/01/2018	Fady Copty (IBM)	First draft sections 5.1.5
	23/01/2018	George Oikonomou (Citrix)	First draft of section 6.1
	23/01/2018	Fady Copty (IBM)	First draft sections 5.3
0.4	26/01/018	George Oikonomou (Citrix)	First draft sections 2
	31/01/2018	Francisco Hernandez (WoS)	Second draft of section 4
	31/01/2018	George Oikonomou (Citrix)	Second draft sections 2
	01/02/2018	Fady Copty (IBM)	Second draft section 5
0.5	02/02/2018	Rodrigo Diaz Rodriguez (ATOS)	Second draft section 1, First draft section 2.1.3

	02/02/2018	Fady Copty (IBM)	First draft executive summary, and section 7
	05/02/2018	Fady Copty (IBM)	Reorder sections 3,4. Reorder subsections of section 2
	06/02/2018	Fady Copty (IBM)	Add acronyms, references, figure and table lists
0.6	08/02/2018	Ciprian Oprisa (BD)	First draft of section 6.3
	08/02/2018	Fady Copty (IBM)	Minor fixes
0.7	12/02/2018	George Oikonomou (CITRIX)	First review comments
0.8	16/2/2018	Fady Copty (IBM)	Resolve comments
0.8.3	26/02/2018	Fady Copty (IBM)	Resolve review comments
1.0	28/02/2018	ATOS	FINAL VERSION TO BE SUBMITTED

Table of Contents

List of Tables

List of Figures

List of Acronyms

Abbreviation / acronym	Description
API	Application Programming Interface
AST	Application Security Testing
AV	Anti-Virus
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
MiTM	Man-in-the-Middle

Abbreviation / acronym	Description
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
VDI	Virtual desktop infrastructure
VM	virtual machine
VPN	Virtual Private Network
WAF	Web Application Firewall
WP	Work Package
XML	Extensible Mark-up Language

Executive Summary

The aim of the present document is to describe the design of the SMESEC Framework with the objective in mind of creating an appropriate prototype. The main challenges addressed in this document are identifying the SMESEC Framework architectural requirements, designing an architecture that meets those requirements, and validating that the integration of SMESEC tools into this design of SMESEC Framework is feasible.

The SMESEC use case requirements were produced in WP2 and a preliminary business study was conducted in WP6. The stakeholder's concerns and requirements stemming up from these work packages were identified, gathered and prioritised as well as some out-of-scope requirements.

The SMESEC Framework was designed to meet those requirements. The design is detailed in five design views: Context view, concept view, pattern view, composition view and interface view. A deployment view is discussed in this document, but it will be developed only in later stages of the SMESEC project.

Last, the integration of SMESEC tools into this design of the SMESEC Framework was validated. The integration of SMESEC tools was found feasible. Future integration of new tools and future innovations into the SMESEC Framework were found feasible as well.

This document will serve as basis for "SMESEC Unified Architecture" (D3.2), and "SMESEC Unified Architecture and the Initial Version of the SMESEC Framework prototype" (D3.3).

1 Introduction

1.1 Purpose of the document

The aim of the present document is to describe the design of the SMESEC security framework with the objective in mind of creating an appropriate prototype. To achieve this objective, the architectural requirements coming from the four pilots (e-Voting, Smart Cities, Industrial Services and Smart Grids) together with the information coming from the different tools provided by the technical partners are gathered. These requirements will be analysed and translated into functional and non-functional features to be incorporated in the first release of the intended framework architecture (D3.2). Additionally, the feedback obtained in the preliminary business study conducted in WP6 and documented in section 3 will be considered with the objective of ensuring that the proposed design fulfils the market needs and demands.

1.2 Relation to other project work

As described in the DoA [12], this document will provide the innovation roadmap to be used in structuring of the unified SMESEC security framework. This document considers as an input the specifications and requirements produced in WP2 and the preliminary business study conducted in WP6. These inputs will serve us to design of the SMESEC Framework that will be needed for the architectural design serving T3.2 ("Enhance SMESEC market products with the latest research innovations").

The definition of the SMESEC Framework will continue with the release of the first version of the SMESEC Unified Architecture (D3.2) in M18 providing system level functionalities for the proposed SMESEC Framework architecture and conclude with the release of the final version of the SMESEC Unified Architecture and the Initial Version of the SMESEC Framework prototype (D3.3) in M24.

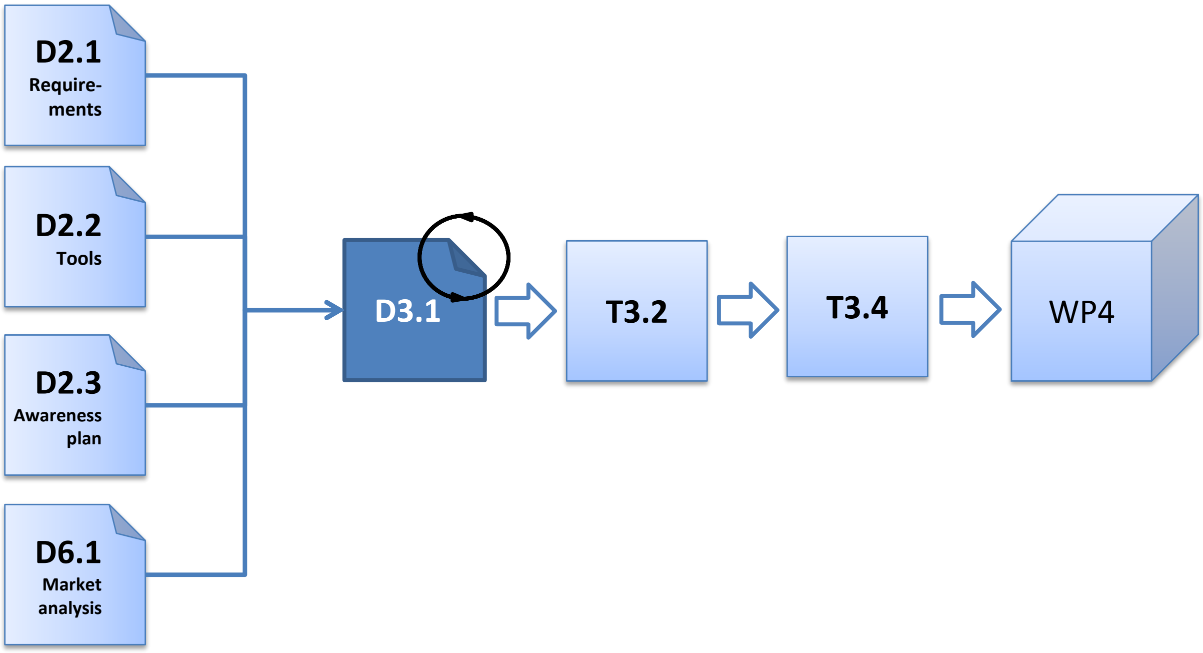

Figure 1: High-level view on the methodology for designing the SMESEC Framework

Figure 1 shows a high-level diagram of the process we followed for designing, refining and enhancing the SMESEC Framework architecture.

1.3 Structure of the document

This document is structured in 7 major chapters:

- **Chapter 1** presents the main objectives of the deliverable and describes the following sections.
- **Chapter 2** introduces the goals of the SMESEC Framework
- **Chapter 3** describes the innovations provided by the SMESEC Framework and the different solutions composing the platform.
- **Chapter 4** summarizes the main requirements to be fulfilled by the SMESEC Framework extracted from WP2 deliverables.
- **Chapter 5** provides the SMESEC Framework design.
- **Chapter 6** presents information to guide the integration activities of the SMESEC tools in the proposed design presented in previous section.
- **Chapter 7** describes the conclusions of the work presented in this deliverable, focusing on the SMESEC Framework design, the process we followed for its definition and the future work we plan for the refinement and enhancement of the architecture and related components.

2 Goals of SMESEC Framework

SMESEC aims at providing a unified security framework for Small Medium Enterprises (SME). SME's are one of the most important drivers for innovation, but they often tend to not properly plan their cybersecurity defence, either by underestimating the risks and consequences of cyberattacks, or by not being able to keep pace with the progress in this ever-evolving field. New threats appear on a daily basis and SMEs are usually unready to protect their IT assets and therefore the business continuity.

The main goal of SMESEC is to identify what are the needs from the SME perspective and translate them into requirements for a unified framework, which will eventually consist of the SMESEC partners' contributed products. The products themselves cover a wide range of security market segments and it is expected that the unification will bring even greater added value to the products and the framework.

In the first phase, the SMESEC use case partners have been asked to provide their valuable input as to the security needs of their organizations, prioritize them and explain how they implement security today. The findings have been described in deliverable D2.1[2]. In the next sections, an analysis of these requirements will follow as they are the key drivers in the system design that will follow.

2.1 SMESEC stakeholders' concerns

The SMESEC stakeholders can be categorized into three groups:

- The SMESEC use-case partners
- The SMESEC tool partners
- The EU commission

The concerns of each stakeholder group are described below.

2.1.1 SMESEC use cases

The SMESEC pilot use cases have identified a number of high-level requirements, which refer mostly to the desired service level that they want to provide to their end users. These are shown in Table 1. It should be noted that the list depicts only those that have been identified by the use case partners, and it is well anticipated that in the next phases of the SMESEC project, more may appear and must be addressed.

Table 1: SMESEC use case needs

Requirement	Description
Availability	The systems and services should be available uninterruptedly to their end users. Any disruption has potentially consequences to the business.

Requirement	Description
Usability	The security framework should be usable by the SME system administrators. Complexity in administration can prevent the effectiveness of the provided security features.
Privacy	One of the highest priority requirements is the protection of privacy, both for the business data and the end-users. Breach of privacy has impact on the trust of end-users to the SME.
Cost	Any solution should be affordable. Cost is often a factor that hinders SMEs to deploy a cyber-protection solution.
Alerting	Alerts should be configurable and in general, monitoring should be complete and thorough. SMEs are particularly interested in getting notified in near real-time for possible threats in their infrastructure.
System integrity	Specifically for systems, the integrity is critical for protecting the services that run on top.
Confidentiality	All information exchanged should remain private (or anonymised).
Non-repudiation	Any system/service should be accessed only by the designated staff and any unauthorised attempt should be reported.
Authentication	Ability to authenticate users effectively when accessing specific resources
Scalability	A strong requirement for a security system is to cover the potential growth of the SME with more users, locations, systems, without jeopardising the continuity of the business.

Regarding cyber-security threats and protection from them, the SMESEC use case partners have identified what they would consider the most critical. These are shown in Table 2.

Table 2: SMESEC use case identified threats

Concern	Description
Code injection	A common type of attack to lots of applications (and mainly web), yet still very important to protect against. Session hijacking is also another potential threat in this category.
DDoS attacks	Another very common type of attack that can render the infrastructure useless, have a great impact on the business and the brand, and frustration to the end users.
Man-in-the-Middle attacks	This type of attacks can breach the confidentiality and privacy of the information and the framework should be able to detect and mitigate them.
Malware	Malware is a raising threat which is getting more and more sophisticated. Similar to anti-virus protection, SMEs recognize the need to protect from this type of attacks.
Database protection	Back-end databases are often a common target, either by reading sensitive data (personal, financial, etc.) or by writing erroneous information. Attacks

Concern	Description
	should be intercepted early, and away from databases.
Virtualization and cloud security	As more and more businesses choose to move to virtualization or cloud to reduce their costs, the assets hosted on a hypervisor or a public cloud infrastructure (or the communication in-between) can fall prey to hackers.
Firewall protection	A security framework should include at least a basic firewall protection, with more sophisticated firewall solutions being highly desirable.
Strong authentication	In both systems and communications, strong authentication is required for access or transmission respectively.

In Table 3 appears the translation of the above requirements and threats of Table 1 and Table 2 to some drivers of the unified framework.

Table 3: SMESEC use case requirements

Unified Framework Requirement	Description
Transparent experience	End-users of the unified framework should not be aware of the internal complexity of the tool. The experience should be smooth for all users and administrators, assuming little or no IT expertise. **Related use case requirements:** Availability, Usability
Easy deployment	All deployment related interactions (installation, upgrades, etc.) should be as smooth as possible without assuming a high technical expertise on behalf of the system administrator. **Related use case requirements:** Usability
Common attack defence	The unified framework should be able to protect against the most common attacks in an effective and timely manner. **Related use case threats:** Code injection, DDoS attacks, MitM attacks, firewall protection **Related use case requirements:** System integrity
Endpoint protection	Endpoints (from user equipment to IoT sensors) should be properly protected to avoid cyber-threats from the trusted side of the network. **Related use case threats:** Malware **Related use case requirements:** System integrity, Confidentiality
Cloud/Hypervisor security	More enterprises turn to virtualization solutions and public/private/hybrid clouds to reduce their IT costs, and these assets (IaaS, PaaS, SaaS) should be

Unified Framework Requirement	Description
	appropriately protected. **Related use case threats:** Virtualization and Cloud security
Flexible alerting	Alerting for real-time threats, but also for the adherence to a long-term protection strategy should be easy to configure and view. **Related use case requirements:** Alerting
Affordability	Cost is always an important (if not the primary) driver for the adoption of a security solution. All services should come at an affordable price for the provided services. **Related use case requirements:** Cost
Strong privacy and authentication	Privacy concerns are always present in any security solution, so the framework should be able to handle and enforce access control (Authentication, Authorization, Accounting) but also ensure the privacy of all stored and processed data. **Related use case requirements:** Authentication, Non-repudiation, Privacy **Related use case threats:** Strong authentication
Security event processing	A multitude of security events are produced and need to be processed and prioritized according to their importance for the overall infrastructure. The unified framework should be able to provide configurable and comprehensive event workflows. **Related use case requirements:** Alerting, System integrity
Scalability	Can be considered a subclass of deployment, but as a business objective, the unified framework should be able to painlessly scale up or down following the size of the business workloads. **Related use case requirements:** Scalability, Cost, Availability

2.1.1.1 Prioritisation of concerns

In terms of priorities of the threats (Table 2), the SMESEC use case partners have prioritized them as shown in Table 4.

Table 4: SMESEC pilots protection requirements with priorities

Protect / Protect against	SCYTL	UOP	WOS	GRIDP
Web application servers	1	1	4	4
Database servers	2			
Network traffic	3	5		
Web servers	4			
Email servers		3		
DDoS		1	5	1
Access abuse			2	
Software misuse			1	
Zero-day attacks			6	
Code injection			8	2
Man-in-the-Middle attacks			3	3

The conclusions from this prioritisation are:

- IoT use case partners focus on lower-level aspects, including physical access and attempts to tamper the device, protection at byte code level, and man-in-the-middle attacks (as devices are dispersed in large areas)
- Enterprise use cases pay more attention to web-nature applications and protection of application servers, databases and email.
- In all cases denial-of-service attacks to frontend applications seem equally important too as this is usually the interface to the end-users.

As part of extending the methodology of assessing the cybersecurity risk within an SME, use case partners had to assess themselves against OWASP Top-10 challenges 0 and by using the CYSFAM maturity model [13]. The results from this research are summarized in section 3 of D2.1 [2] and section 3.2 of D2.3 [4] respectively.

2.1.2 SMESEC product partners

The SMESEC product partners are well aware of the challenges of designing and creating a unified framework, and the use case requirements validate them. The security market is currently very wide with most key players having deep vertical solutions. Creating a unified framework that will expand horizontally across multiple market segments requires careful design and clear interfaces.

In the process of identifying what are the potential benefits for the integration effort, SMESEC product partners have identified the following high-level requirements for their products:

- Integrating with other products: It is well understood that the products will need to be able to exchange information with each other. The added value of unifying the products exceeds by far the sum of the individual product values, so it is critical to identify which are those connections that will maximize the impact.

- Getting feedback from SMEs: Having access to a wider customer base, the product partners can assess the feedback and drive the development towards the customer needs.
- Extending product capabilities: Each product falls into one or more security market segments. Enriching products with more features as the result of the integration is a key advantage for the product partners.

2.1.2.1 Security market analysis

An extensive research of the security market landscape from a technical point of view has been conducted in Work Package 2 (section 4 of D2.1 [2]), and from a business perspective in D6.1 [5] The key security market segments as of today have been identified as presented in Table 5, along with the related contributed products.

Table 5: Security market segments and related SMESEC products in each

Feature	Description	SMESEC product
Encryption	Includes products able of encrypting and protecting sensitive data, centrally or at endpoints.	-
Governance, Risk Management and Compliance	The products in this category handle the workflows that ensure that the information is handled properly, adheres to laws and regulations, and can identify/predict/react to possible risks.	FHNW CYSEC
Security Information and Event Management	SIEM products handle the security information flow (logs, events, etc.), and data aggregation and correlation with other sources in order to provide meaningful security insights.	ATOS XL-SIEM
Data Loss Prevention	Products that enforce a set of security controls that prevents information from being disclosed to unauthorized users.	-
Unified Threat Management	These solutions integrate seamlessly a variety of other security products (e.g. firewall, antivirus)	The unified SMESEC Framework
Intrusion Detection and Prevention Systems	Systems that are able to identify and mitigate intrusion attacks from unauthorized users.	FORTH EWIS
Distributed Denial-of-Service defence	Systems that are able to deter DDoS attacks in most common protocols before the systems become unresponsive.	CITRIX AppFirewall, FORTH EWIS
Business continuity/ Disaster recovery	Products that can handle the crisis management plans, continuously analysing the risks ensuring that the business processes are followed, and disaster recovery activities are initiated if needed.	-

Feature	Description	SMESEC product
Web Application Firewall	Firewalls with specialization on protecting web applications and mitigating HTTP application-layer attacks.	CITRIX AppFirewall
Secure Web Gateways (SWG)	The SWG products can inspect and filter incoming and outgoing web traffic for malicious content, even if it is encrypted.	Citrix SWG, BitDefender GravityZone
Endpoint Protection Platforms	Products that focus on the protection of endpoints, usually provided services like anti-virus, personal firewall, application control, etc.	BitDefender GravityZone
Application Security Testing	A proactive process of testing application for potential vulnerabilities that can be exploited by malicious users.	EGM TaaS, IBM AngelEye, IBM ExpliSAT
Security Awareness and Training	The products in this category take into account the human factor and intend to ensure that they are well informed of the risks, and able to identify an attack an early stage and take the appropriate preventive action.	-
Deception Technology	Products with sophisticated methods of deploying decoys in parts of the infrastructure in order to attract attackers there and protect the sensitive services.	IBM AntiROP
Endpoint Detection and Response	An evolvement of the Endpoint Platform Protection, with more sophisticated methods of detecting and mitigating attacks.	BitDedender GravityZone
Cloud Access Security Brokers	Ensures a seamless access between the on-prem and cloud resources of an enterprise (including SaaS applications), enforcing the corporate policies.	-
User Entity Behaviour Analytics	An emerging field of security analytics with emphasis on the behaviour patterns that appear in the data coming from various sources.	-
Identity and Access Management	Products that offer centralized control of access for a multitude of services inside an enterprise. Enables seamless integration of third-party SaaS applications, centralized access monitoring and notifications in case of suspicious authentication events.	-

2.1.3 European Commission

The SMESEC project overall objectives align to the terms identified in H2020 Call Topic DS-02-2016 [6] Cyber Security for SMEs, local public administration and Individuals, as it is described in Description of Action Annex I Part B. As a consequence, the SMESEC Framework SMESEC Framework should contribute to mitigating the following concerns:

- **High degree of usability and automation:** taking into account SMEs are the primary users of the framework, an easy to use system is a must to first, attract the potential end-users to the tools; second, facilitate these getting used to operate with the system fast; and third, ensure a wider and long-term adoption by incorporating the system to their individual and organization procedures.
- **Provide an adequate degree of cyber situational awareness and control for end-users:** an easy to use framework does not necessary implies presenting to the end user low quality cybersecurity information, limiting their scope or reducing arbitrarily the level of details, so the overall situational awareness offered is degraded. On the other hand, the SMESEC Framework must ensure that the end-user remains well aware of the cybersecurity status of their systems at any time, and have the means to take action if needed.
- **Incorporate the "human factor" (focusing on psychological and behavioural factors) in the design process:** a crucial factor to provide a security framework that adapts to the end-user needs, resources, environment and level of expertise is to analyse their behaviour, attitude and perception towards the security solutions. Based on that, custom configuration of the framework tools and security training courses adapted to the specific user profile can be provided. This will guarantee a higher rate of success in the ultimate objective of enhancing the cybersecurity education and level of protection of SMEs.
- **Follow existing relevant best practises and adoption of standards, tailored to SMEs and individuals:** to foster a wider and faster adoption of the SMESEC Framework, it is recommended to adhere to existing security models and standards such as ISFAM model, ISO27K, CISSP, Standard of Good Practice of the Information Security Forum (ISF), ISO-light, etc. This will also contribute to facilitate integration of third party security tools into the SMESEC Framework and thus, secure a successful and sustainable path to market.

3 Innovation of the SMESEC Framework

SMESEC project aims to create a high-quality security framework which can easily make available the cyber-security levels of SMEs, providing robust, affordable and easy to apply solutions. To achieve this final goal, innovation is a cornerstone that cannot be neglected, and it must lead to basic directives that will help to implement the work and objectives foreseen in WP3.

In the frame of the project, innovation is not seen as a mere research activity to enhance the state-of-the-art technologies provided by the partners, but to respond to the technical and business requirements preliminary identified in WP2 and WP6, taking into account the particular needs of each use case partner.

As result of the innovation process, some of the SMESEC products and components are expected to touch new market segments by the end of the actual implementation, while bearing in mind the existing market security products and the SMEs constraints to adopt new technologies (i.e. budget, human power, etc). Besides, it should not be forgotten the real added-value of SMESEC concept is the integration of different solutions working in an orchestral approach. This is, per se, a major innovative item since the information crossover can bring improved or new functionalities to the existing technologies and solutions. For this reason, interconnection is the main motto of the innovation activities in SMESEC.

In view of the above, and with a firm determination of working to get attainable results, the innovation analysis and priorities fixed at consortium level should prioritize the use of resources to develop some of the desired technical features already identified at deliverable D2.1[2]. The final objective is therefore, attaining, at any rate, a competitive advantage of the final SMESEC Framework released by the end of the project.

In this sense, five key differentiating criteria have been considered to guide the decision-making process that will shape the technical work, and as a result, gain a competitive advantage to the consortium over competitors. These criteria are:

- **Simplicity:** innovation items to be developed in the project should decrease the usual complexity level of security tools, making them more attractive for adoption by the SMEs. The complexity term basically refers to usability, but also the installation and updating requirements of these tools.

- **Protection:** SMESEC solutions should provide better or at least comparable level of cyber-security protection to the offered by the existing solutions in the market.

- **Cost-effectiveness:** since one of the main entrance barrier of cyber-security solutions in the SMEs ecosystem are the budget constraints, any incremental innovation must keep costs low.

- **Training and awareness:** apart from the technical aspects, SMESEC wants to evangelize the importance of cyber-security protection among SMEs. Innovation road-mapping will also consider the development of supporting material to attain this non-technical objective.

- **Interconnection:** as said before, the high-priority for SMESEC is the orchestration of the different products and solutions, favoring the data crossover and the validation of new functionalities.

Considering the aforementioned criteria classification, the potential technical improvements identified in section 7 of deliverable D2.1[2] have been preliminary sorted out and will be later prioritized seeking to obtain a robust architecture of the final SMESEC Framework in line with the provisions of the grant agreement, the suitability for the use cases in the project, and the expected evolvement of the security market in the coming years.

It goes without saying that the follow-up and updating of the road-mapping of the innovation activity is a live process that may need continuous realignments to adapt it to the implementation and market reality, and the potential problems that may arise.

The Innovation Committee (IC) of SMESEC responsible for continuously updating Table 6, and it will provide the necessary inputs to the governing bodies of SMESEC to keep the technical development aligned with the criteria outlined in the grant agreement. This process will be iterative and will restrict the final shaping of the non-functional requirements identified in section 4 for the entire architecture.

Figure 2 provides a first visualization of the SMESEC products position in the security market landscape at present time (red), and the expected coverage (yellow) that the integrated framework might have by the end of the project: the added benefit should come from the technical evolution of the individual products but mainly from the integration efforts and the new functionalities. It goes without saying that the future position scenario is subjective, and it basically represents the first expectations of the consortium, which will go through a refinement process during the project implementation.

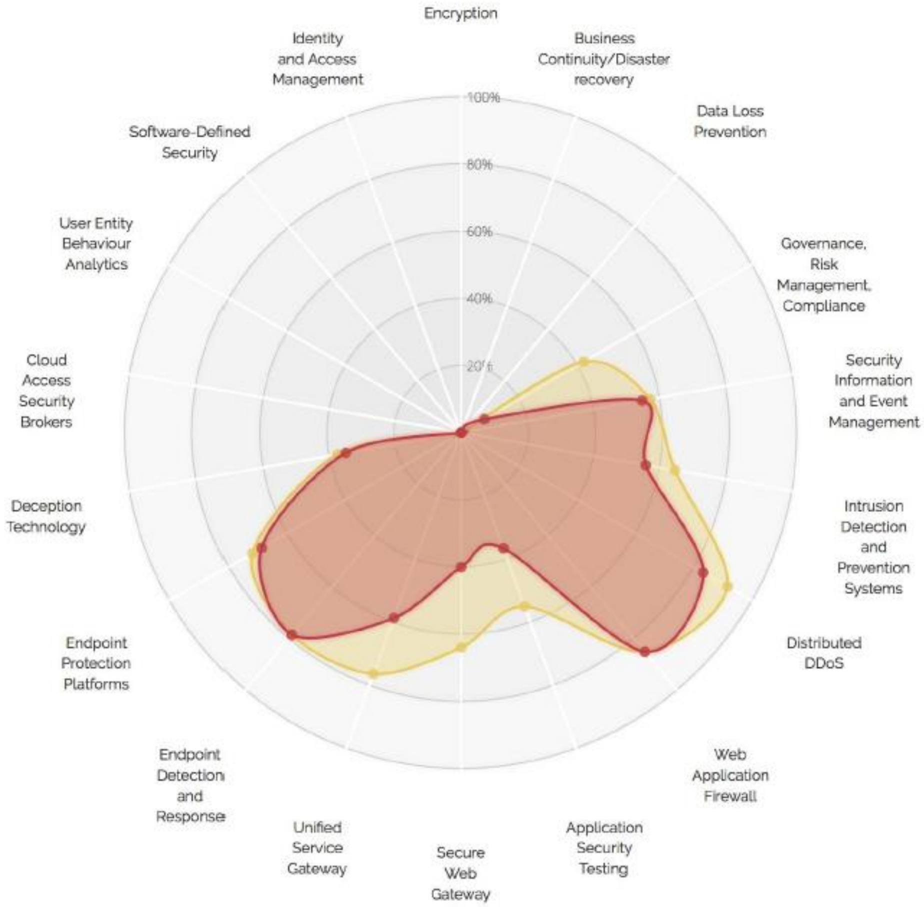

Figure 2: Visualization of SMESEC products in the security market

In the Table 6 below, the main innovation items per product and market segment which will provide by the partners to SMESEC are identified. When possible, their matching with the technical extensions per product already identified in the WP2 is indicated. This preliminary table will be prioritized considering the above discussed criteria, and the output will be used as one of the main drivers in the definition process of the final SMESEC architecture.

Table 6: Future innovations

Partner	Product	Technological Areas	Market	Innovation ID	Technical Extension ID	Innovation Name	Description
ATOS	Risk Assessment Engine	Security Information and Event Management	SIEM	ID1	ATOS.PE01	Overview of indicators about cybersecurity threats and attacks	New tool
ATOS	IoT-SIEM	Security Information and Event Management	SIEM	ID2	ATOS.PE02	Extension of SIEM to IoT domain	Extension current product
BD		Endpoint Protection Platform	EPP	ID3	BD.PE01	Integration of Gravity Zone with SIEM	Interconnection

Partner	Product	Technological Areas	Market	Innovation ID	Technical Extension ID	Innovation Name	Description
CITRIX	NetScaler AppFirewall	Firewall	WAF	ID4	CITRIX.PE01	Deploy as-a-service	New service
CITRIX	NetScaler AppFirewall	Firewall	WAF	ID5	CITRIX.PE02	DDoS detection: optimize techniques by leveraging SMESEC Framework data	Interconnection
CITRIX	NetScaler AppFirewall	Firewall	WAF	ID6	N/A	Easy-to-implement instructions for the product	Extension current product
CITRIX	NetScaler Gateway	VPN	EPP, USG	ID7	CITRIX.PE03	Deploy as-a-service	New service
CITRIX	NetScaler Gateway	VPN	EPP, USG	ID8	CITRIX.PE04	Integration with SIEM solutions	Interconnection
CITRIX	NetScaler Gateway	VPN	EPP, USG	ID9	N/A	Offer simplified training material for deployments	Extension current product
CITRIX	NetScaler Secure Web Gateway	SSL interception, URL filtering	USG, SWG	ID10	CITRIX.PE08	Support more connectors and interfaces to other products	Interconnection
CITRIX	NetScaler Secure Web Gateway	SSL interception, URL filtering	USG, SWG	ID11	CITRIX.PE09	Sophisticated policies to integrate with multiple features	New tool
CITRIX	NetScaler Secure Web Gateway	SSL interception, URL filtering	USG, SWG	ID12	N/A	Produce and offer some simplified training videos	New tool
CITRIX	NetScaler Secure Web Gateway	SSL interception, URL filtering	USG, SWG	ID13	CITRIX.PE05	Anti-malware protection (SWG)	New function
CITRIX	NetScaler Secure Web Gateway	SSL interception, URL filtering	USG, SWG	ID14	CITRIX.PE06	Anti-bot protection (SWG)	New function
CITRIX	NetScaler Secure Web Gateway	SSL interception, URL filtering	USG, SWG	ID15	CITRIX.PE07	Email security	New function
EGM	EGM-TaaS	Testing	AST	ID16	EGM.PE06	Tests for known IoT vulnerabilities with the full test suites	Extension current product
EGM	EGM-TaaS	Testing	AST	ID17	EGM.PE01	Test for fuzzing and brute force attacks	Extension current product
EGM	EGM-TaaS	Testing	AST	ID18	EGM.PE02	Detect privileged access related vulnerabilities, linked to IoT systems	Extension current product
EGM	EGM-TaaS	Testing	AST	ID19	EGM.PE03	Detect OWASP top-10, WASC & SANS top-25 vulnerabilities	Extension current product
EGM	EGM-TaaS	Testing	AST	ID20	EGM.PE04	Detects applications DoS vulnerabilities	Extension current product
EGM	EGM-TaaS	Testing	AST	ID21	EGM.PE05	Integrate with bug tracking tools	Interconnection
FHNW	CYNET	Risk Assessment	GRC	ID22	FHNW.PE01	Support commonly used policy templates	New function
FHNW	CYNET	Risk Assessment	GRC	ID23	FHNW.PE02	Risk register	New function
FHNW	CYNET	Risk Assessment	GRC	ID24	FHNW.PE03	Support for Risk Frameworks	New function
FHNW	CYNET	Risk Assessment	GRC	ID25	FHNW.PE04	KRI (Key Risk Indicator) library	New function
FHNW	CYNET	Risk Assessment	GRC	ID26	FHNW.PE05	Risk Assessment Questionnaires	New tool
FHNW	CYNET	Risk Assessment	GRC	ID27	FHNW.PE06	Risk-based scoping	New function
FHNW	CYNET	Risk Assessment	GRC	ID28	FHNW.PE07	Workpaper management	New function
FHNW	CYNET	Risk Assessment	GRC	ID29	FHNW.PE08	Audit Calendar Management	New function

Partner	Product	Technological Areas	Market	Innovation ID	Technical Extension ID	Innovation Name	Description
FHNW	CYNET	Risk Assessment	GRC	ID30	FHNW.PE09	Integration with 3rd party tools (patch management, vulnerability assessment, etc.) through an API definition with SMESEC partners' tools	Interconnection
FHNW	CYNET	Risk Assessment	GRC	ID31	FHNW.PE10	Data aggregation from multiple sources (SIEM, DLP service desk, etc.)	Interconnection
FHNW	CYNET	Risk Assessment	GRC	ID32	FHNW.PE11	Federated architecture	Consulting
FHNW	CYNET	Risk Assessment	GRC	ID33	FHNW.PE12	Custom role-based dashboards	New methodology
FORTH	EWIS	Early Warning Intrusion Detection System	IDS/IPS	ID34	FORTH.PE01?	Interconnection with other tools	Interconnection
FORTH	Cloud-based IDS	Early Warning Intrusion Detection System	IDS/IPS	ID35	FORTH.PE04	Defence of web and applications on the Cloud	New methodology
FORTH	Cloud-based IDS	Early Warning Intrusion Detection System	IDS/IPS	ID36	FORTH.PE05	GPU for pattern matching	New methodology
FORTH	Cloud-based IDS	Early Warning Intrusion Detection System	IDS/IPS	ID37	FORTH.PE02	IPS profiles to activate/deactivate protections based on severity, protocols, confidence interval, etc.	New methodology
FORTH	Cloud-based IDS	Early Warning Intrusion Detection System	IDS/IPS	ID38	FORTH.PE03	Prioritize and send alerts to users	New methodology
IBM	AngelEye	Virtual Patching	AST	ID39	IBM.PE06	Automatic virtual patching tool - Learning of fuzz testing data	New tool
IBM	AngelEye	Virtual Patching	AST	ID40	IBM.PE07	Automatic virtual patching tool - automatic updating	New tool
IBM	AngelEye	Virtual Patching	AST	ID41	IBM.PE02	Integration with WAF vendors (AngelEye)	Interconnection
IBM	AngelEye	Virtual Patching	AST	ID42	IBM.PE03	Integration with MDM/EMM vendors (AngelEye)	Interconnection
IBM	Anti-ROP	IBM Anti-ROP compiler plugin	DT	ID43	IBM.PE08	Moving target defense	New methodology
IBM	Anti-ROP	IBM Anti-ROP compiler plugin	DT	ID44	IBM.PE04	Identify attacks without known attack patterns or signatures (AntiROP)	New methodology
IBM	Anti-ROP	IBM Anti-ROP compiler plugin	DT	ID45	IBM.PE05	Integration with MDM/EMM vendors (AntiROP)	Interconnection
IBM	ExpliSAT	Model checking	AST	ID46	IBM.PE01	Fully automated testing (ExpliSAT)	New methodology

4 Requirements fulfilled by the SMESEC Framework

The functional requirements identified for SMESEC Framework fall into two main categories: threat defence and security management. The threat defence includes the following functional requirements:

- Protect the SME infrastructure from adversary's attacks.
- Detect adversary's attacks on the SME infrastructure.
- Monitor the SME infrastructure.
- Alert when an attack on the SME infrastructure is detected.
- Respond to adversary's attacks on the SME infrastructure.
- Discover vulnerability in the SME infrastructure.

The security management requirements include:

- Provide assessment of security level.
- Provide suggestions for improving security level.
- Provide evaluation of security risk and consequences.
- Provide assessment of criticality.

The non-functional requirements identified for the SMESEC Framework fall into the following categories:

- Modularity of Deployment – The SMESEC Framework must allow modular deployment of SMESEC security solutions at the SME's system. This requirement stems up from security management consideration, and from the need for high flexibility in the overhead cost for SME's caused by the deployment of security solutions.
- Modularity of Development – The SMESEC Framework must allow modular development of SMESEC tools. This stems up from the nature of the project that integrates various security tools into this framework and should allow future development of innovative orchestration, integration of new tools into the framework, and external integration to other tools.
- Confidentiality – The SMESEC Framework must allow governance of SME data and allow SME to decide the level of confidentiality of the data collected by the SMESEC Framework and tools.
- Usability – The SMESEC Framework should meet high usability standards and offer a unified interface for all tools included in the SMESEC Framework.
- Scalability – The SMESEC Framework must allow load scalability, multi-tenancy, and easy expansion of the framework.

The following functional requirements were identified as out of SMESEC Framework scope:

- Enforcing privacy regulation – The SMESEC Framework will comply with privacy regulations, but it will not serve as a tool for enforcing privacy compliance of the SME's systems.

- Incident response – The SMESEC Framework does not support incident response functionalities like handling a breach and managing the consequences for minimizing the impact, but it should be easily extendable.

Cloud deployment was identified as one of the important non-functional requirements for the SMESEC Framework. It was decided not to include it the current requirements list, and to leave it for later developments of the SMESEC Framework. More details on this topic are provided in section 5.1.6.

5 Design of SMESEC Framework

In this chapter we follow the IEEE Standard 1016-2009 Software Design Description[1]. This is a standard that describes software designs and establishes the information content and organization of a software design description (SDD). This standard is used to describe, organize and prepare the information content needed of a SDD (software design description). It was chosen as a basis for the design section to help ensure the design descriptions are complete, consistent, well organized, easy to communicate and appropriate for recording our decisions.

5.1 SMESEC Framework design views

This section describes: context view, concept view, pattern view, composition view, and discusses deployment view.

5.1.1 Context view

The context view is used to identify actors, services and identify system boundary. The following UML images show the actors and services for our SMESEC Framework. The actors identified for the required use cases are either IT system administrators of the SME's or a specialized Chief Information Security Officer (CISO) of the SME's system. The SME's system is composed of various types of devices and of users who interact with those devices.

The actors introduce two categories of requirements: threat defence, and security management. Threat defence can be expressed in the form of monitoring a system for security events, protecting the system from adversary attacks, or hardening the system through user training and application hardening techniques like testing and patching. Security management can be expressed in the form of continues security assessment of a system against risks and regulations, or as security configuration for the SME's needs.

Figure 3 describes the threat defence use cases:

- An actor requests monitoring of an SME's devices and users; the SMESEC Framework returns alerts on possible attacks, and alerts on publicly known vulnerabilities in the SME's system.
- An actor requests protection and reporting of known attacks; the SMESEC Framework protects the SME's system and report events to the actor.
- An actor requests discovering vulnerabilities in the SME's software; the SMESEC Framework searches for vulnerabilities and reports back to the actor.
- An actor requests training of the SME's employees; the SMESEC Framework provides training and reports back results.
- An actor requires randomization of software; the SMESEC Framework provides a number of uniquely randomized copies of the software.

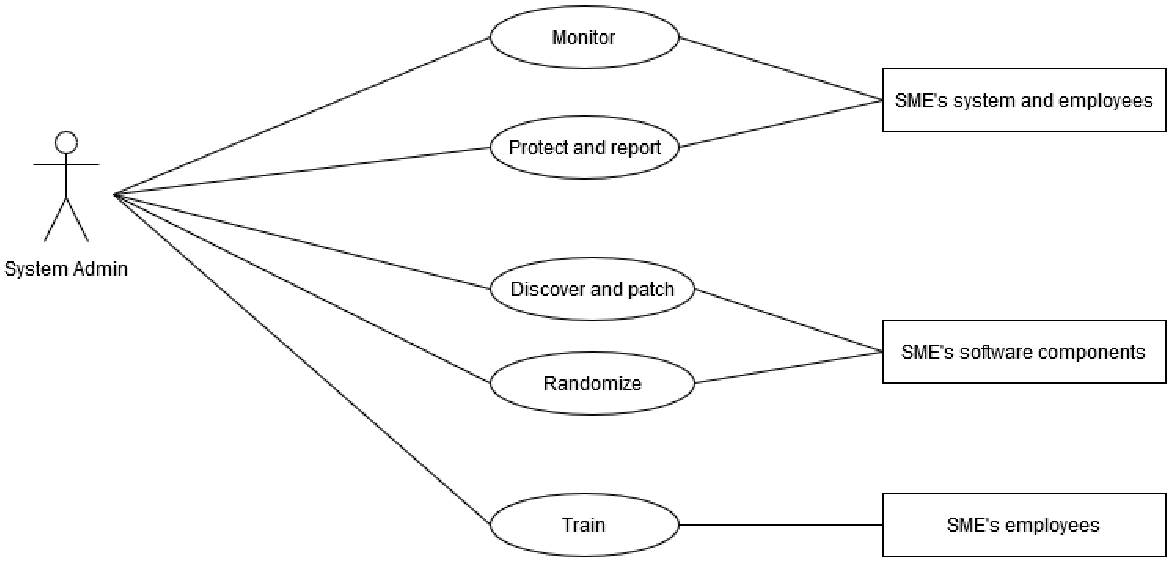

Figure 3: Threat defence use cases

The following Figure 4 describes the security management use cases:

- An actor inserts system information and requirements; the SMESEC Framework returns assessment of the current security level and recommendation for actions by the actor.
- An actor requests updates in the SMESEC Framework configuration; the SMESEC Framework updates the configuration and reports back to the actor.

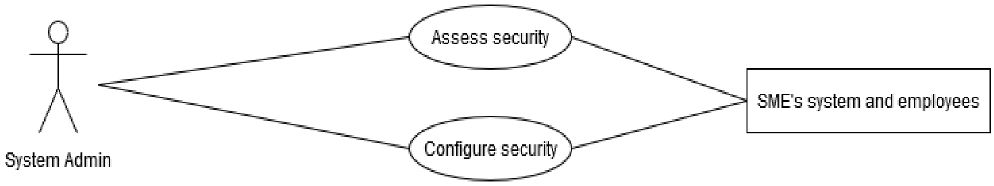

Figure 4: Security management use cases

5.1.2 Concept view

The main design concerns addressed in this concept view are:

- How to design a Framework that orchestrates all SMESEC partner tools
- How to design a Framework that answers the various use case requirements

To answer the above, the SMESEC Framework design extends the standard definition of a security event of adversary attacks detected with the following events: Lack of user training, requirements mismatch, standards non-compliance, user behaviour events, and recommendations not met. This concept of security event allows building a comprehensive end-to-end security solution that solves all SME security concerns in one single framework.

Further, the suggested framework integrates those events into an extended XL-SIEM, creating a security operations centre (SOC) that provides capabilities beyond the classical XL-SIEM. This SOC will serve as a hub for tool interaction and will enable security management as well as threat defence. The SOC will contain information regarding: Requirement managements, Security recommendations, Security intelligence, and Training management. This concept is visually depicted in Figure 5.

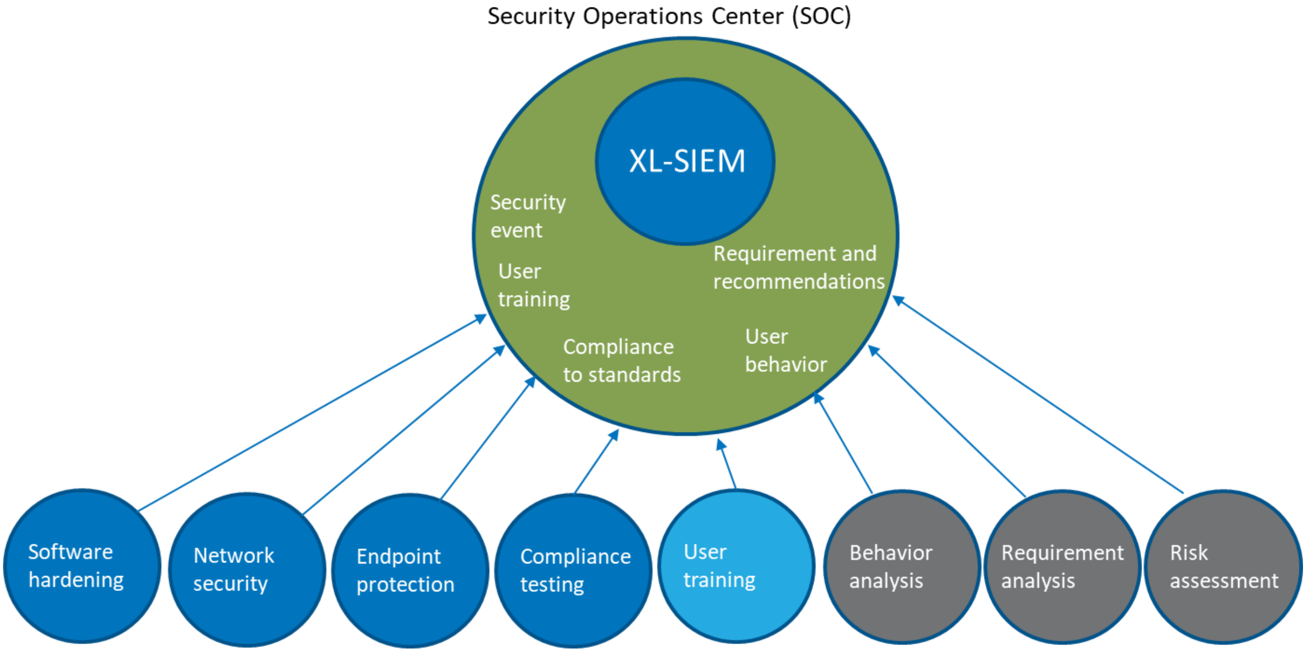

Figure 5: Concept view

5.1.3 Pattern use view

The main design concerns addressed in this view are:

- The SMESEC Framework must allow modular development and deployment of the underlying tools, and the SMESEC Framework. This will allow:
  - Every SMESEC partner tool to exist as a standalone
  - Any suggested integration of partner tools to exist as a standalone
  - Easily adding new tools to the SMESEC Framework
  - Leveraging cloud security capabilities if the SME’s system is deployed on the cloud
- The SMESEC Framework must maintain confidentiality in a hybrid cloud setup
  - Must allow separation of data based on confidentiality
  - Must allow data obfuscation

Figure 6 describes the SMESEC Framework pattern. The pattern is composed of three integration layers and a tools layer. The tools layer includes all SMESEC partner tools except the XL-SIEM that resides in the top layer. New capabilities offered by the orchestration of SMESEC tools will reside in the integration layers. Each one of the integration layers can be composed of several nodes that include presentation, control, and data storage capabilities. The integration layers are:

- Top level layer, composed of one node, responsible for user interface and orchestration of use cases categories.
- Meta-integration layer composed of several nodes and responsible of serving use case categories.
- Tool integration layer composed of several nodes and responsible for integrating tools with similar functionality categories.

Three abstraction terms are used in describing this pattern:

- Presentation (P) is any means of a component to present output to users (e.g., user interface or API)
- Control (C) is any means of controlling a component (e.g., configuration)
- Data (D) is any data stored by the component (raw or obfuscated)

Presentation flows from each tool into a unified SMESEC Framework presentation. Control requests flows from SMESEC Framework actors to every layer of the SMESEC Framework, and from integration layer decisions into lower layers. Data flows across all nodes and is subject to data governance and obfuscation at any level.

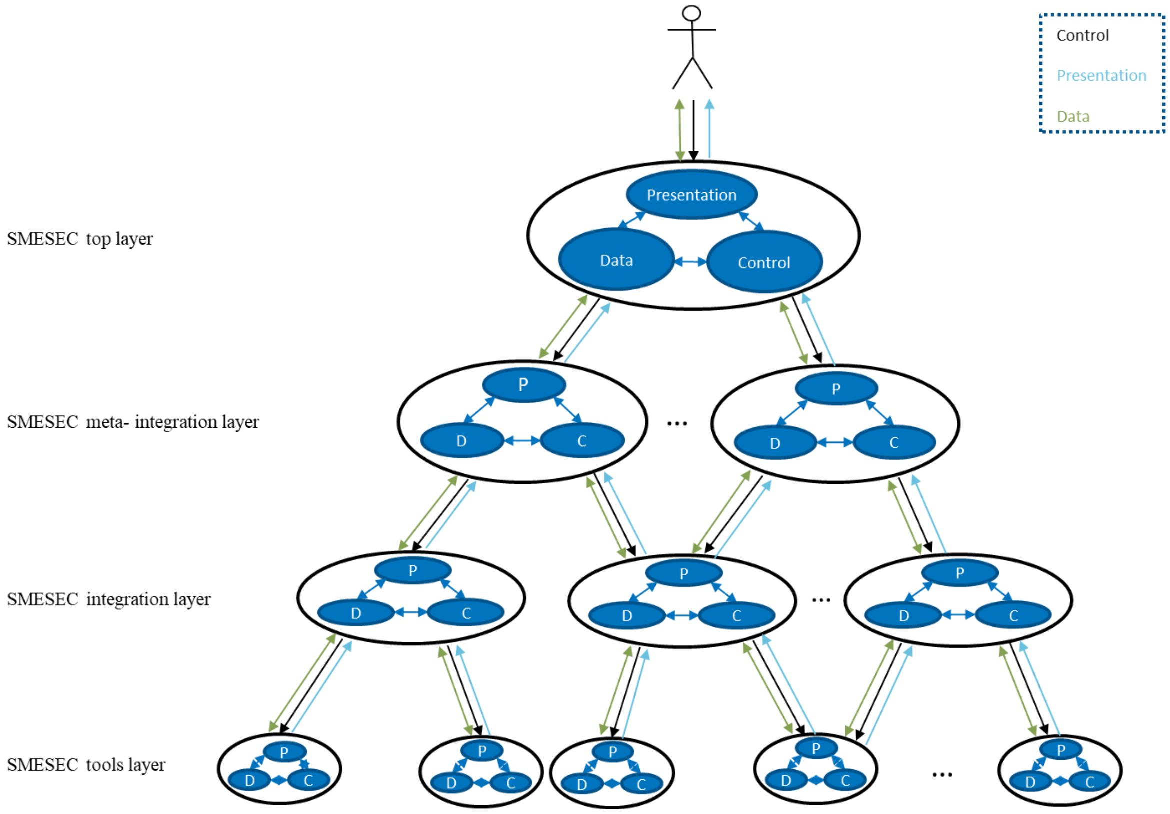

Figure 6: Template view

5.1.4 Composition view

In this view, we address concerns that are related to the composition and modular assembly of the system.

Figure 7 and Table 7 show the component diagram and components description of our system. The diagram shows the nine main components categorized into three integration layers: top layer and interface to users, meta integration layer and tool integration layer. The top layer includes one component: the SOC. The meta integration layer includes two components: Threat Defence meta layer and Security Management meta layer. The tools integration layer includes six components: Monitor and Protect, Vulnerability Discovery, Moving Target, User Training, Security Assessment, and Security Configuration. The system is modular and enables the composition of different implementation of each component. The interface of each component will be described in section 5.1.5 bellow.

The SMESEC Framework is designed around one centralized Security Operations Centre and supports distributed monitoring, protection, vulnerability discovery, user training, security configuration, and security management. It enables adding new capabilities, serving new use case throughout the central SOC, and instantiating multiple instances of integration layer to allow multi-tenancy and load scalability.

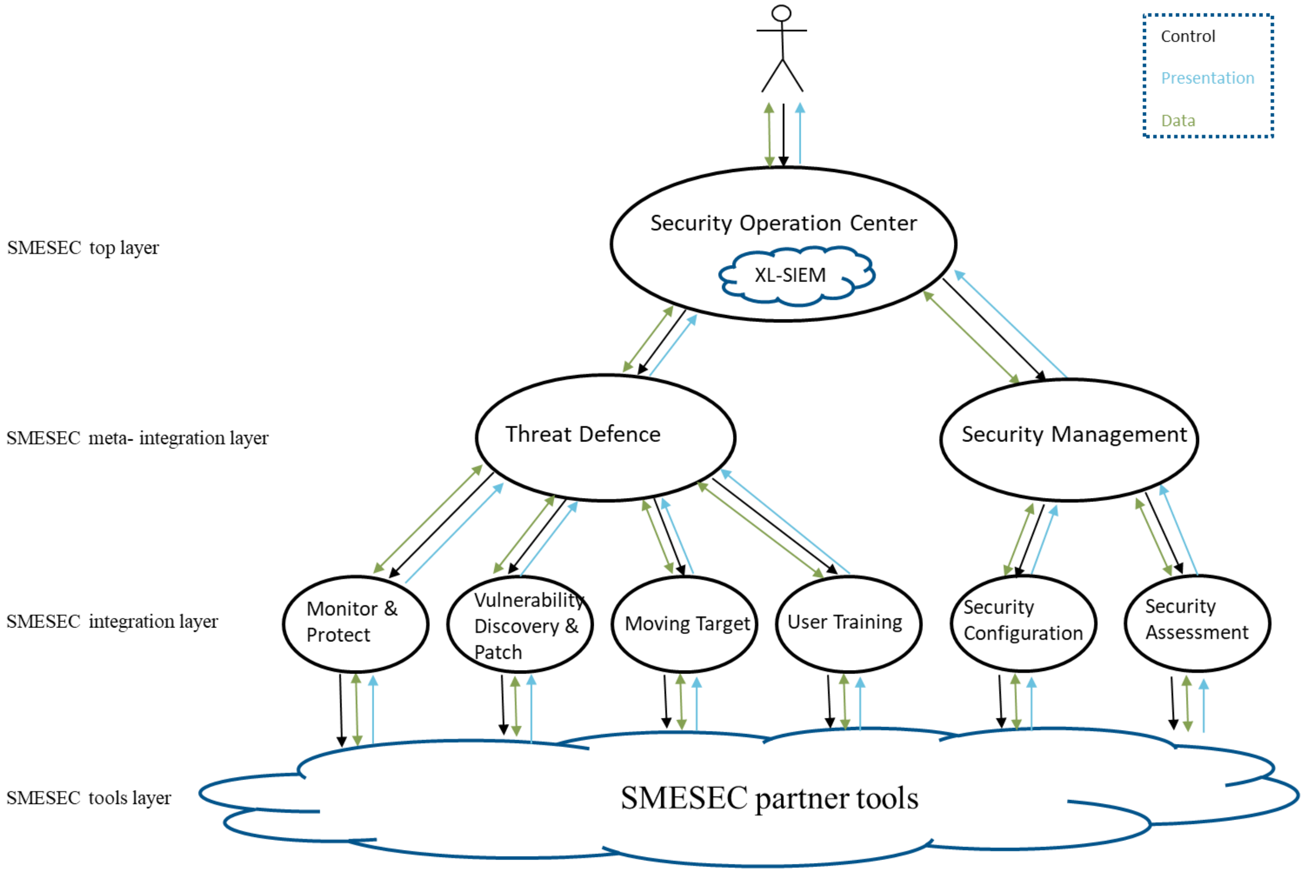

Figure 7: High-level composition view

All packages below use the following template described in Figure 8:

- The Presentation component provides API to the higher-level package to receive requests and data, and to send data and presentation. It forwards requests to the Reasoning component. It receives data and results from the Reasoning component and the History component.
- The Reasoning component provides API to the lower-level package to send requests and data. It receives data from History component and requests from the Presentation components. It sends data to the Presentation component.
- The History component provides API to the lower-level components to receive presentation and data. It sends results data to the Reasoning and Presentation components.

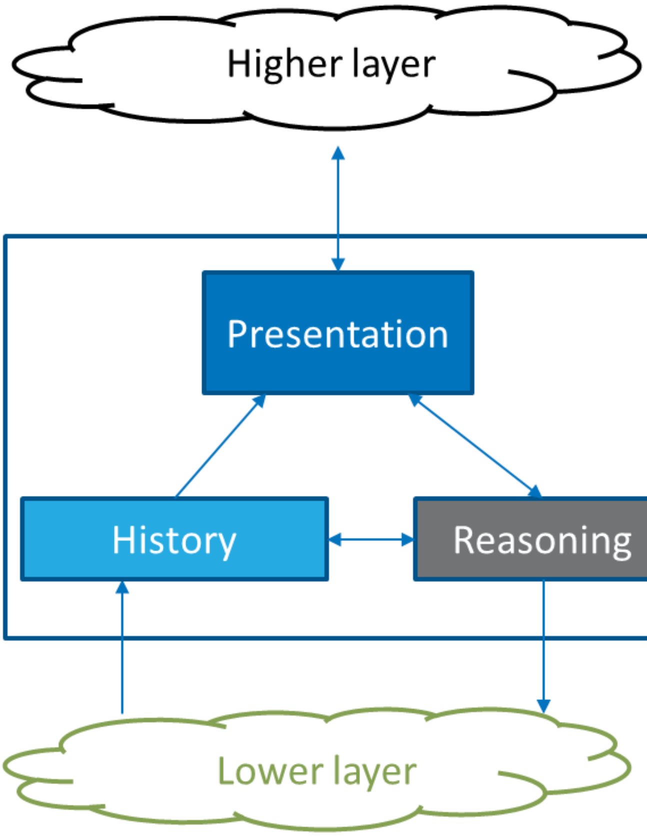

Figure 8: Component template

5.1.4.1 Security Operations Centre - top layer

The Security Operations Centre package described in Figure 9 contains three components: The History component, the Reasoning component, and the Presentation component.

- The Presentation Component provides visual, textual and application program interface to the user.
    - It is responsible on presenting results to the user and receiving user requests.
    - It receives configuration, training, testing, randomization, and assessment requests from the user.

  - It presents real time monitoring results, as well as requests status, insight, recommendation, detect vulnerabilities, and training results. It is responsible for user interaction and interface.
  - It forwards requests to the Reasoning component. It receives data and results from the Reasoning component and the History component.
- The Reasoning component provides API to the meta integration layers to send configuration, training, testing, randomization, and assessment requests.
  - It is responsible for orchestration of meta layers results.
  - It receives data from History component and requests from the Presentation components.
  - It sends data to the Presentation component.
- The History and Status component API to receive presentation, request status and results from the meta layers.
  - It is responsible for gathering request history form Presentation component, status of requests from meta layers, and caching the results history.
  - It reports status, history and results to the Reasoning component and the Presentation component.

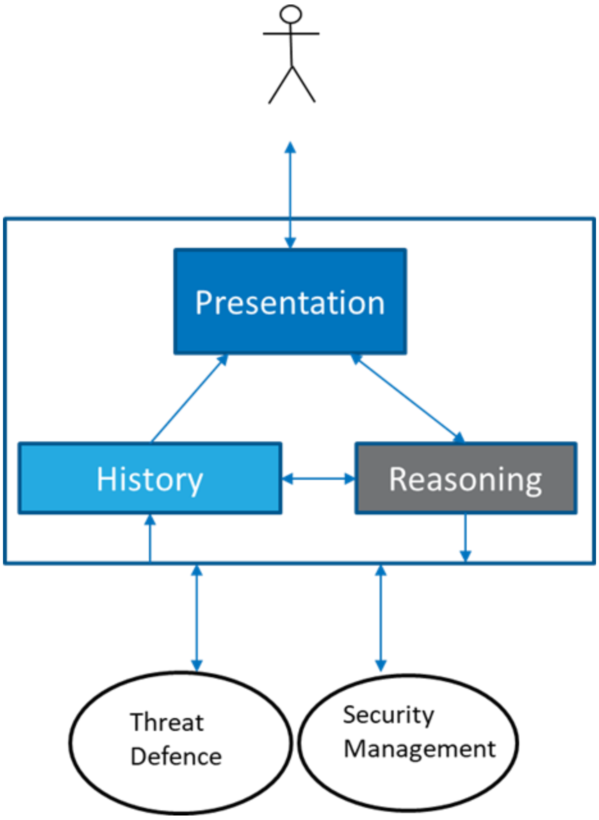

Figure 9: SOC package

5.1.4.2 Threat Defence – meta integration layer

The Threat Defence package described in Figure 10 includes three components: Presentation component, Reasoning component, and History component.

- The Presentation component provides API to the Security Operations Centre package to receive training, testing, randomization requests and report real time monitoring results, as well as requests status, insights, detected vulnerabilities, and training results.
    - It is responsible for interaction with the Security Operations Centre package.
    - It forwards requests to the Reasoning component.
    - It receives data and results from the Reasoning component and History component.
- The Reasoning component provides API to the tool integration layers to send test, train, and randomize requests.
    - It is responsible for orchestration of tool integration layers results.
    - It receives data from the History component and requests from the Presentation components.
    - It sends data to the Presentation component.
- The History component provides API to the tool integration layers to receive monitoring data, test results, train results, and randomization results.
    - It is responsible for caching the results history.
    - It sends results data to the Reasoning and Presentation components.

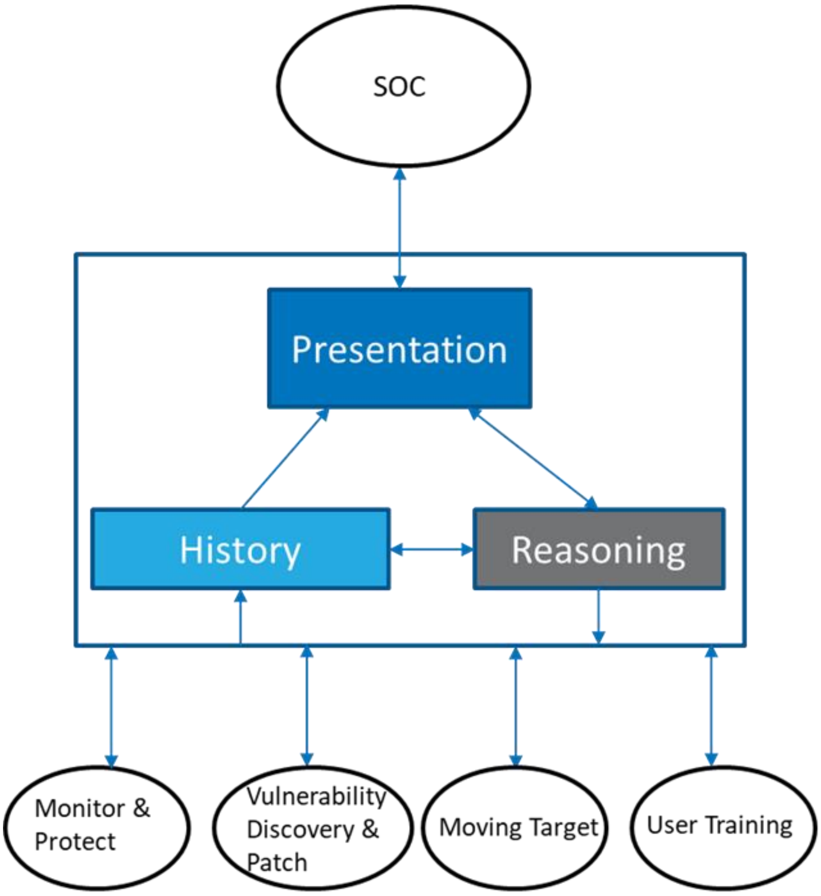

Figure 10: Threat Defence

5.1.4.3 Security Management – meta integration layer

The Security Management package described in Figure 11 includes three components: Presentation component, Reasoning component, and History component.

- The Presentation component provides API to the Security Operations Centre package to receive assessment and configuration requests and report configuration status and security recommendation results.
    - It is responsible for interaction with the Security Operations Centre package.
    - It forwards requests to the Reasoning component.
    - It receives data and results from the Reasoning component and History component.
- The Reasoning component provides API to the tool integration layers to send configuration and assessment requests.
    - It is responsible for orchestration of tool integration layers results.
    - It receives data from History component and requests from the Presentation components.
    - It sends data to the Presentation component.
- The History component provides API to the tool integration layers to receive assessment results and configuration status.
    - It is responsible for caching the results history.
    - It sends results data to the Reasoning and Presentation components.

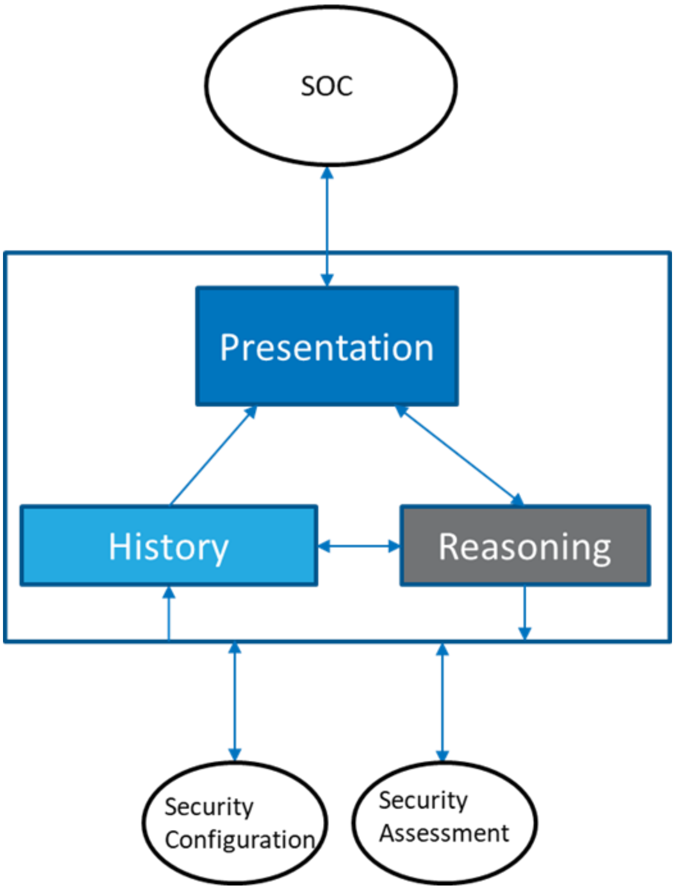

Figure 11: Security Management

5.1.4.4 Monitor and Protect – tool integration layer

The Monitor and Protect package described in Figure 12 includes three components: Presentation component, Reasoning component, and History component.

- The Presentation component provides API to the Threat Defence package to report real time monitoring results.
  - It is responsible for interaction with the Threat Defence package.
  - It forwards requests to the Reasoning component.
  - It receives data and results from the Reasoning component and History component.
- The Reasoning component provides API to the tools' layer to send further protection measures requests.
  - It is responsible for orchestration monitoring tools results.
  - It receives data from History component and requests from the Presentation components.
  - It sends data to the Presentation component.
- The History component provides API to the tools' layer to monitoring and protection results.
  - It is responsible for caching the results history.
  - It sends results data to the Reasoning and Presentation components.

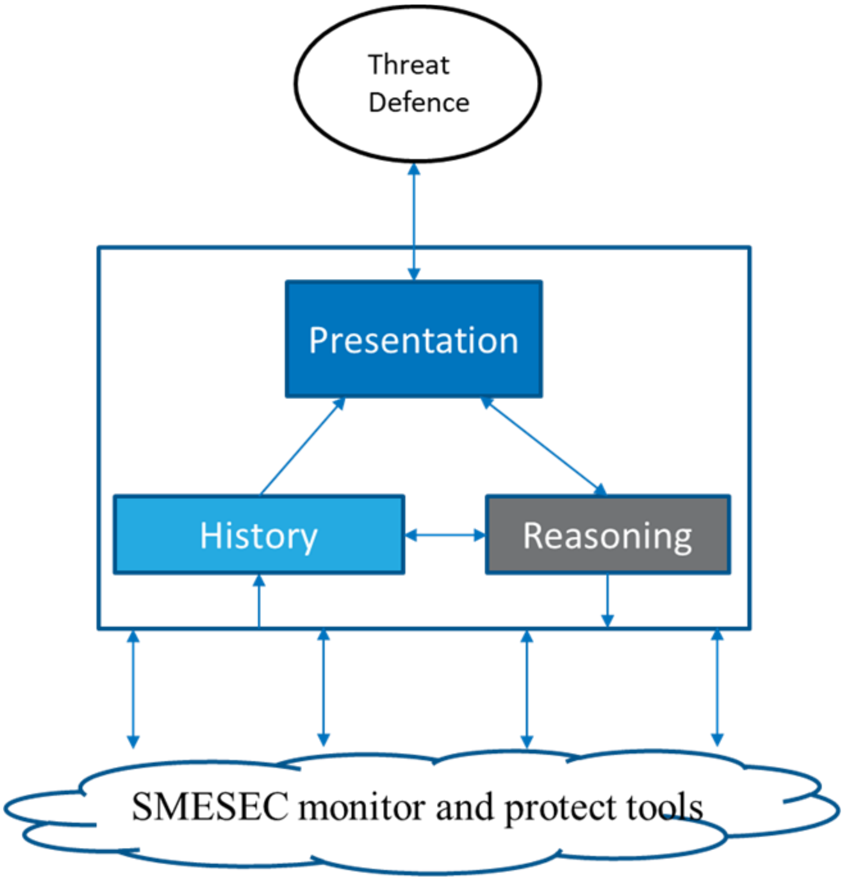

Figure 12: Monitor and Protect

5.1.4.5 Vulnerability Discovery and Patch – tool integration layer

The Vulnerability Discovery and Patch package described in Figure 13 includes three components: Presentation component, Reasoning component, and History component.

- The Presentation component provides API to the Threat Defence package to receive testing and patching requests.
  - It is responsible for interaction with the Threat Defence package.
  - It forwards requests to the Reasoning component.
  - It receives data and results from the Reasoning component and History component.
- The Reasoning component provides API to the tools' layer to send testing and patching requests.
  - It is responsible for orchestration of the tools results.
  - It receives data from History component and requests from the Presentation components.
  - It sends data to the Presentation component.
- The History component provides API to the tools' layer to receive testing results, testing insights, and patching results.
  - It is responsible for caching the results history.
  - It sends results data to the Reasoning and Presentation components.

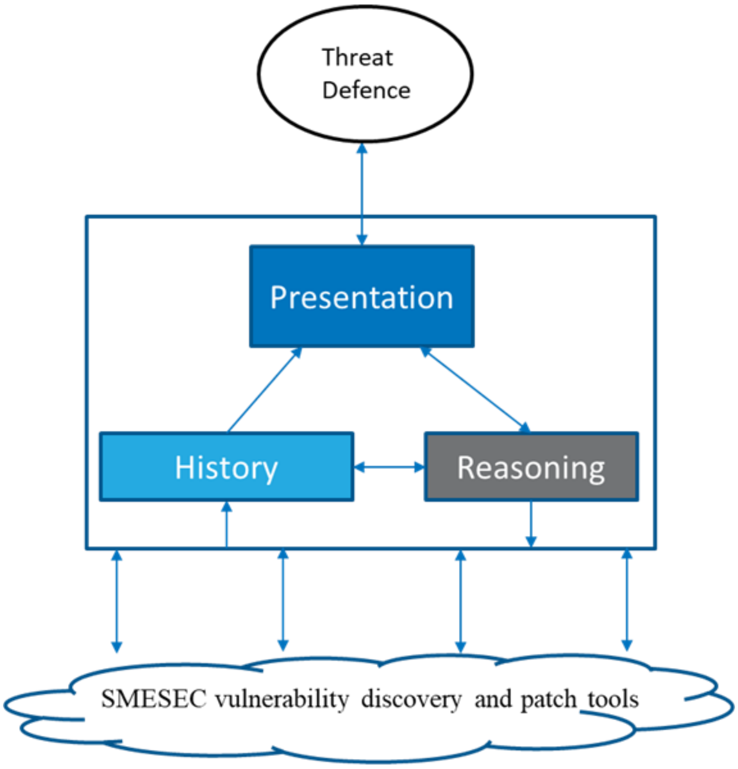

Figure 13: Vulnerability discovery and patch

5.1.4.6 Moving Target – tool integration layer

The Moving Target package described in Figure 14 includes three components: Presentation component, Reasoning component, and History component.

- The Presentation component provides API to the Threat Defence package to receive randomization requests and report results.
    - It is responsible for interaction with the Threat Defence package.
    - It forwards requests to the Reasoning component.
    - It receives data and results from the Reasoning component and History component.
- The Reasoning component provides API to the tools' layer to send randomization requests.
    - It is responsible for orchestration of randomization tools.
    - It receives data from History component and requests from the Presentation components.
    - It sends data to the Presentation component.
- The History component provides API to the tools' layer to receive randomization results.
    - It is responsible for caching the results history.
    - It sends results data to the Reasoning and Presentation components.

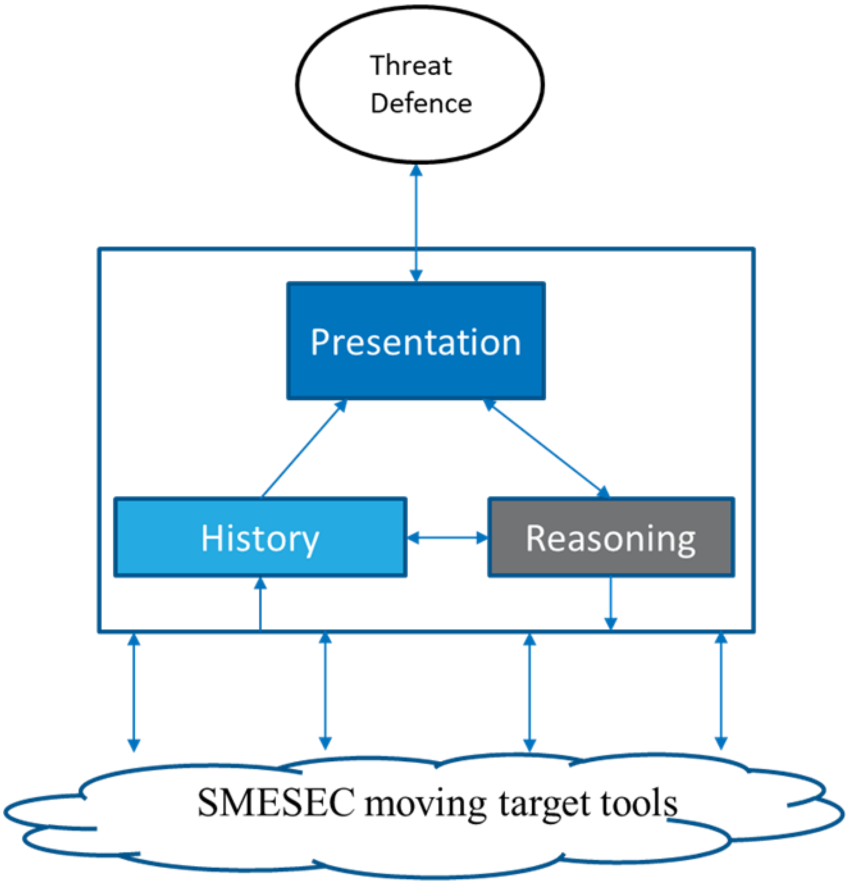

Figure 14: Moving Target

5.1.4.7 User Training – tool integration layer

The User Training package described in Figure 15 includes three components: Presentation component, Reasoning component, and History component.

- The Presentation component provides API to the Threat Defence package to receive training requests and report results.
    - It is responsible for interaction with the Threat Defence package.
    - It forwards requests to the Reasoning component.
    - It receives data and results from the Reasoning component and History component.
- The Reasoning component provides API to the tools' layer to send training requests.
    - It is responsible for orchestration of training tools.
    - It receives data from History component and requests from the Presentation components.
    - It sends data to the Presentation component.
- The History component provides API to the tools' layer to receive training results.
    - It is responsible for caching the results history.
    - It sends results data to the Reasoning and Presentation components.

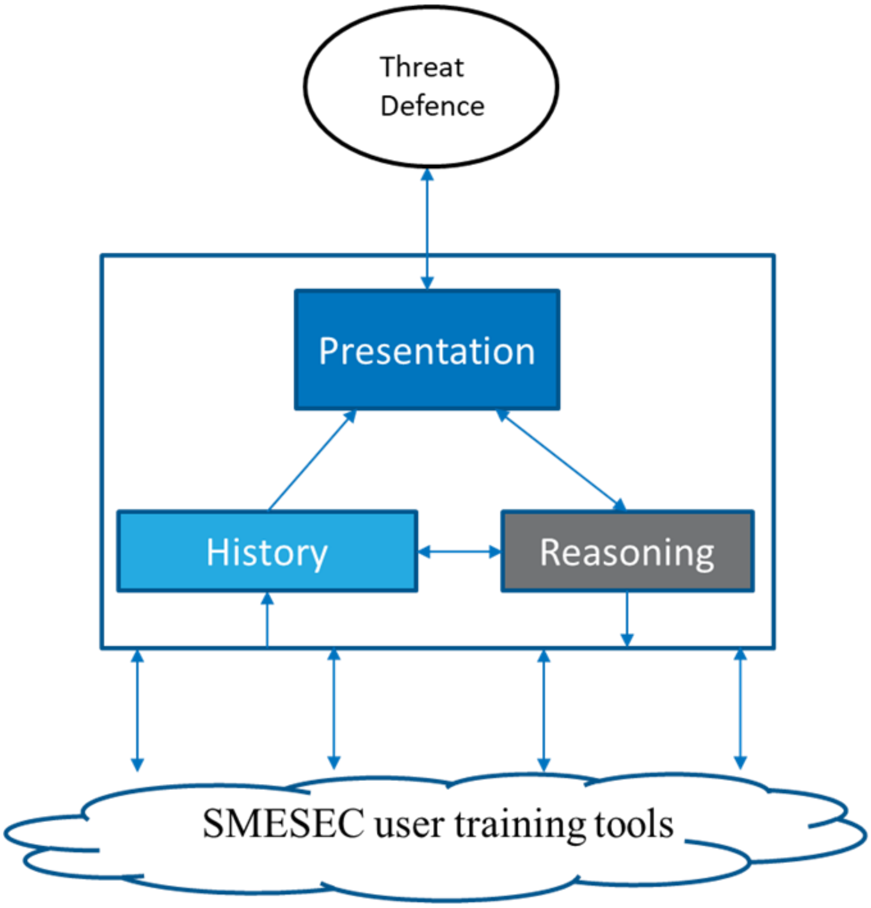

Figure 15: User Training

5.1.4.8 Security Assessment – tool integration layer

The Security Assessment package described in Figure 16 includes three components: Presentation component, Reasoning component, and History component.

- The Presentation component provides API to the Security Management package to receive assessment requests and report results and recommendations.
  - It is responsible for interaction with the Security Management package.
  - It forwards requests to the Reasoning component.
  - It receives data and results from the Reasoning component and History component.
- The Reasoning component provides API to the tools' layer to send assessment requests.
  - It is responsible for orchestration of assessment results.
  - It receives data from History component and requests from the Presentation components.
  - It sends data to the Presentation component.
- The History component provides API to the tools' layer to receive assessment results.
  - It is responsible for caching the results history.
  - It sends results data to the Reasoning and Presentation components.

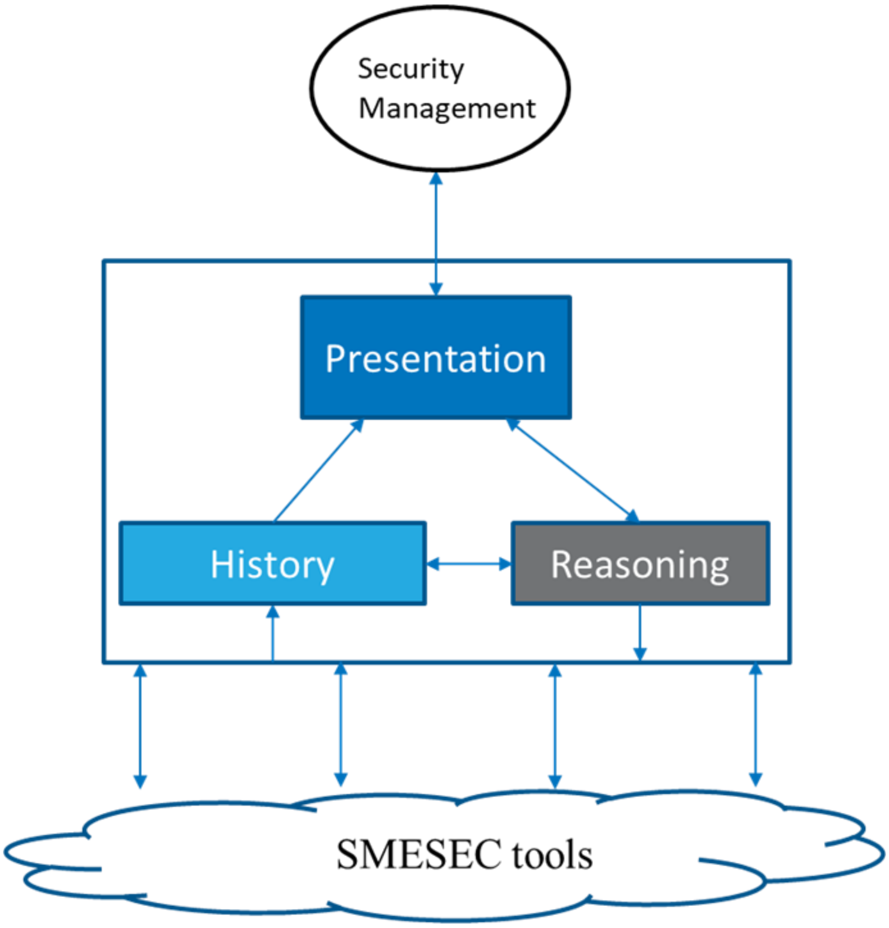

Figure 16: Security Assessment

5.1.4.9 Security Configuration – tool integration layer

The Security Configuration package described in Figure 17 includes three components: Presentation component, Reasoning component, and History component.

- The Presentation component provides API to the Security Management package to receive configuration requests and report status.
  - It is responsible for interaction with the Security Management package.
  - It forwards requests to the Reasoning component.
  - It receives data and results from the Reasoning component and History component.
- The Reasoning component provides API to the tools' layer to send configuration requests.
  - It is responsible for orchestration of configuration results.
  - It receives data from History component and requests from the Presentation components.
  - It sends data to the Presentation component.
- The History component provides API to the tools' layer to receive configuration results and status.
  - It is responsible for caching the results history.
  - It sends results data to the Reasoning and Presentation components.

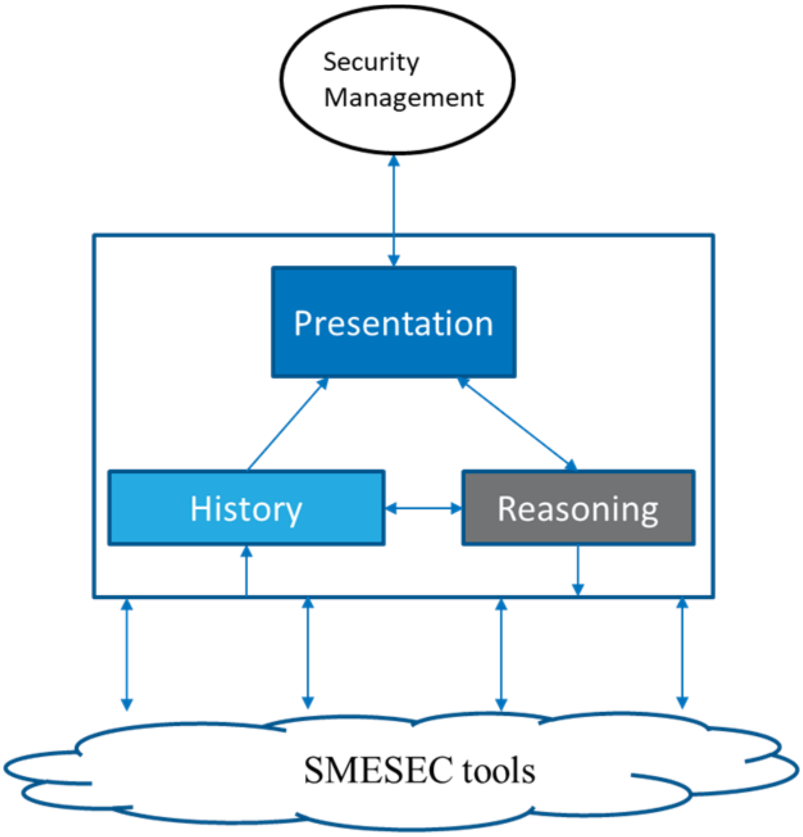

Figure 17: Security Configuration

5.1.4.10 Summary

Table 7 summarizes the packages and components description of our system.

Table 7: Composition view component summary

Package	Component	Description
SOC	Presentation	User interaction and interface
	Reasoning	Orchestration of meta layers results
	Request and History	Gathering request history and status of requests, and caching the results history
Threat Defence	Presentation	Interaction with the Security Operations Centre package
	Reasoning	Orchestration of Monitor and Protect, Vulnerability Discovery and Patch, Vulnerability Discovery and Patch, Moving Target, and User Training integration layers results
	History	Caching the results history
Security Management	Presentation	Interaction with the Security Operations Centre package
	Reasoning	Orchestration of Security Assessment and Security Configuration layers results
	History	Caching the results history
Monitor and Protect	Presentation	Interaction with the Threat Defence package
	Reasoning	Orchestration monitoring tools results
	History	Caching the results history
Vulnerability Discovery and Patch	Presentation	Interaction with the Threat Defence package
	Reasoning	Orchestration testing and patching tools results
	History	Caching the results history
Moving Target	Presentation	Interaction with the Threat Defence package
	Reasoning	Orchestration randomization tools results
	History	Caching the results history
User Training	Presentation	Interaction with the Threat Defence package
	Reasoning	Orchestration training tools results

Package	Component	Description
	History	Caching the results history
Security Assessment	Presentation	Interaction with the Security Management package
	Reasoning	Orchestration of assessment results
	History	Caching the results history
Security Configuration	Presentation	Interaction with the Security Management package
	Reasoning	Orchestration of configuration results
	History	Caching the results history

5.1.5 Interface view

The interface view is used to specify the internal interfaces of the SMESEC Framework. The internal interfaces are divided into two categories: Interfaces between the SMESEC Framework packages, and interfaces between SMESEC Framework packages and the tools provided by SMESEC partners.

Table 8 describes the interfaces between SMESEC Framework packages

Table 8: SMESEC Framework internal package interface

Package 1	Package 2	Send	Receive
SOC	Threat Defence	• Discover vulnerabilities in SW request • Train user request • Randomize SW system request	• Security detected events report • Old software versions report • Discovered vulnerabilities report • User training status report • Randomization status report • Randomized unique software system copies
SOC	Security Management	• Configuration update request • Assessment requests • SME system characteristics and requirement (e.g., the system must comply with an ISO standard)	• Status of tool configuration • Assessment of security level report • Recommendations report

Package 1	Package 2	Send	Receive
Threat Defence	Monitor and Protect	• Discovered vulnerabilities reports • Virtual patch updates	• Security detected events • Old software versions
Threat Defence	Vulnerability Discovery and Patch	• Testing requests • Patching requests	• Vulnerabilities discovered • Testing insights • Virtual patches
Threat Defence	Train Users	• Training requests	• Training status
Threat Defence	Moving Target	• Randomization requests	• Randomization status • Randomized unique SW system copies
Security Management	Security Assessment	• Assessment requests • SME system characteristics and requirement	• Assessment of security level report • Recommendations report
Security Management	Security Configuration	• Configuration update request	• Status of tool configuration

Table 9 describes the interfaces between SMESEC tools to SMESEC Framework packages:

Table 9: SMESEC Framework internal interface to tools

Package	Tool(s)	Send	Receive
Monitor and Protect	Monitoring tools (BD Control Centre, Citrix NetScaler, FORTH EWS)	• Further protection measures request	• Security detected events • Old software versions
Vulnerability Discovery and Patch	Test and Patch tools (EGM tools, IBM AngelEye)	• Test request • Patch request	• Vulnerabilities discovered • Testing insights • Virtual patches
Train Users	Training tools	• Training requests	• Training status
Moving Target	Moving target tools (IBM AntiROP)	• Randomization requests	• Randomization status • Randomized unique software system copies
Security	Security assessment	• Assessment requests	• Assessment of security

Package	Tool(s)	Send	Receive
Assessment	tools (FNHW CYSEC)	• SME system characteristics, security system info, and requirement	level report • Recommendations report
Security Assessment	ALL tools and packages	• Security system info request	• Security system info report
Security Configuration	ALL tools and packages	• Tool configuration request	• Tool configuration report

5.1.6 Discussion about deployment view

The main design concerns in this view are cloud readiness and cloud deployment. Cloud deployment was identified as one of the possible measures to reduce cost of the SMESEC Framework for SME's. A survey of cloud readiness of SMESEC partner tools was conducted in WP2 and reported in D2.1 [3] section 5.5. SMESEC partner tools are not fully ready for cloud deployment. SMESEC Framework will, initially, support on-premise deployment, and gradually shift into public and hybrid cloud deployment.

The design pattern of SMESEC Framework allows modular and gradual deployment, and is trusted to be a solid basis for cloud deployment. This document will not answer the cloud deployment concerns. These concerns will be addressed in future versions of this document.

5.2 User interface and interaction view

The main concerns of this view are usability and extendibility of the SMESEC Framework. The SMESEC Framework design offers a unified interface for all tools included in the SMESEC Framework. Also, the SMESEC Framework interface offers an open integration to external tools.

The user interface is provided using visual, textual, and command line interface. The user interaction is provided using a REST API to the top-level layer. This allows integration of the SMESEC Framework into other tools, and integration of security tools into the Framework.

The user interface and interaction reports:

- Real-time security detected events
- Discovered SW vulnerabilities
- Vulnerable SW versions detected
- User training status
- Randomized unique copies of a SW
- Assessment of security level
- Security recommendations

- Status of tool configuration

The user interface and interaction receives requests to:

- Discover vulnerabilities in SME's SW
- Train SME employees
- Create new unique copies of a SW
- Assess of security level
- Input system information and security requirements
- Update tool configurations

5.3 Design rationale

This section captures the reasoning that led us to this design and the description of the reason of why each element exists.

The SMESEC Framework design as described above aims to allow SME's to deploy a security solution that answers their security concerns as well as their budget limitations. The budget limitation can limit the time and effort they are willing to invest, as well as how much they are willing to pay for purchase of security solutions. The budget limitations as well as the security concerns are often dynamic and change as the SME develops. These concerns led us to the following decisions:

- The design must support security management solutions that: aid the SME in fitting the budget limitations to their security needs and obligations, operate independently of the provided security solutions, and are provided through the same interface as the security solutions.
- The designed system must be highly flexible and allow: customization of security solutions, modular deployment of security solutions, integration to 3rd party solutions, support load scalability, and dynamic on-demand scalability

Those decisions led us to choose a design pattern that is composed of independent components cooperating to perform a unified functionality. This pattern also allows modular deployment, which is an important requirement that stems up from SMESEC tool partners and serves future partners.

In addition, we identified two major use cases for SMESEC Framework: security management and threat defence. This imposed on the design a clear separation of the packages serving those use cases, and a design that supports security management full control of the various threat defence components.

6 Integration of SMESEC tools into the proposed design

6.1 Description of SMESEC tools

The SMESEC contributed products cover a wide range of security market segments, and it is expected that their integration will bring more added value to the products themselves, but also to the unified SMESEC Framework as a whole. The SMESEC product partners have answered in a series of questionnaires in WP2 for the technical capabilities of the products and identified some key interconnection points that can be leveraged for the tool integration. These results are analysed in deliverables D2.1 [2] and D2.2 [3]. A list of all contributed products follows in the next subsections with some key characteristics of each one.

6.1.1 XL-SIEM (ATOS)

ATOS 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. In the core of XL-SIEM, there is a Risk Assessment Engine (RAE) that can run 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.

XL-SIEM uses some open source tools in its core like Apache Storm [6] and AlienVault OSSIM [8] . From an architectural point of view, XL-SIEM can consume information from several agents, consult the OSSIM database, and through a set of Storm-powered processes, triggers the appropriate alarms, actions or monitors events for subsequent use. This is depicted in Figure 18.

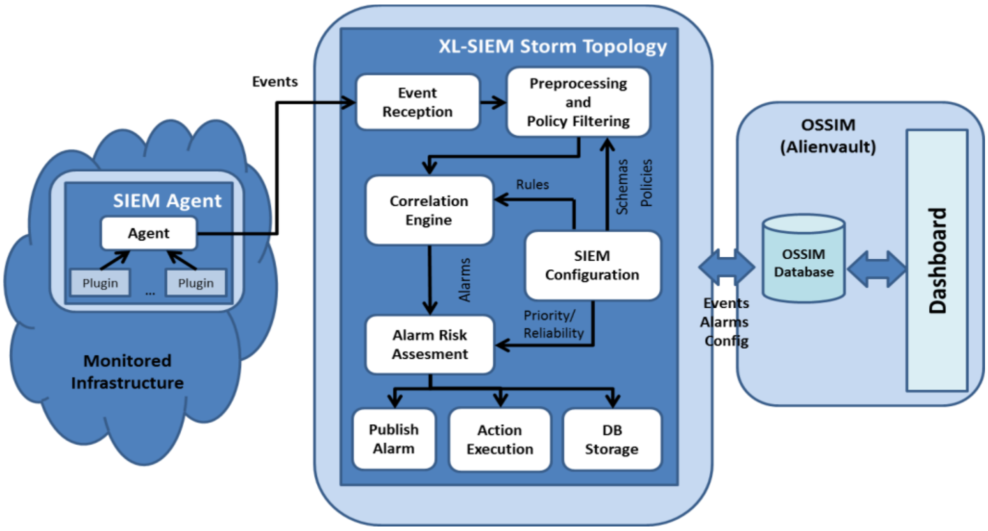

Figure 18: ATOS XL-SIEM architecture

6.1.2 BitDefender GravityZone

BitDefender 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.

GravityZone leverages Bitdefender Advanced Threat Control (ATC) [9] to permanently monitoring running processes for signs of malicious behaviour. With over 500 million machines protected, the Bitdefender Global Protective Network performs 11 Billion queries per day and uses machine learning and event correlation to detect threats without slowing down users.

Figure 19 shows the basic architecture of BitDefender GravityZone and the possible inputs for the GravityZone Control Centre (mobile devices, endpoints, virtualized environments) and the web-based UI where user interacts.

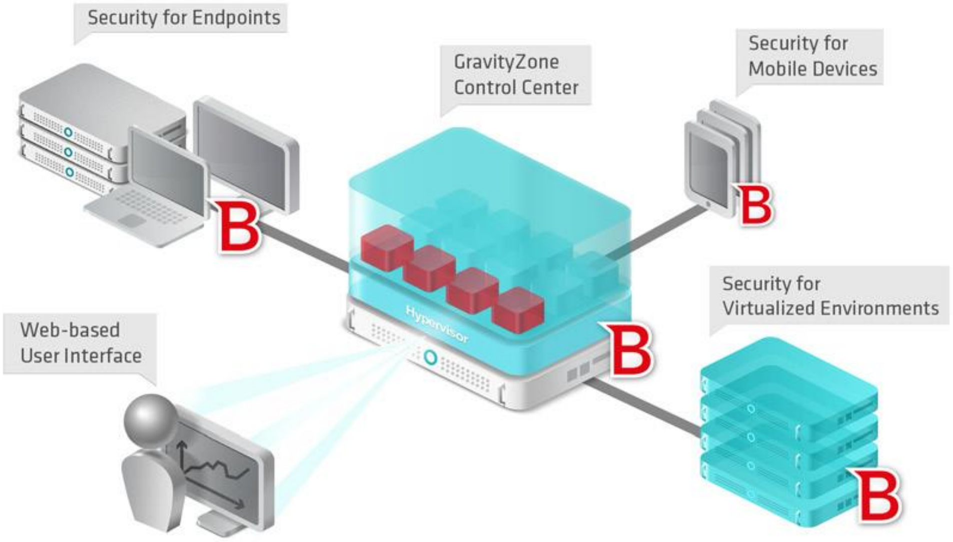
Security for Endpoints
GravityZone Control Center
Security for Mobile Devices
Web-based User Interface
Security for Virtualized Environments
Hypervisor
B
B
B
B

Figure 19: GravityZone solution architecture

6.1.3 Citrix NetScaler AppFirewall

Citrix AppFirewall is a Web Application Firewalls (WAF) [10], that protects web applications and sites from both known and unknown attacks, including all application-layer and zero-day threats.

It provides the ability to perform deep-packet inspection of HTTP, HTTPS and XML as well as protection against OWASP top 10. NetScaler AppFirewall threat protection includes, and is not limited to, SQL injection attacks, cross-site scripting attacks, cookie tampering, form validation and protection, HTTP and XML reply and request format validation, JSON payload inspection, signature and behaviour based protections, data loss prevention (DLP) support including the monitoring of traffic for intended and unintended data exposure, DoS protection, authentication, authorization and auditing support and reporting, and policy tools that provide for easier PCI-DSS compliance verification.

AppFirewall analyzes the traffic to the upper levels of the OSI model. Figure 20 shows the distinction between the network/transport layers and the application layer that AppFirewall operates.

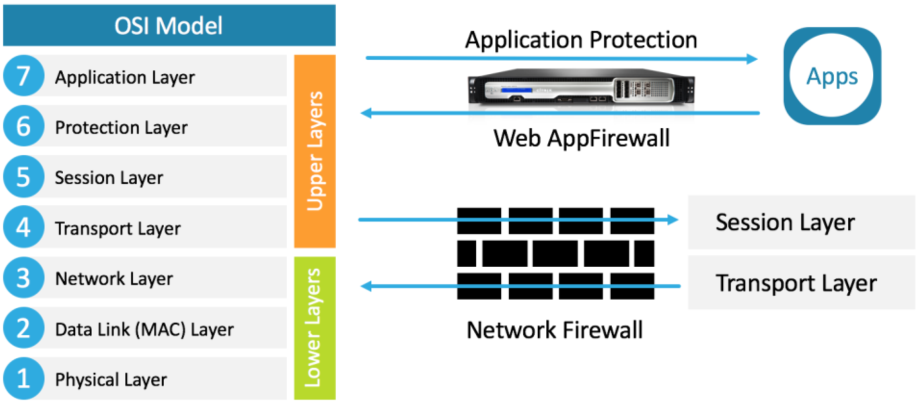

Figure 20: Citrix NetScaler App Firewall

6.1.4 Citrix NetScaler Gateway

NetScaler Unified Gateway [11] consolidates remote access infrastructure to provide single sign-on across all applications whether in a data centre, in a cloud, or delivered as SaaS. It allows people to access any app, from any device, through a single URL.

NetScaler Unified Gateway consolidates multiple remote access solutions, provide Single Sign-On (SSO), multi-factor authentication, end-to-end monitoring across all application traffic and contextual access control across on-premise VDI, web, cloud and SaaS apps. It helps reduce costs, simplify management, and improve the user experience.

Figure 21 demonstrates the variety of device access to a variety of destination applications through Citrix NetScaler Gateway.

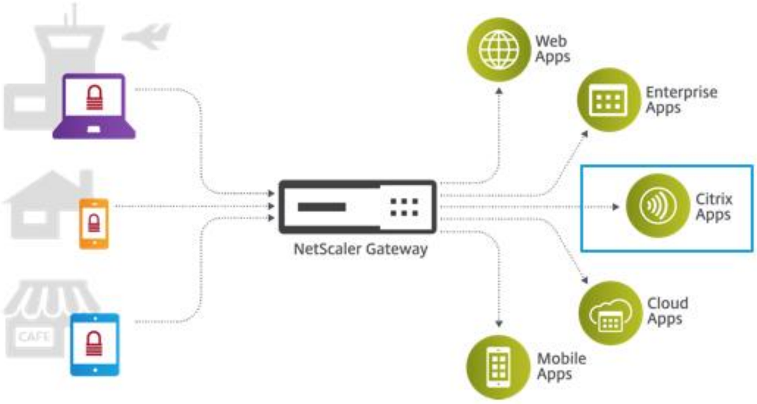

Figure 21: Citrix NetScaler NetScaler Gateway

6.1.5 Citrix NetScaler Secure Web Gateway

Citrix NetScaler Secure Web Gateway (SWG) enforces company security and compliance policies, and gets insights into user behaviour. While encryption protects the privacy and integrity of data, it also creates blind spots that attackers can exploit to evade security controls. Over half of all internet traffic today is encrypted, which creates a rather large gap, exposing a business to increased vulnerability and risk. With SSL decryption, NetScaler Secure Web Gateway helps cost-effectively eliminate blind spots in the business environment and strengthen the security posture.

NetScaler SWG uses a cloud-based service and a local cache to check for URL reputation and category. Address zero-day attacks up to 10 times faster than other forward proxies that have to download a full or partial database. Through this, enforces company security policies on all outgoing web traffic, while blocking access to inappropriate sites on a per user/group basis.

Figure 22 shows where NetScaler stands in the network and the type of services it can offer.

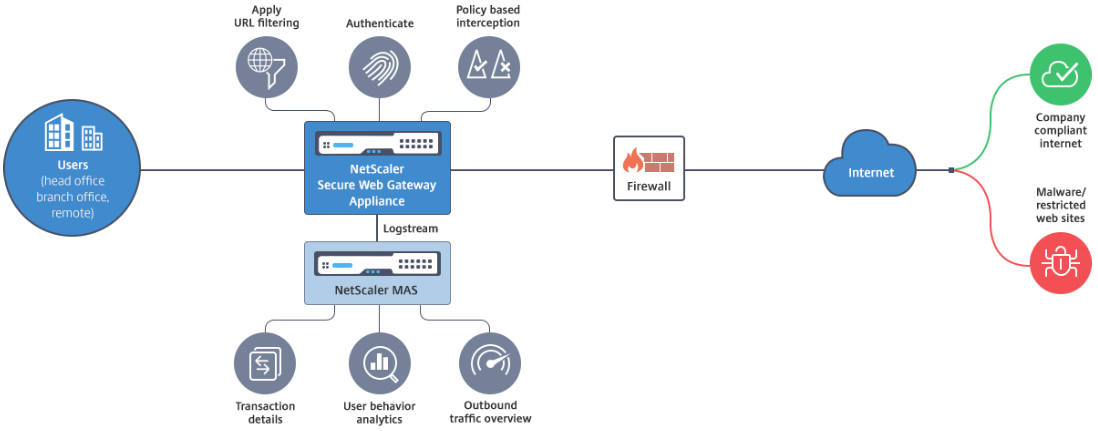
Apply URL filtering
Authenticate
Policy based interception
Users (head office branch office, remote)
NetScaler Secure Web Gateway Appliance
Firewall
Internet
Company compliant internet
Malware/ restricted web sites
Logstream
NetScaler MAS
Transaction details
User behavior analytics
Outbound traffic overview

Figure 22: Citrix NetScaler Secure Web Gateway

Finally, Figure 23 demonstrates a sample topology of all Citrix contributed products (AppFirewall, Gateway, and Secure Web Gateway) as a full network security protection solution.

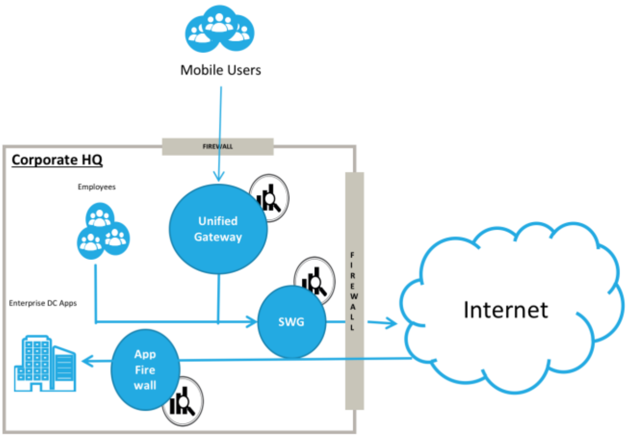

Figure 23: Citrix's overall architecture

6.1.6 EGM 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. End-users can define the configuration 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, etc.

Figure 24 and Figure 25 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.

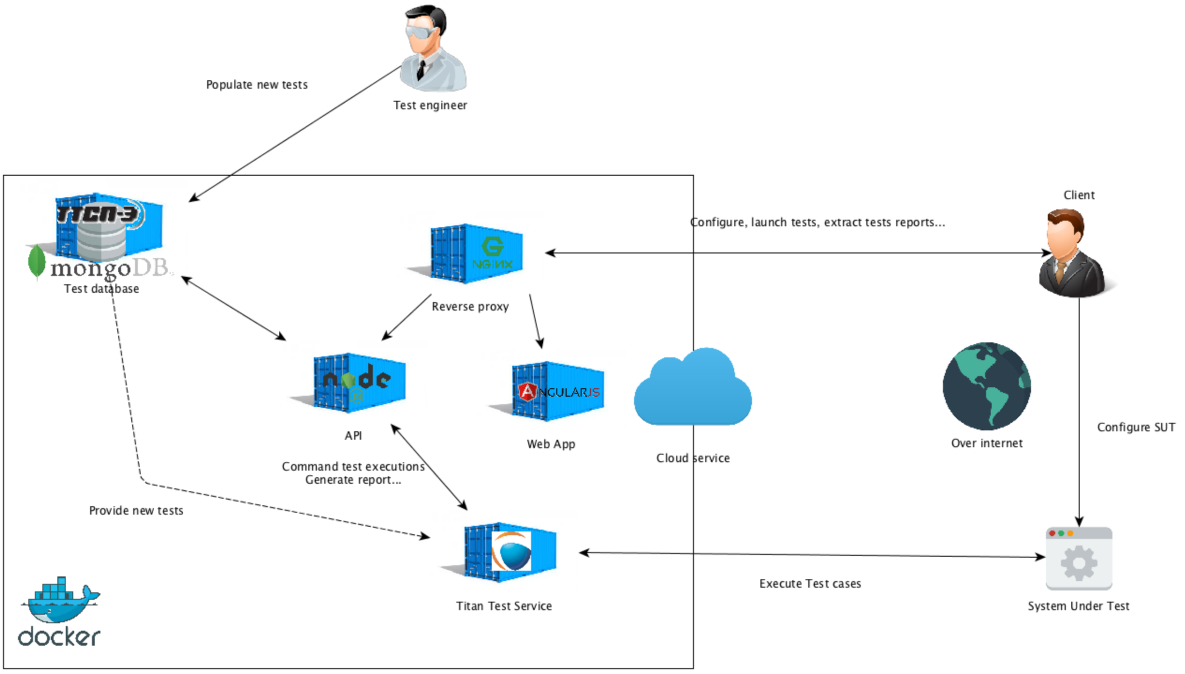

Figure 24: EGM TaaS architecture

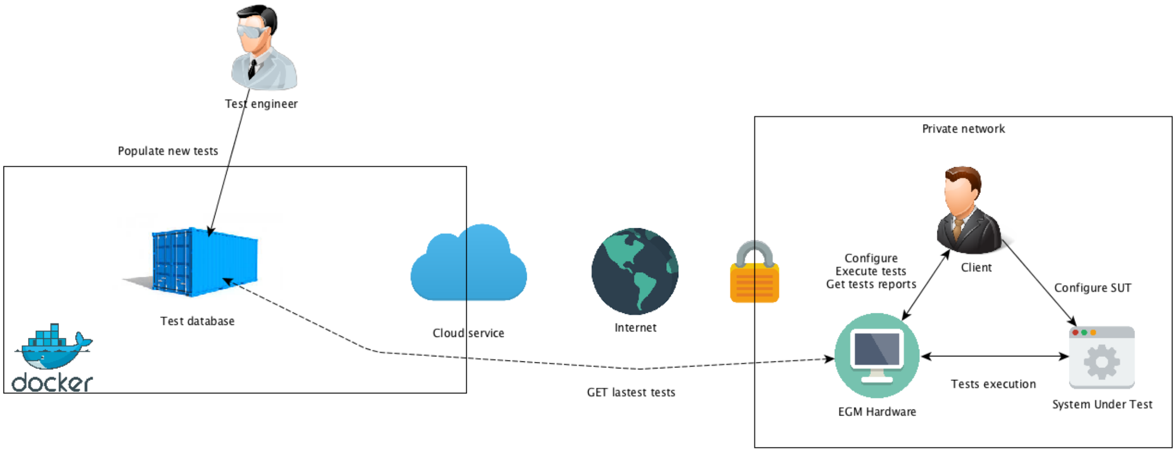

Figure 25: EGM offline testing

6.1.7 FHNW Adherence Monitor and SUPERSEDE Feedback Framework

FHNW solution consists of a web application that can guide the end user (or enterprises) through the process of becoming more secure. The application evaluates answers to questions, and offers pointers to training resources and/or products (including SMESEC ones) for helping to achieve the security goals.

Figure 26 shows the basic architecture of FHNW CYSEC tool. It can run centrally on a server or a VM, but also as a packaged standalone application.

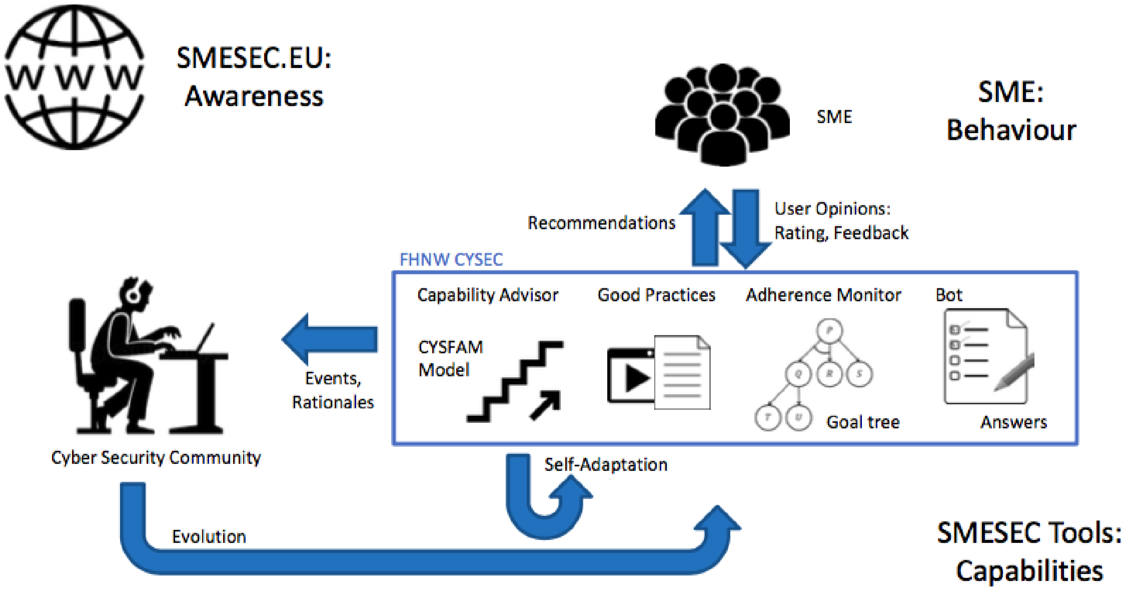

Figure 26: FHNW CYSEC architecture

6.1.8 FORTH EWIS

FORTH 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.

6.1.9 FORTH Cloud-based IDS

A cloud-based IDS tool running on top of Xen hypervisor, which is able to monitor all inter- and intra-hypervisor traffic. Leverages the popular Snort tool as well as other free components to display real-time results via a web interface. Figure 27 shows the basic architecture of the tool.

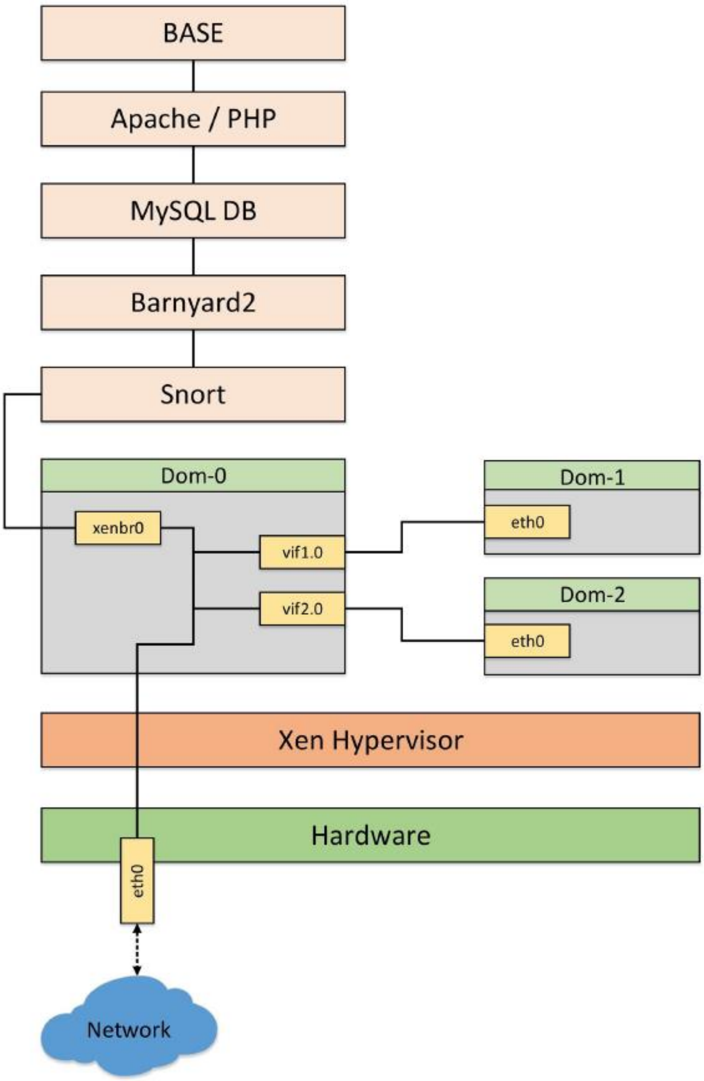

Figure 27: FORTH Cloud-based IDS solution

6.1.10 IBM AngelEye

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 28 shows the overall AngelEye architecture.

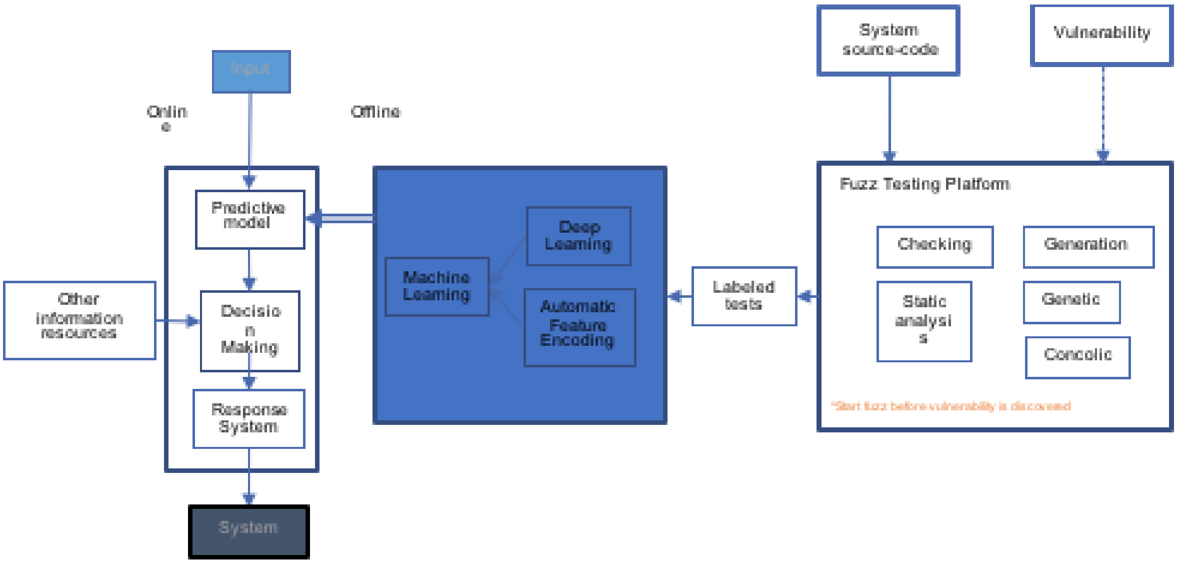

Figure 28: AngelEye solution architecture

6.1.10.1 IBM ExpliSAT

ExpliSAT is integrated into AngelEye 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 29 shows the architecture of the interaction of ExpliSAT (symbolic interpreter) and genetic fuzz testing.

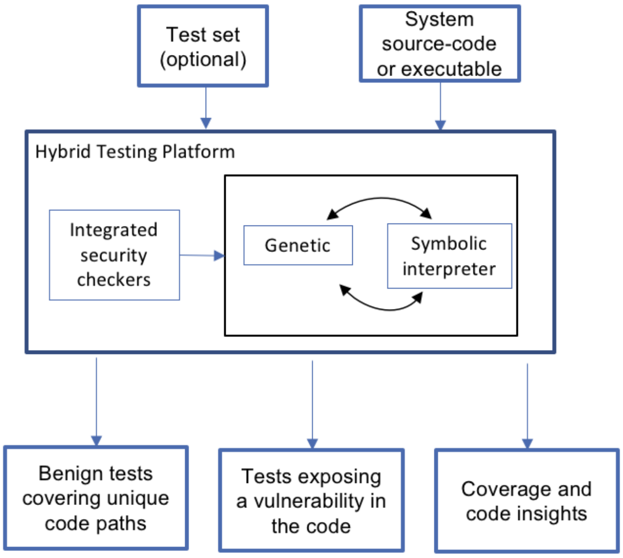
Test set (optional)
System source-code or executable
Hybrid Testing Platform
Integrated security checkers
Genetic
Symbolic interpreter
Benign tests covering unique code paths
Tests exposing a vulnerability in the code
Coverage and code insights

Figure 29: Hybrid testing platform

6.1.11 IBM AntiROP

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 30.

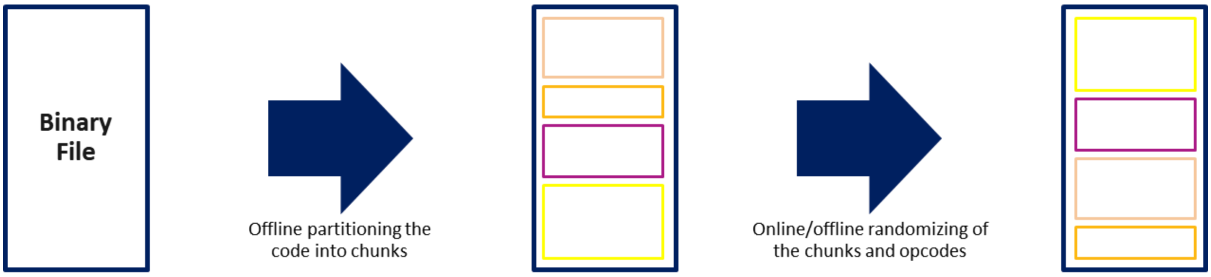
Binary File
Offline partitioning the code into chunks
Online/offline randomizing of the chunks and opcodes

Figure 30: 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 31.

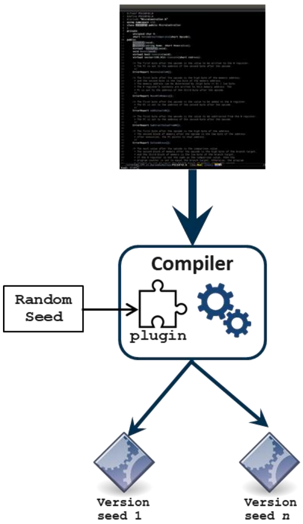

Figure 31: Anti-ROP for source

6.1.12 Discussion about user training tool

In addition to the tools identified above, a set of training tools will be developed and integrated into the SMESEC Framework. More details about these tools will be included in future versions of this document.

6.2 Description of communication of the SMESEC tools

The SMESEC tools form a loosely coupled security framework, where each of them contributes to fulfilling the security requirements of the SMEs. Deliverable D2.2[3] performed a classification of these tools by the communication model into real-time tools and offline tools. The first category are tools that are always on, monitoring the network, the servers, the workstations and any other devices, protecting from incoming attacks in a live manner. The second category of tools usually run on demand, for assessing and improving the security in different areas, from a single file to whole systems.

The identified real-time tools are:

- Bitdefender GravityZone has two components: The EndPoint protection tool, that can be installed on most workstations, regardless of platform or operating system and the Bitdefender Control Centre, that gathers information from the endpoints and is able to control them. The endpoints communicate with the Control Centre using a proprietary protocol, while the Control Centre can output events using the syslog protocol.
- Citrix NetScaler has three components: The app firewall, the SWG and the unified gateway. All three tools can output events. Some events are emitted using the syslog protocol, while other events are outputted using AppFlow and the Nitro API.
- FORTH EWIS detects early intrusions by using honeypots and is also able to output syslog events.

  Atos XL-SIEM is a powerful Security Information and Events Management tool that can receive security events from both the aforementioned tools and from other existing sensors. It supports a wide range of protocols and can trigger various security alerts.

The offline tools asses or improve security and form a heterogeneous framework.

- IBM Anti-ROP receives a PE executable or source code and outputs a shuffled version, resistant to ROP vulnerabilities.
- IBM AngelEye/Explisat receives binary files or source code and outputs tests and virtual patches.
- FHNW CySec tracks information collected over time, observing SMEs involvement into security, by collecting information about the products.
- EGM Test-as-a-Service receives as input the System Under Test and the test suites and outputs the test results. Its architecture makes it easy to orchestrate the usage of the previous tools, providing a unified interface.

6.3 Tool integration into the proposed design

Figure 32 describes the integration plan of the SMESEC tools into the SMESEC Framework design. Each and every one of the tools can have control, presentation, and data API to one of the relevant integration layer. All of the tool have a data API to the Security Assessment component to report the security status of the tool, and all have a control API to the Security Configuration API to allow configuration of the tools. The XL-SIEM tool integration is planed into the top layer.

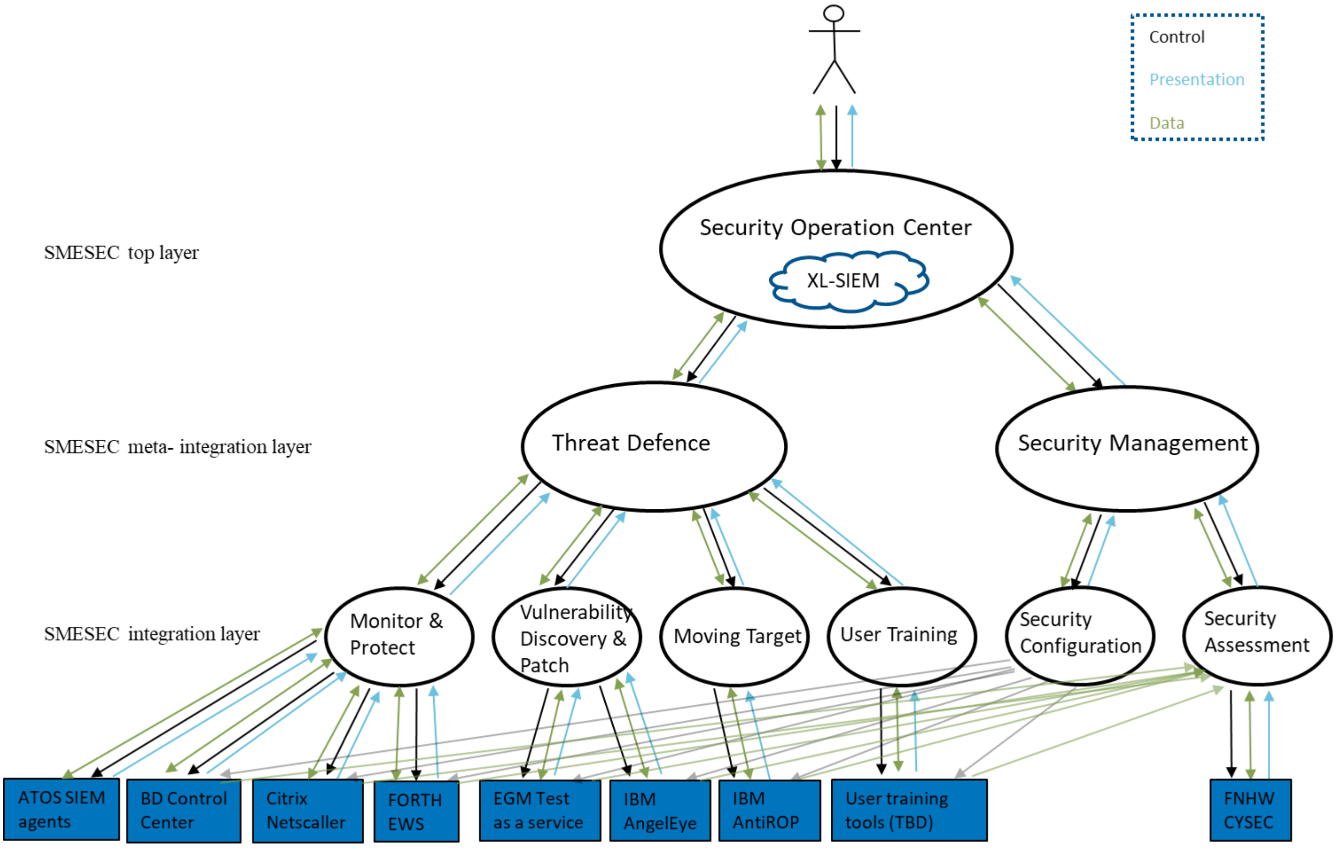

Figure 32: Tool integration

To validate the tool integration into the proposed design we will present several sequence diagrams depicting the most common interactions between the SMESEC components.

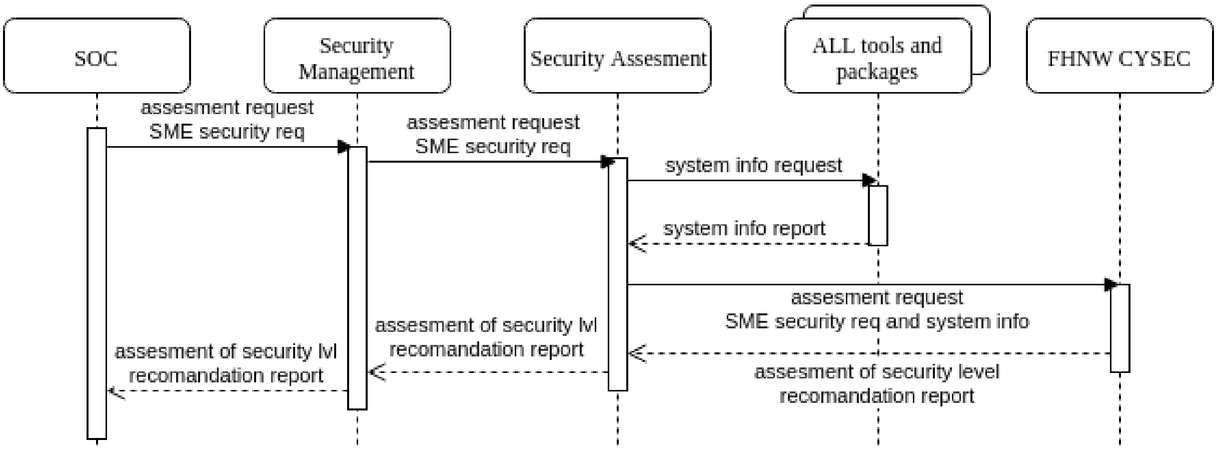
SOC
Security Management
Security Assesment
ALL tools and packages
FHNW CYSEC
assesment request SME security req
assesment request SME security req
system info request
system info report
assesment request SME security req and system info
assesment of security level recomandation report
assesment of security lvl recomandation report
assesment of security lvl recomandation report

Figure 33: Security assessment

Figure 33 presents the scenario where a security assessment is required. The request propagates from the top level (SOC) to the Security Assessment layer that further requests for information from all tools and packages. Using the collected information, along with SME security requirements received from the top level, the FHNW CYSEC tool can produce an assessment of the security level and a recommendations report that is propagated back to the top level.

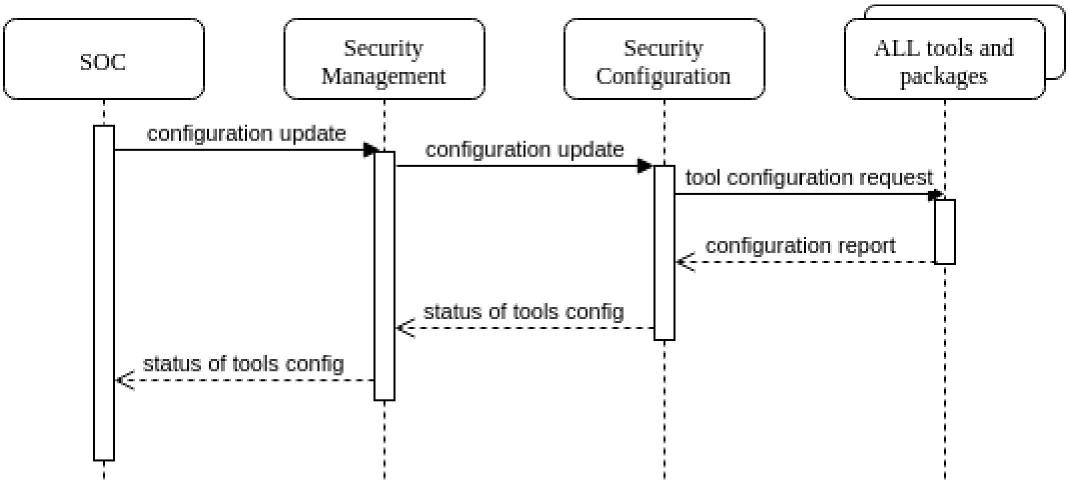

Figure 34: Configuration update

Figure 34 depicts a configuration update scenario where the request is propagated from top layer through Security Management and Security Configuration, reaching all tools and packages that need configuration changes. The configuration report than propagates back to the top layer.

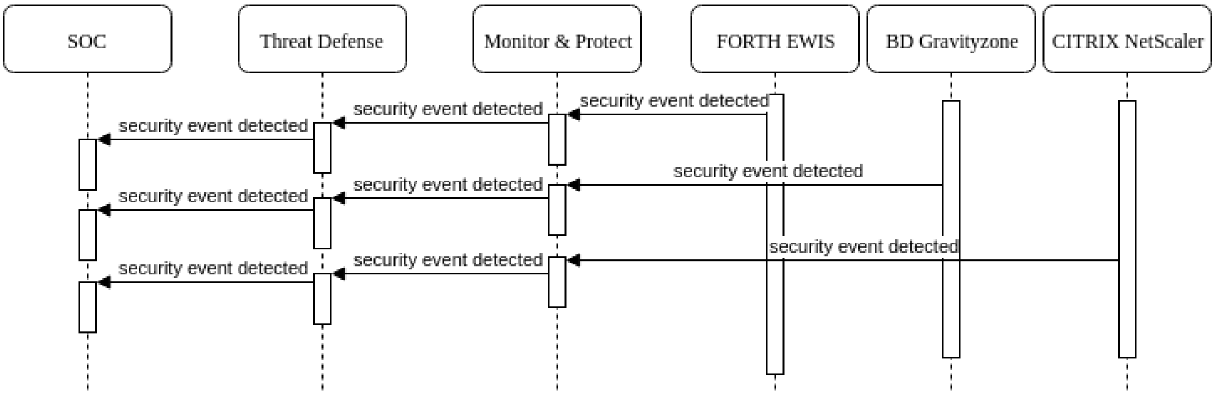

Figure 35: Security event detected

The scenario where a security event is detected (Figure 35) is different from the previous ones. As FORTH EWIS, BD GravityZone and CITRIX NetScaler are real-time tools, they are always active and detection can occur at any moment. When a detection event is triggered, the event is propagated upwards through Monitor & Protect and Threat Defence components to the ATOS XL-SIEM.

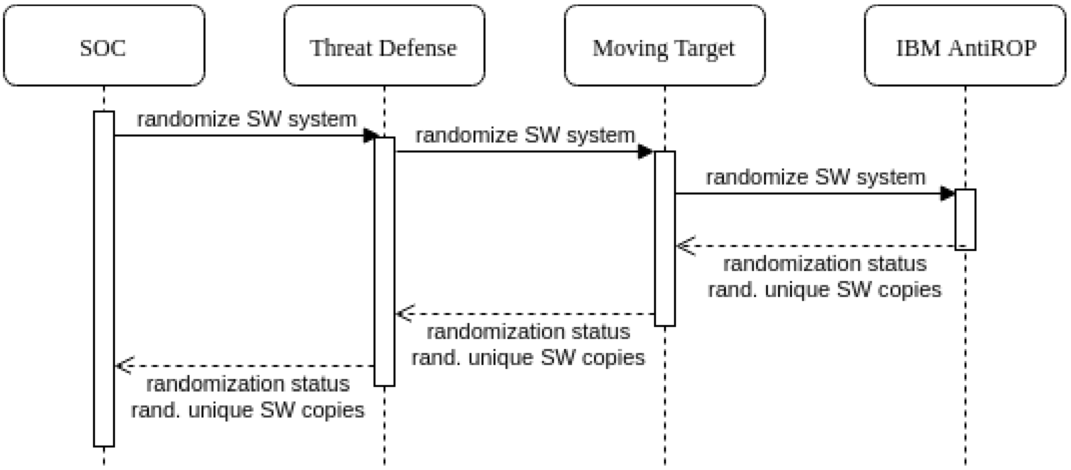

Figure 36: Software system randomization request

In Figure 36 the user performs a software system randomization request through the Security Operations Centre that is propagated through Threat Defence and Moving Target components to IBM AntiROP tool. The tool performs the required randomization, producing randomized copies and outputting a randomization status that propagates back to the SOC.

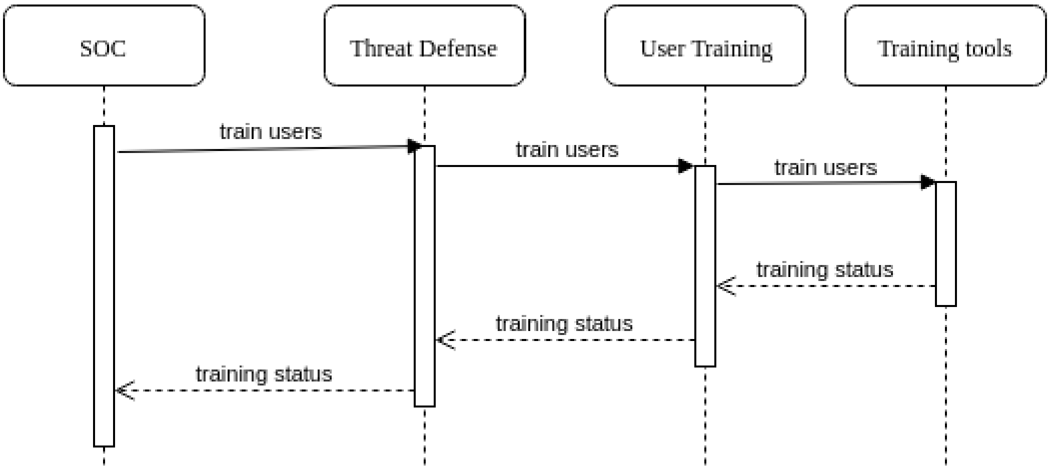

Figure 37: Users training

The user training scenario is depicted in Figure 37. The training request will come from the SOC (top-level) and will propagate through Threat Defence and User Training to the training tools. The training tools, yet to be added to the SMESEC Framework will enable the training modules in the employee's dashboard and notify them that new training modules are available. Since the training will not be done instantly, the training status will be sent back to the SOC asynchronous, after the employees complete their training.

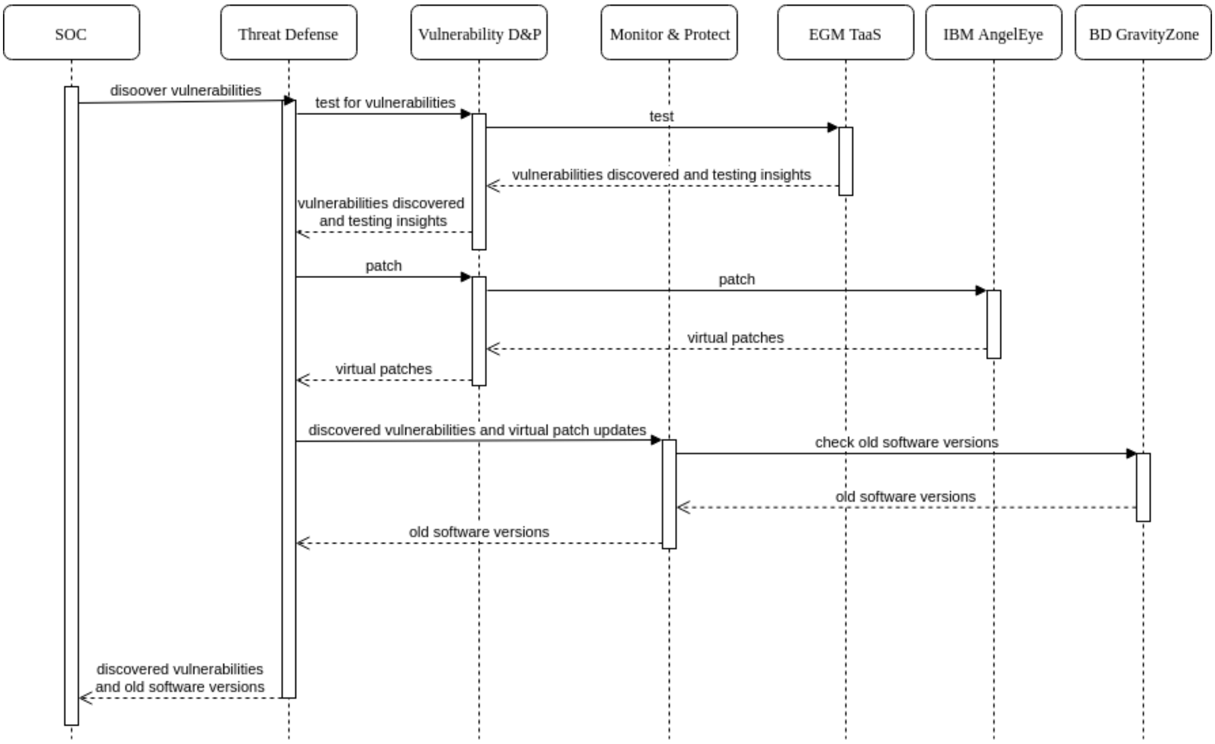
SOC
Threat Defense
Vulnerability D&P
Monitor & Protect
EGM TaaS
IBM AngelEye
BD GravityZone
disoover vulnerabilities
test for vulnerabilities
test
vulnerabilities discovered and testing insights
vulnerabilities discovered and testing insights
patch
patch
virtual patches
virtual patches
discovered vulnerabilities and virtual patch updates
check old software versions
old software versions
old software versions
discovered vulnerabilities and old software versions

Figure 38: Vulnerability assessment and patch

The vulnerability assessment and patch scenario is mostly handled by the Threat Defence component, as depicted in Figure 38. The EGM TaaS tool will receive the test request and will perform the required vulnerability and compliance testing, outputting the discovered vulnerabilities and the testing insights. Based on the test results, the patch request can be issued for the IBM AngelEye tool that is responsible for virtual patches. Finally, Bitdefender GravityZone can identify outdated and vulnerable 3rd party software and even patch in some scenarios. The request for this operation will be propagated through the Monitor and Protect component and the output will be a list of old software found. Finally, the results from all these tools will reach back the XL-SIEM at the SOC.

7 Conclusions

We have identified the stakeholders of the project as the SMESEC use-case partners, SMESEC tool partners, and the EU commission. The main use case partners concerns are security, usability, cost, privacy. The main tool partners' concerns are orchestration between tools, extending current tools, and getting feedback from customer base to drive the development based on customers' needs. The EU commission concerns are providing high degree of usability and automation, adequate degree of cyber situational awareness and control for end-users, incorporating the "human factor" in the design process, and following existing relevant best practices and adoption of standards, tailored to SMEs and individuals.

To respond to the technical and business requirements identified in WP2 and WP6, and to meet the needs of each use case partner an innovation process was established. The main innovation expected from the SMESEC Framework is the integration of different solutions working in an orchestral approach. Future innovation directions of the SMESEC tools were collected and prioritized according to five criteria: Increasing Simplicity of security tools, increase protection level, cost-effectiveness, support training and awareness, and increasing interconnection.

These concerns were translated into functional and non-functional requirements. The functional requirements can be categorised into threat defence and security management requirements. Under threat defence requirements we identified: Protect, detect, monitor, alert, respond, and discover requirements. Under security management we identified: Assess security level, suggest improvements, evaluate risk and consequences, and assess criticality. We have also identified that the incident response and privacy regulation enforcement are out-of-scope for the SMESEC Framework at this stage. The non-functional requirements identified were: modularity of development and deployment, usability, confidentiality, load scalability, multi-tenancy, and expansibility of the framework.

To answer these requirements and concerns we have proposed a design for the SMESEC Framework and developed five design views: Context view, concept view, pattern view, composition view and interface view. The context view describes the various use cases that answer the above requirements. The concept view describes a concept that extends the standard definition of a security event of adversary attacks detected with the following events: lack of user training, requirements mismatch, standards non-compliance, user behaviour events, and recommendations not met. This concept of security event allows building a comprehensive end-to-end security solution that solves all SME security concerns in one single security centre of operation. The pattern view describes a design pattern that allows high modularity and confidentiality by separating the framework into a tree structure of layered nodes, allowing modular development and deployment as well as orchestration innovations and data segregation. The composition view describes each one of these components and its responsibility; these components are all designed using the same template and contain presentation, reasoning, and history components. The interface view describes interfaces between each one of these components. A deployment view is discussed and will be developed only in later stages of the SMESEC project.

A user interface and interaction view is designed to answer usability and extendibility of the SMESEC Framework. The user interface is provided using visual, textual, and command line interface. The user interaction is provided using a REST API to the top-level layer. This allows integration of the SMESEC Framework into other tools, and integration of security tools into the Framework.

The user interface and interaction reports: Real-time security detected events, discovered SW vulnerabilities, vulnerable SW versions detected, user training status, randomized unique copies of a SW, assessment of security level, security recommendations, and status of tool configuration. The user interface and interaction receives requests to: discover vulnerabilities in SME's SW, train SME employees, create new unique copies of a SW, assess of security level, input system information and security requirements, and update tool configurations.

Last, we have examined each SMESEC tool architecture and interface and proposed an integration of the tools into the designed framework. We have successfully validated that the SMESEC tools' integration into this framework is feasible and that future innovations and future tool integration are feasible using sequence view diagrams of the various use-cases.

This document will serve as basis for "SMESEC Unified Architecture" (D3.2), and further development of design views and SMESEC innovation directions will reside in the "SMESEC Unified Architecture" (D3.2).

References

[1] "IEEE Standard for Information Technology-Systems Design- Software Design Descriptions, IEEE STD 1016-2009," 2009.

[2] SMESEC deliverable 2.1 “SMESEC security characteristics description, security and market analysis report”, George Oikonomou, 2017.

[3] SMESEC deliverable 2.2 “SMESEC security products unification report”, Ciprian OPRIȘA, 2017

[4] SMESEC deliverable 2.3 “Security Awareness Plan Report”, Samuel Fricker, 2017

[5] SMESEC deliverable 6.1 “Dissemination plan and market analysis”, Giunta Nicolas, 2017.

[6] H2020 Call Topic DS-02-2016, https://ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/ds-02-2016.html , 2018/02/06.

[7] Apache Storm, https://storm.apache.org , 2018/02/06.

[8] OSSIM: The Open Source SIEM | Alien Volt, https://www.alienvault.com/products/ossim , 2018/02/06.

[9] Bitdefender Advanced Threat Control, http://businessresources.bitdefender.com/hubfs/Bitdefender-Business-2015-SolutionPaper-ATC-93030-en_EN-web.pdf?adobe_mc=MCMID%3D31252819958915218863986580919594145170%7CMCORGID%3D0E920C0F53DA9E9B0A490D45%2540AdobeOrg%7CTS%3D1516630856 , 2018/02/06.

[10] Citrix NetScaler AppFirewall and Web App Service, https://www.citrix.com/content/dam/citrix/en_us/documents/products-solutions/citrix-netscaler-application-firewall-datasheet.pdf , 2018/02/06.

[11] NetScaler Unified Gateway, https://www.citrix.com/products/netscaler-unified-gateway/ , 2018/02/06.

[12] SMESEC Grant Agreement no. 740787 – Annex I Description of the Action (Part B), April 2017.

[13] Spruit, M. CYSFAM, Cyber Security Focus Area Maturity Model, in publication.

OWASP Top Ten Project, https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project, 2018/02/16