Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

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

D2.2 SMESEC security products unification report

Document Identification			
Status	Final	**Due Date**	30/11/2017
Version	2.1	**Submission Date**	19/12/2017

Related WP	WP 2	**Document Reference**	D2.2
Related Deliverable(s)	D2.1, D2.3	**Dissemination Level (*)**	PU
Lead Organization	BD	**Lead Author**	Ciprian OPRIȘA (BD)
Contributors	ATOS, BD, CITRIX, EGM, FHNW, FORTH, IBM	**Reviewers**	Kostas LAMPROPOULOS (UoP)
			Fady COPTY (IBM)

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.

Keywords:
security, market, survey, WP2, requirements, capabilities, use case, protection, endpoint, IoT, sensors, incidents, threats, smart city, smart grid, e-voting, firewall, SIEM, anti-virus, testing, compliance, maturity model, honeypots, patching, anti-ROP

Document Information

List of Contributors	
Name	Partner
Ciprian OPRIȘA	BD
Ovidiu MIHĂILĂ	BD
George OIKONOMOU	CITRIX
Fady COPTY	IBM
Ayman JARROUS	IBM
Dov MURIK	IBM
Jordan MARTIN	EGM
Phillipe COUSIN	EGM
Samuel FRICKER	FHNW
Martin GWERDER	FHNW
Sotiris IOANNIDIS	FORTH
Christos PAPACHRISTOS	FORTH
Susana GONZALEZ ZARZOSA	ATOS
Jose Francisco RUIZ	ATOS

Document History			
Version	Date	Change editors	Changes
0.1	01/11/2017	Ciprian OPRIȘA BD	The 1st draft of the deliverable including all the information collected from partners.
0.2	15/11/2017	Ovidiu MIHĂILĂ BD	2nd draft including the Executive Summary, Introduction, Conclusions within new template.
0.3	27/11/2017	Ovidiu MIHĂILĂ BD	3rd draft considering ATOS new input, UoP feedback as 1st reviewer and newer text revision.
0.4	29/11/2017	Ovidiu MIHĂILĂ BD	4th draft considering IBM's input as 2nd reviewer
1.0	30/11/2017	Ovidiu MIHĂILĂ BD	5th draft with minor revisions
2.0	15/12/2017	Ovidiu MIHĂILĂ BD	FINAL VERSION
2.1	18/12/2017	ATOS	Quality check and submission to EC.

Table of Contents

List of Tables

List of Figures

List of Acronyms

Abbreviation / acronym	Description
AMQP	Advanced Message Quering Protocol
API	Application Process Instruction
DNS	Domain Name System
DDOS	Distributed Denial of Service (network attack; also seen as DDSA)
Dx.y	Deliverable number y belonging to WP x
EC	European Commission
EWIS	Early Warning Instruction detection System
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDS	Instruction Detection System
IPS	Instruction Prevention System
IPFIX	Internet Protocol Flow Information Export
JSON	Javascript Object Notation
LDAP	Lightweight Directory Access Protocol
MBT	Model Based Testing Tool
MSSQL	Microsoft Structured Query Language
My SQL	My Structured Query Language
M2M	Machine-to-Machine (communication, mainly mobile)
NIDS	Network Intrusion Detection Systems
OSSIM	Open Source Security Information Management
REST	Representational state transfer
SDK	Software Development Kit
SMB	Server Message Block (protocol)
SME	Small Medium Enterprise
SMTP	Simple Mail Transfer Protocol (internet email)
SSL	Security Socket Layer
STIX	Structured Threat Information eXpression

Abbreviation / acronym	Description
SUT	System Under Tool
SWG	Secure Web Gateway
TTCN-3	Test Control Notation version 3
TFTP	Trivial File Transfer Protocol
WP	Work Package

Executive Summary

This document presents the results of the technical analysis of each and every cybersecurity solution which will be part of the SMESEC unified framework whose role is to protect SMEs against complex and various threats. Furthermore, it contains a high-level snapshot of the common integration principles together with a perspective on innovation enhancements.

This document is the 2nd deliverable of WP2 and represents a key-element of this WP, interconnected with D2.1 – "*SMESEC security characteristics description, security and market analysis report*" and D2.3 – "*Security Awareness Plan Report*". Also, D2.2 is an important input for WP3 as it provides key-data for designing the SMESEC framework architecture and also for WP5 as it illustrates the innovation approach.

The solutions of SMESEC partners have been analysed on the technical level aiming to obtain a well-documented illustration of the characteristics and features of each security solution which will be integrated within the SMESEC security framework, starting with: (1) solutions' overview to understand their role within the SME protection, (2) description of technical characteristics to understand their complementarity, (3) detailed description of each solution's architecture to learn about integration options and (4) overview of data flow in a form of input / output.

These solutions cover a wide range of the security areas and some key-interconnection points have been identified and described. The goal was to understand how these solutions could be effectively and commonly integrated in order to match the SME pilots' requirements.

Finally, an approach on innovation has been drafted in order to develop key-differentiators of SMESEC unified framework which will bring competitive advantage within the market of cybersecurity solutions for SMEs.

The key-output of this deliverable is that the solutions providers are key-actors of the market of cybersecurity solutions with proven and successful track-records and they provide key-security solutions covering different requirements which proves the complementarity among solutions and contributes to a consistent unified framework. Also, the report shows that there are similarities among the solutions which facilitate the common integration, thus the resulted unified SMESEC framework will provide added-value to all individual solutions, multiplying the benefits for SMEs.

1 Introduction

1.1 Purpose of the document

The document is the 2nd deliverable of WP2 "*Adaptation of SMESEC security components to SMEs requirements*". This WP's main impact within the SMESEC project is to offer a solid ground in terms of technical and market information which will constitute the foundation for the following WPs: WP3 – definition of SMESEC framework architecture, WP5 – innovation enhancement and WP6 – security market approach.

D2.2 – "*SMESEC security products unification report*" provides a well-documented illustration of the characteristics and features of each security solution which will be integrated within the SMESEC security framework, starting with: (1) solutions' overview to understand their role within the SME protection, (2) description of technical characteristics to understand their complementarity, (3) detailed description of each solution's architecture to learn about integration options and (4) overview of data flow in a form of input / output. On the next level, the document presents basic principles for integration of all security solutions into a unified framework which matches the SMEs requirements.

All in one, the purpose of this deliverable is to collect essential information about the security solutions which will be part of SMESEC and to provide the foundation for the integration approach for a unified framework, considering innovation enhancements.

1.2 Relation to another project work

D2.2 – "*SMESEC security products unification report*" covers a technical understanding of all security solutions. This is essential for the following interdependencies with:

- WP2 / D2.1 – "*SMESEC security characteristics description, security and market analysis report*"; it complements the key-findings about SMEs' needs in terms of cybersecurity by illustrating how the solutions' features cover individual and common specificities;
- WP2 / D2.3 – "*Security Awareness Plan Report*"; complements the initial pilot risk assessment;
- WP3 – provides key-information about security solutions which supports the definition of the unified architecture;
- WP5 – provides information about future innovation.

1.3 Structure of the document

The document is structured in 5 major chapters as following:

- Chapter 1 introduces the reader to the main information about D2.2.

- Chapter 2 describes each security solution (Atos' **XL-SIEM**, Bitdefender's **GravityZone**, Citrix's **NetScaler**, Easy Global Market's **Test-as-a-Service platform**, Fachhochschule Nordwestschweiz' **CySec**, FORTH's **EWIS** and IBM's **AngelEye** and **Anti-ROP**) which will be integrated within the SMESEC security framework, starting with: (1) solutions' overview to understand their role within the SME protection, (2) description of technical characteristics to understand their complementarity, (3) detailed description of each solution's architecture to learn about integration options and (4) overview of data flow in a form of input / output.
- Chapter 3 presents the basic principles for a common integration of all solutions into a unified framework considering: backend mechanism, deployment and update implementations and an integration timeline.
- Chapter 4 provides with information about innovation enhancements.
- Chapter 5 lists the conclusions of this document.

2 SMESEC Solution and Services

2.1 Overview

2.1.1 ATOS XL-SIEM solution overview

Table 1. XL-SIEM solution overview

Partner and solution name	ATOS / XL-SIEM
Actions performed by your solution (Detection, Prevention, Mitigation, Avoidance)	Detection
Impact on SMEs (what benefits your solution brings when deployed on a SME)	The tool will allow SMEs to detect intrusions and malicious activities in their system (e.g. database, communication channels, interfaces, etc.). The tool also provides reports of the status of the system so SMEs can always be up-to-date of known weakness of the system, safety, etc.
Foundations (functions, algorithms, models, etc.)	(Security Information and Event Management system with high performance correlation engine, but not based on specific functions/algorithms/models).
Deployment, delivery or operation perspective	Deployment of sensors and XL-SIEM agents on SMEs and XL-SIEM server running in a backend (VM/host in SME or external).
Artefacts and tools supported	Any sensor compatible with XL-SIEM plugins (logs generated by sensors/tools/artefacts that can be parsed using regular expressions). For example: firewalls, Network Intrusion Detection Systems, honeypots, etc.
Methodologies or good practices	N/A

2.1.2 Bitdefender GravityZone solution overview

Table 2. GravityZone solution overview

Partner and solution name	Bitdefender / GravityZone, Total Security
Actions performed by your solution (Detection, Prevention, Mitigation, Avoidance)	Detection of malicious files on access and on demand. Blocking malware sources like malicious URLs. Dynamic detection of unknown threats based on their actions.
Impact on SMEs (what benefits your solution brings when deployed on a SME)	Protection against malware.
Foundations (functions, algorithms, models, etc.)	Machine learning models (Support-Vector Machine, Binary Decision Trees), trained to identify malware samples. Binary programs code analysis for detecting malicious patterns. Cloud-based detection.
Deployment, delivery or operation perspective	GravityZone can be deployed on a network and provide protection for individual workstations and also centralize the results for event correlation and detecting advanced threats.
Artefacts and tools supported	Supports a wide range of platforms and operating systems.
Methodologies or good practices	All the security components must be kept up-to-date in order to ensure detection for new malware.

2.1.3 Citrix NetScaler solution overview

Table 3. NetScaler solution overview

Partner and solution name	Citrix / NetScaler (AppFirewall, Unified Gateway, SWG)
Actions performed by your solution (Detection, Prevention, Mitigation, Avoidance)	Detection, Mitigation, Endpoint protection, URL filtering
Impact on SMEs (what benefits your solution brings when deployed on a SME)	The NetScaler platform provides advanced network security solution for enterprises: AppFirewall protects enterprise web applications (and not only), Gateway provides end-to-end security between remote devices and enterprise resources, where a Secure Web Gateway inspects outgoing traffic and applies policies.
Foundations (functions, algorithms, models, etc.)	Each of the 3 products utilizes various methods in order to get the results. Most remarkably, AppFirewall uses learning algorithms for identifying patterns, heuristics for detecting possible attacks, etc.
Deployment, delivery or operation perspective	NetScaler can be deployed into the network path between an internal enterprise network (SME in this case) and the Internet, allowing bidirectional protection: AppFirewall for protecting against incoming attacks to web services, Gateway for protecting endpoints through trusted and secure connections, and SWG for securing outgoing traffic.
Artefacts and tools supported	NetScaler is configured through CLI and GUI, as well as an API (Java/Python) which supports the creation of control tools or integration to existing ones.
Methodologies or good practices	(See also comments on Deployment, Delivery, Operation) For AppFirewall: ideally all exposed web services should be protected by AppFirewall. For Unified Gateway: NetScaler can act as a network security system in front of the SME network. Employees' mobile devices that need to consume enterprise resources can connect through the Gateway product. For SWG: SWG can offer policies that restrict access to potentially malicious resources (though site reputation) or inspect outgoing HTTPS traffic (SSL Intercept). Being a new product, SWG will benefit also from SMESEC use cases.

2.1.4 EGM solution overview

Table 4. EGM TaaS solution overview

Partner and solution name	Easy Global Market / EGM TaaS
Actions performed by your solution (Detection, Prevention, Mitigation, Avoidance)	Detection
Impact on SMEs (what benefits your solution brings when deployed on a SME)	The EGM TaaS is a web service allowing users to execute test suites. Those test suites will concern security issues addressed by the SMESEC project. The tool provides full test suites to the users (security test suites have to be fully developed in the context of SMESEC). It allows SMEs to access a testing database for ensuring a certain level of confidence in the security of IoT systems. The test execution will pinpoint failing security requirements. With the associated test logs and the requirement traceability, the test will indicate what kind of vulnerabilities were found and where, thus allowing the SMEs to fix it.
Foundations (functions, algorithms, models, etc.)	Model-Based Testing is used for the test generation.
Deployment, delivery or operation perspective	Test as a service platform (EGM-TAAS) is available at two levels: The first one is online, as a web service. A client can connect to the services and execute some tests. The second, in case of private networks, is available as a hardware. EGM will purchase and install the Test as a Service platform on a device which will be sent to the client. This hardware component will be used as an internal server and allow all the private network to use the web service internally (in case of confidentiality / privacy / certification issues). In the first case, if a client wants to install the web service, he must have a Linux based machine with Docker. After that, it is merely a script that will automatically download the required images and launch them. The Docker images are on a cloud, using EGM credentials to access them. It can be manually deployed on other servers. In the second case, at EGM a manually process will be implemented while setting-up a dedicated hardware and installing the Docker images directly on the hardware instead of querying them from an online cloud.

Artefacts and tools supported	Currently, the test suites are generated offline with a Model-Based Testing tool (MBT) named CertifyIt and publisher as TTCN-3 code. TTCN-3 is an abstract test language designed for communicating systems. The test executor is based on open source project Titan, which is a TTCN-3 compiler and executor. Those two tools are not relevant for an end-user. It is a sort of black-box. We are also using Docker to deploy the container. This part can be relevant to an end-user or the SMESEC consortium. We do not currently connect the EGM-TAAS to other technologies such as bug trackers or integration tools, but this is considered as future improvements of our solution. One main additional point for security testing is the fact that some test-case may require the installation of some technology. For example, a sniffer may be implemented and installed on the network to spy on the whole system. Some test-cases may also require implementation of upper tester on components. Those upper testers shall have a defined interface by EGM and shall be able to trigger some actions in order to force the system in a particular state. Other additional component will be necessary and will be specific to the vulnerabilities tested: this is an important part which should not be overlooked.
Methodologies or good practices	EGM TaaS only requires two simple actions from a user, not really a methodology or a practice: Setup the System(s) Under Test (SUT) according to its configurations and test suites requirements. Select the tests the user is interested in order to launch them. It can be a set of test-cases or whole test suites.

2.1.5 FHNW CySec solution overview

Table 5. CySec solution overview

Partner and solution name	FHNW / Cyber Security Coach
Actions performed by your solution **(Detection, Prevention, Mitigation, Avoidance)**	Guide cyber security improvements, based on CYSFAM. Monitor adoption and adherence to practices. Elicit rationales for non-adoption and non-adherence. Allow community discussion about practices. Company and community reports.
Impact on SMEs **(what benefits your solution brings when deployed on a SME)**	Positively influences the social side of cyber security. Validation of hypotheses and discovery of needs related to cyber security behaviour of SME.

Foundations (functions, algorithms, models, etc.)	CYSFAM Goal monitoring Socio-technological alignment learning
Deployment, delivery or operation perspective	FHNW SMESEC Cloud and on premise.
Artefacts and tools supported	N/A
Methodologies or good practices	CYSFAM Practices CYSFAM-related Guidelines and Tool Recommendations

2.1.6 FORTH EWIS solution overview

Table 6. EWIS solution overview

Partner and solution name	FORTH / Early Warning Intrusion Detection System
Actions performed by your solution (Detection, Prevention, Mitigation, Avoidance)	• Early Warning Intrusion Detection System(EWIS) • Detection of DDoS attacks.
Impact on SMEs (what benefits your solution brings when deployed on a SME)	The above services can be beneficial to every SME due to the basic functionality that they offer to all computing systems. It covers: • Attacks against databases: MSSQL, MySQL, ORACLE, POSTGRES. • Attacks against communication/transfer protocols: FTP, TFTP, HTTP, HTTPS, TELNET, DNS, SMTP, MS Windows RPC, SMB. • DDOS Attacks on the SME's Infrastructure.

Foundations (functions, algorithms, models, etc.)	Uses the same basic functions and algorithms used by all low Interaction honeypot solutions and IDSes. Also, the Server-Client model is used for controlling the sensors and the configuration system. Finally, the REST API model is used between different services for communication, collaboration and alerting.
Deployment, delivery or operation perspective	As the whole honeypot system works in a virtual machine, the system is also able to operate in cloud environments. It can be operated through a provided web based control panel.
Artefacts and tools supported	Logging incidents into a database, email alerting support, SIEMs and other visualization tool cooperation.
Methodologies or good practices	N/A

2.1.7 IBM AngelEye solution overview

Table 7. AngelEye solution overview

Partner and solution name	IBM / AngelEye virtual patching
Actions performed by your solution (Detection, Prevention, Mitigation, Avoidance)	Automatically builds a virtual patch solution with ahead of threat capabilities that can be integrated in IDS/IPS or Endpoint solutions.
Impact on SMEs (what benefits your solution brings when deployed on a SME)	AngelEye can provide virtual patching for application and libraries that are not covered by any commercial tool, or open-source virtual patch library (like SNORT). This is extremely useful for creating a virtual patch for applications installed on IoT devices.
Foundations (functions, algorithms, models, etc.)	Applies machine learning on the tests created by a hybrid test generation platform (genetic, generation and symbolic) to comprehensive virtual patch of the system under test.

Deployment, delivery or operation perspective	The virtual patch can be integrated into a variety of systems that attempt to predict if a given file can exploit a vulnerability in an application. The patch can replace any rules or pattern matching technique. Examples for integration: IDS/IPS (both network or host), Endpoint protection.
Artefacts and tools supported	N/A
Methodologies or good practices	N/A

2.1.8 IBM Anti-ROP solution overview

Table 8. Anti-ROP solution overview

Partner and solution name	IBM / Anti-ROP
Actions performed by your solution **(Detection, Prevention, Mitigation, Avoidance)**	A prevention and detection technique deployed on the endpoint software application against ROP and memory corruptions attacks.
Impact on SMEs **(what benefits your solution brings when deployed on a SME)**	Anti-ROP is moving target defence that can provide protection for application and libraries resulting in creating unique libraries and devices since each device has its own version of the code even that the software preserves the same functionality. This is extremely useful for creating a protection mechanism for applications installed on IoT devices.
Foundations (functions, algorithms, models, etc.)	Two possible implementations: • Analysing binary code and randomizing it; • Running block randomization while compiling source code.
Deployment, delivery or operation perspective	• Anti-ROP for binary, where this tool is based on disassembly tool like IDA which randomizes the binary application in memory; • Anti-ROP for source, where a compiler plugin is provided that randomizes the intermediate representation of the compiler.

Artefacts and tools supported	Anti-ROP can be developed to all platforms, we have developed two branches: • Anti-ROP for binary is developed to Windows 32/64 bit and uses Hex-Ray Ida to in analysing PE files; • Anti-ROP for source is developed to Linux with Clang LLVM compiler versions 3 and 4.
Methodologies or good practices	N/A

2.2 Technical specifications and requirements

2.2.1 ATOS XL-SIEM technical specifications

Table 9. XL-SIEM technical specifications

Hardware, Firmware, or Software solution	Software solution
Maturity/readiness level (Production Status)	TRL6
Final Product or Simple Module / Artefact	Complete Product
Third Party Tools Integration Availability	Integration of events/logs compatible with XL-SIEM agents.
Communications standards (LAN, Bluetooth, WAN, GSM…) and protocols (HTTP, FTP, SMTP…)	Communication between XL-SIEM agents and server using LAN. XL-SIEM web interface using HTTPS.
For Hardware Solutions: Technical specifications (RAM, HDD, Screen, Interfaces (HDMI, USB, …), Battery Life…)	N/A

For Software Solutions: Deployment model (Platform Requirement, OS, Application Environment, Hardware Requirements)	XL-SIEM main node (backend server): Requirements: OSSIM + Apache Storm RAM: 24 GB CPU: 8-cores (optional) Additional XL-SIEM worker nodes: Requirements: Apache Storm RAM: 16 GB CPU: 4-cores OS: Linux Debian/Ubuntu XL-SIEM agent node (Sensors + Agent): RAM: 8GB CPU: 4-cores OS: Linux Debian/Ubuntu/CentOS
Documentation Available (Yes/No)	Yes (installation / user manuals)
Benchmarking, Production Issues	According to EsperTech benchmark [2,3] Esper (the CEP included in XL-SIEM) "exceeds over 500 000 event/s on a dual CPU 2GHz Intel based hardware, with engine latency below 3 microseconds average (below 10us with more than 99% predictability) on a VWAP benchmark with 1000 statements registered in the system - this tops at 70 Mbit/s at 85% CPU usage.". We have tested that XL-SIEM is able to detect 1000 concurrent attacks with delay < 0,5 seconds in each correlation process, where each attack is composed by seven events sent in a period of 6 seconds.

2.2.2 Bitdefender GravityZone technical specifications

Table 10. GravityZone technical specifications

Hardware, Firmware, or Software solution	Software solution
Maturity/readiness level (Production Status)	TRL 8/9
Final Product or Simple Module / Artifact	Final product
Third Party Tools Integration Availability	Yes. SDK available.
Communications standards (LAN, Bluetooth, WAN, GSM…) and protocols (HTTP, FTP, SMTP…)	• LAN • HTTP

For Hardware Solutions: Technical specifications (RAM, HDD, Screen, Interfaces (HDMI, USB, …), Battery Life…)	N/A
For Software Solutions: Deployment model (Platform Requirement, OS, Application Environment, Hardware Requirements)	• Workstation operating systems - Windows 7, 8, 8.1, 10 - Windows Vista (SP1, SP2), Windows XP (SP3) - Mac OS X Lion (10.7.x), Mountain Lion (10.8.x), Mavericks (10.9.x), Yosemite (10.10.x), El Capitan (10.11.x) • Tablet and embedded operating systems - Windows Embedded Standard, POSReady, 2009, 7 - Windows Embedded Enterprise 7 - Windows XP Embedded (SP 2), Tablet PC Edition • Server operating systems - Windows Server 2012, 2012 R2 - Windows Small Business Server (SBS) 2008, 2011 - Windows Server 2008, 2008 R2 - Windows Small Business Server (SBS) 2003 - Windows Server 2003 (SP 1), 2003 R2 - Windows Home Server - Red Hat Enterprise Linux / CentOS 5.6 or higher, Ubuntu 10.04 LTS or higher, SUSE - Linux Enterprise Server 11 or higher, OpenSUSE 11 or higher, Fedora 15 or higher, - Debian 5.0 or higher, Oracle Solaris 11, 10 (only in VMware vShield environments) • Mobile operating systems - Apple iPhones and iPad tablets (iOS

	5.1+) - Google Android smartphones and tablets (2.2+) • Virtualization solutions - VMware vSphere 6.0. 5.5, 5.1, 5.0 P1 or 4.1 P3 ESXi 4.1, 5.0, 5.1, 5.5 - VMware vCenter Server 6.0, 5.5, 5.1, 5.0 or 4.1 - VMware vShield Manager 5.5, 5.1, 5.0 - VMware vShield Endpoint - VMware vCNS 5.5 - VMware Tools 8.6.0 build 446312 - VMware View 5.1, 5.0 - Citrix XenDesktop 5.5, 5.0 - Citrix XenServer 6.0, 5.6 or 5.5 including Citrix Xen Hypervisor - Citrix VDI-in-a-Box 5.x - Microsoft Hyper-V Server 2012, 2008 R2 including Microsoft Hyper-V Hypervisor - Red Hat Enterprise 3.0 including Red Hat KVM Hypervisor - Oracle VM 3.0
Documentation Available (Yes/No)	Yes
Benchmarking, Production Issues	External benchmarks performed by Antivirus testing companies like AV-Test, AV Comparatives.

2.2.3 Citrix NetScaler technical specifications

Table 11. NetScaler technical specifications

Hardware, Firmware, or Software solution	Hardware (MPX/SDX model) and Software: Virtualized (VPX) and in containers (CPX – though some functions are not implemented here)
Maturity/readiness level (Production Status)	TRL 6 for AppFirewall, Gateway, TRL 2-3 for SWG
Final Product or Simple Module / Artifact	Final commercially available product.

Third Party Tools Integration Availability	For configuration, an API is available and the available SDK is called "Nitro". For output, NetScaler produces AppFlow records which are IPFIX compatible (RFC 7011) as well as statistics available through the Nitro API.
Communications standards (LAN, Bluetooth, WAN, GSM…) and protocols (HTTP, FTP, SMTP…)	Mainly HTTP, HTTPS. Mobile clients can connect over mobile network technologies to Gateway (e.g. HTTPS over 3G/4G).
For Hardware Solutions: Technical specifications (RAM, HDD, Screen, Interfaces (HDMI, USB, …), Battery Life…)	NetScaler as hardware solution runs on dedicated hardware (MPX or SDX models), and cannot run on e.g. PC-based architecture.
For Software Solutions: Deployment model (Platform Requirement, OS, Application Environment, Hardware Requirements)	VPX is the virtualized version of NetScaler and is offered for XenServer/VMware/Hyper-V/KVM. Hardware requirments for VPX: 2 CPU cores, minimum 2GB RAM (recommended 4GB), 16GB disk space recommended. CPX can run on Docker containers or in a Kubernetes-based architecture.
Documentation Available (Yes/No)	Yes https://www.citrix.com/products/NetScaler-appfirewall/; https://www.citrix.com/products/NetScaler-unified-gateway/; https://www.citrix.com/products/NetScaler-secure-web-gateway/;
Benchmarking, Production Issues	Various performance benchmarks are available on the Internet, a Google search will return several results. NetScaler is already deployed in thousands of enterprises, including organizations and telcos that serve millions of users every day.

2.2.4 EGM technical specifications

Table 12. EGM's TaaS platform technical specifications

Hardware, Firmware, or Software solution	Test-as-a-Service Platform, or EGM Hardware
Maturity/readiness level (Production Status)	TRL 6
Final Product or Simple Module / Artifact	Final Product
Third Party Tools Integration Availability	At this moment, no third-party tools integration is available. However, we are planning to connect some tools such as Bug Tracker or Jenkins CI.
Communications standards (LAN, Bluetooth, WAN, GSM…) and protocols (HTTP, FTP, SMTP…)	A REST API is used for communicating with the TAAS platform. Additionally, many protocols may be used during the test execution, depending on the test suite and the System Under Test (SUT). Those protocols may include HTTP, CoAP, MQTT and others.
For Hardware Solutions: Technical specifications (RAM, HDD, Screen, Interfaces (HDMI, USB, …), Battery Life…)	The hardware will be provided by Easy Global Market. Basically, it's a small computer with at least: 4GB of Ram; 80Go HDD; HDMI/VGA output; Ethernet; CPU Intel Core I3 2.4Ghz; No screen, keyboard or mouse provided.
For Software Solutions: Deployment model (Platform Requirement, OS, Application Environment, Hardware Requirements)	TAAS platform will be available on an elastic cloud provider. Each node will require Linux operating system such as Debian (min version 8) and with Docker installed. A master node will orchestrate the cloud with Kubernetes.
Documentation Available (Yes/No)	Yes

Benchmarking, Production Issues	N/A

2.2.5 FHNW CySec technical specifications

Table 13. CySec technical specifications

Hardware, Firmware, or Software solution	Software
Maturity/readiness level (Production Status)	TRL4 at this moment.
Final Product or Simple Module / Artifact	Product
Third Party Tools Integration Availability	PIWIK SUPERSEDE
Communications standards (LAN, Bluetooth, WAN, GSM…) and protocols (HTTP, FTP, SMTP…)	HTTPS, REST
For Hardware Solutions: Technical specifications (RAM, HDD, Screen, Interfaces (HDMI, USB, …), Battery Life…)	N/A
For Software Solutions: Deployment model (Platform Requirement, OS, Application Environment, Hardware Requirements)	VM
Documentation Available (Yes/No)	Partially

Benchmarking, Production Issues	N/A

2.2.6 FORTH EWIS technical specifications

Table 14. EWIS technical specifications

Hardware, Firmware, or Software solution	Software
Maturity/readiness level (Production Status)	TRL7(EWIS) TRL4(DDOS)
Final Product or Simple Module / Artifact	The system is already being used in production mode by several organizations. DDOS system is currently under development and early testing.
Third Party Tools Integration Availability	SIEM systems can be easily integrated with the solution provided. Logs can be easily transformed in various formats. Other visualization tools can use the output of our system to produce alerts and security reports.
Communications standards (LAN, Bluetooth, WAN, GSM…) and protocols (HTTP, FTP, SMTP…)	LAN. FTP, TFTP, HTTP, HTTPS, TELNET, DNS, SMTP, MS Windows RPC, SMB.
For Hardware Solutions: Technical specifications (RAM, HDD, Screen, Interfaces (HDMI, USB, …), Battery Life…)	N/A
For Software Solutions: Deployment model (Platform Requirement, OS, Application Environment, Hardware Requirements)	No specific HW. Linux system. Minimum HW requirements depend on the amount of the monitored IP addresses. e.g. Monitoring 1500 IP addresses need 1.5TB of storage per year. RAM:>=4GB NIC: 1x1Gbps (for monitoring purposes) and 1x100Mbps (for management purposes).

Documentation Available (Yes/No)	Yes
Benchmarking, Production Issues	N/A

2.2.7 IBM AngelEye technical specifications

Table 15. AngelEye technical specifications

Hardware, Firmware, or Software solution	Software
Maturity/readiness level (Production Status)	TRL1/2
Final Product or Simple Module / Artifact	N/A
Third Party Tools Integration Availability	N/A
Communications standards (LAN, Bluetooth, WAN, GSM…) and protocols (HTTP, FTP, SMTP…)	N/A
For Hardware Solutions: Technical specifications (RAM, HDD, Screen, Interfaces (HDMI, USB, …), Battery Life…)	N/A
For Software Solutions: Deployment model (Platform Requirement, OS, Application Environment, Hardware Requirements)	The tool is developed for Linux or similar OS.

Documentation Available (Yes/No)	No
Benchmarking, Production Issues	We have conducted preliminary experiments that show promising results of ahead-of-threat detection

2.2.8 IBM Anti-ROP technical specifications

Table 16. Anti-ROP technical specifications

Hardware, Firmware, or Software solution	Software
Maturity/readiness level (Production Status)	TRL 4 – 5
Final Product or Simple Module / Artifact	N/A
Third Party Tools Integration Availability	N/A
Communications standards (LAN, Bluetooth, WAN, GSM…) and protocols (HTTP, FTP, SMTP…)	N/A
For Hardware Solutions: Technical specifications (RAM, HDD, Screen, Interfaces (HDMI, USB, …), Battery Life…)	N/A
For Software Solutions: Deployment model (Platform Requirement, OS, Application Environment, Hardware Requirements)	The tool is developed for Linux or similar OS or any open source OS and compiler.

Documentation Available (Yes/No)	No
Benchmarking, Production Issues	We have deployed Anti-ROP (for binary) solution at IBM HRL where Mozilla Firefox, VLC and Adobe reader where shuffled. In addition, we have developed isomorphism tool that check call graphs between binaries to validate the correctness of the shuffling.

2.3 Solution architecture and details

2.3.1 ATOS XL-SIEM solution architecture

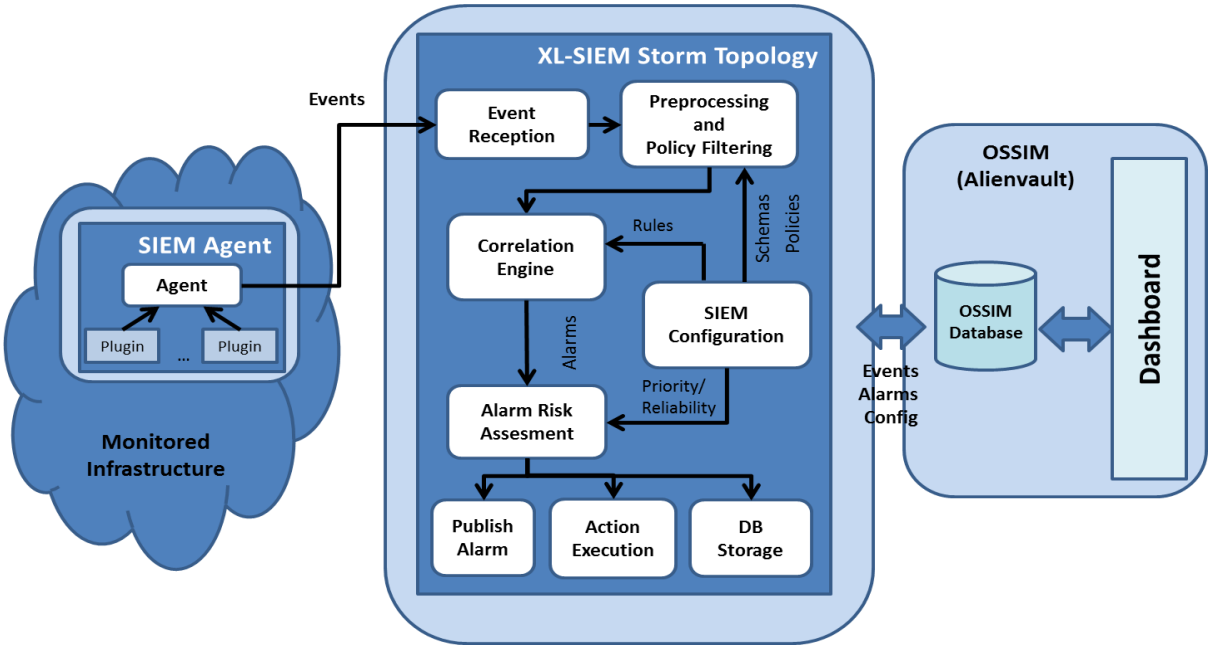

Figure 1. XL-SIEM solution architecture

The XL-SIEM developed by Atos uses as basis the open source AlienVault SIEM OSSIM [1]. The SIEM Agent is extended with different functionalities (such as support for anonymization/encryption of fields and usage of TLS certificates in the communication or support to send the events using the Advanced Message Queuing Protocol (AMQP) to a RabbitMQ Server) whereas the processing performed by the OSSIM server is replaced by a set of new processes running in an Apache Storm topology, including a high-performance correlation engine based on the Esper library [2].

The deployment of the XL-SIEM is done in an open source environment called Apache Storm [4]. We choose this environment due to its support for processing/managing streams of data and its

compatibility with several programming languages, which would facilitate its work with any domain of application.

The architecture of the XL-SIEM is presented in Figure 1. The "Monitored Infrastructure" is in charge of collecting the data using the SIEM Agents. They are deployed in the system under monitoring (e.g. end-user devices, servers, etc.). The events are then sent to the XL-SIEM, which processes and inspects them in order to identify anomalies. This is done by the "Correlation Engine", which is configured by the XL-SIEM administrators. Finally, events and alarms are stored in the same databases used by OSSIM (which have been extended and adapted to support the storage of the information required by the new XL-SIEM processing in Apache Storm) for visualization and access of the users of the XL-SIEM.

2.3.2 Bitdefender GravityZone solution architecture

The figure below shows the architecture for a network protected by Bitdefender GravityZone. The central component is GravityZone Control Center that controls the components deployed on

- Endpoints
- Mobile devices
- Virtualized environments

The scanning results can be integrated with third party tools through the API or they can be visualized in the Web-based user interface.

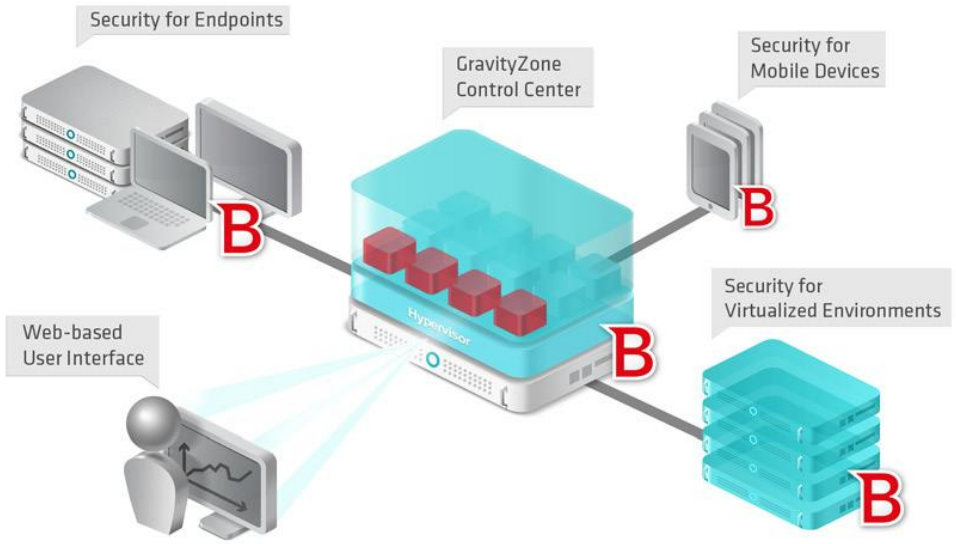

Figure 2. GravityZone solution architecture

2.3.3 Citrix NetScaler solution architecture

The overall architecture of how the contributed products work is shown in Figure 3. The following figures show where AppFirewall, Gateway and SWG are positioned in the network path.

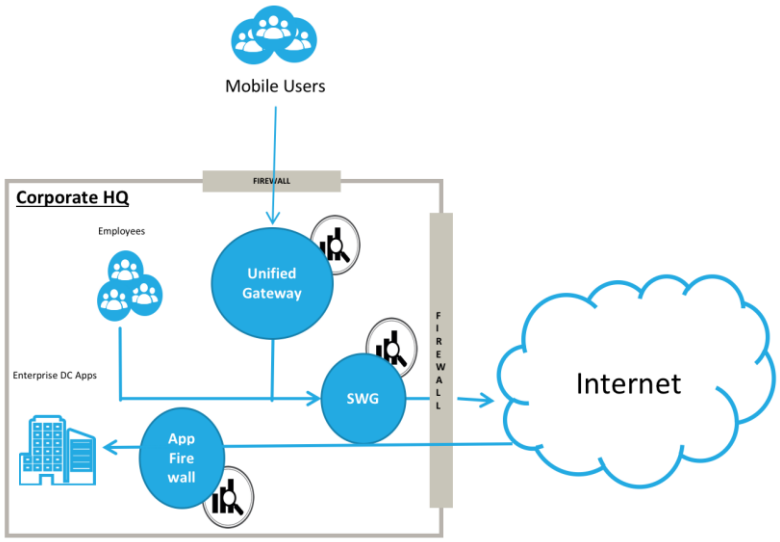

Figure 3. Citrix's overall architecture of contributed products

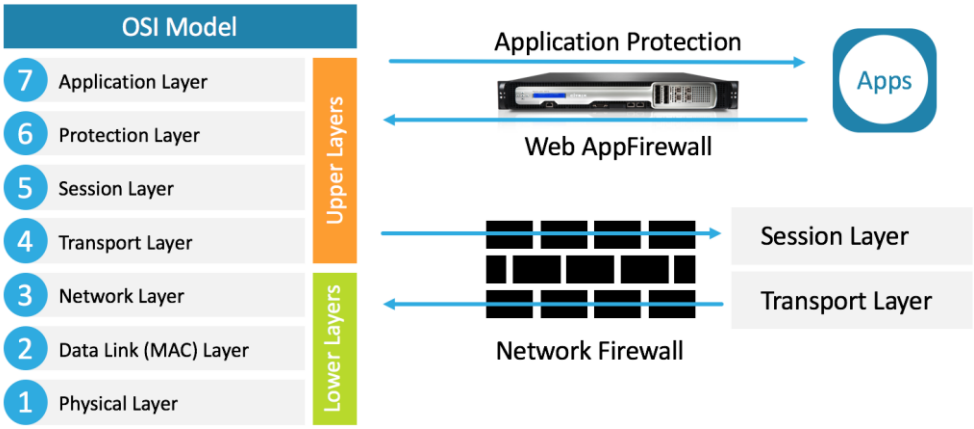

Figure 4. App Firewall provides application protection, above network layer

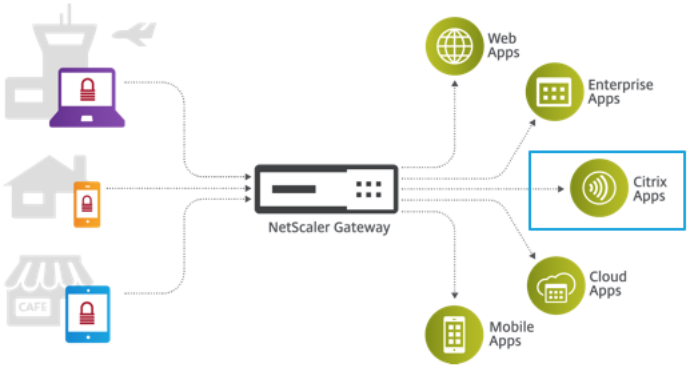

Figure 5. NetScaler Gateway allows access from any device to trusted apps

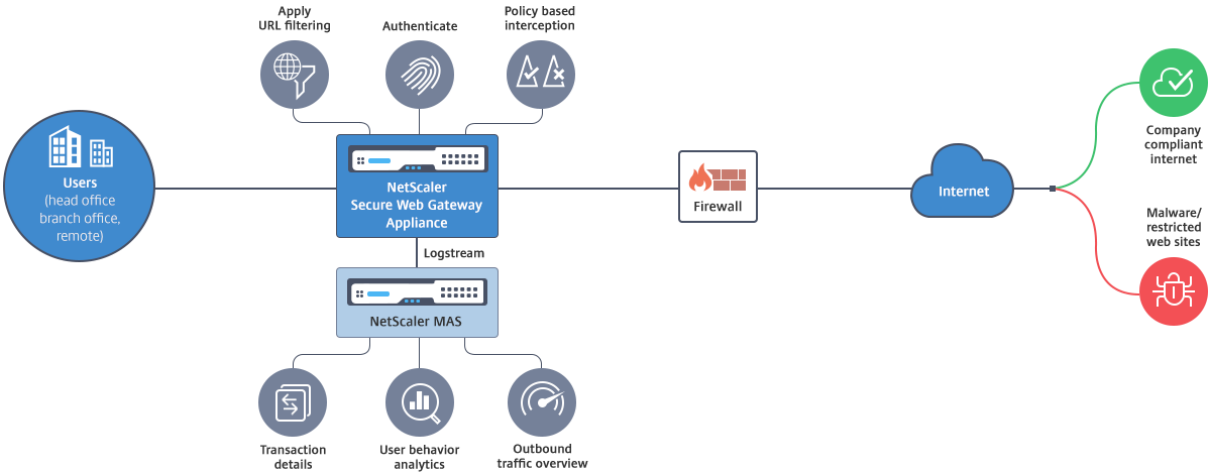

Figure 6. NetScaler Secure Web Gateway examining outgoing traffic

2.3.4 EGM solution's architecture

As described earlier we are offering two solutions:

- an online testing solution, where EGM-TAAS is an online server on a cloud anyone can connect to,
- and an offline testing solution, where we provide a client hardware version of the TAAS serve that may be included on a private network without requiring any external communications.

Our solution is also designed to provide a Continuous Integration test mechanism, allowing end users to benefit from the last versions of the test suites.

Online testing

The Online testing will allow user to setup their System Under Test (SUT) configuration online. Then they will be able to launch the tests execution without any kind of installation on their own machine.

End users just have to connect to the web application, access the SUT configuration and parameters, select some test-cases or whole test suites they are interested in confronting their systems against, and launch the execution. The tool will automatically take care of the execution and generate readable reports in the web interface. The users can then access the reports, statistics and details about the failures.

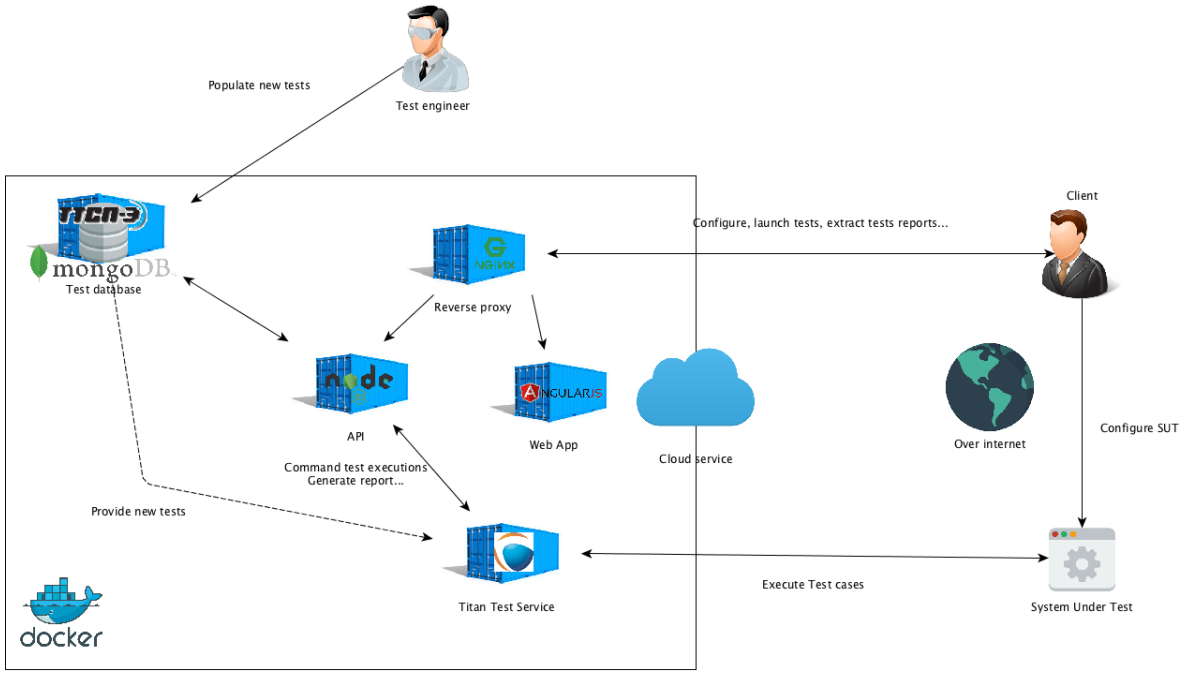

Figure 7. The relations between services components and the users

The service is composed of Docker containers orchestrated by Kubernetes.

Figure 7 describes the relations between our services' components and the users (both the client and theirs SUT).

Nginx

The Nginx container will act as a reverse proxy to serve Web App content and an access to the API for the client.

API

The API is basically the main service, responsible of fetching and providing data to the Webapp and Titan service. It also manages users, authentication, test execution, report generation etc.

Powered by **Nodejs** and the express framework.

Web Application

This container is in charge of serving web site static files to users' browsers. The app is written in AngularJS2, using API endpoints to fetch data.

Powered by **AngularJS** with a **Nginx** reverse proxy behind.

Database

The database service contains every required data for running the user applications: users, tests metadatas, SUT configuration etc., excepting the TTCN-3 tests (or test suites), which are binary files.

Thus, they are not physically present inside the database. Instead, they are hosted on another particular file system which we access when required.

Powered by **MongoDB**.

Titan test service

This container is managing test execution. It provides the technology to execute the TTCN-3 tests cases present in our database. Logs, such as data payloads between Titan and the SUT, generate Junit XML reports which are being sent back to the TAAS API, allowing the web interface to read, compile and display the results in a suitable format for the users.

Powered by **Eclipse Titan** https://projects.eclipse.org/projects/tools.titan.

Offline testing

The offline testing solution provides the same functionality as the online testing solution with one exception. All technical solutions are embedded on single computer hardware (EGM hardware). This hardware allows the client to test his system from inside his own private network without the need to publicly expose his SUT on the internet for privacy, security or confidentiality concerns.

The only moment that requires an internet connection is the setup of the EGM hardware requiring to control the license to fetch the latest available test suites provided by our test engineer.

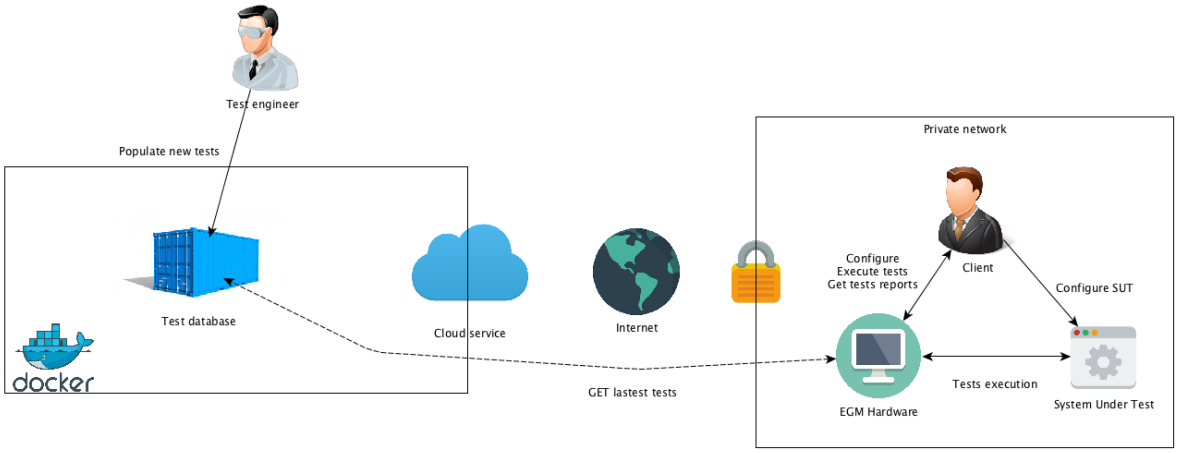

Figure 8. The offline testing

A deepest look in the EGM Hardware, will show a nearly same technical environment as the cloud solution. Except that the tests aren't stored (from the beginning) in a local database system and will depend on the user licence.

2.3.5 FHNW CySec solution architecture

The solution itself is a reference implementation for the API Systems such as www.smesec.eu, monitoring.smesec.eu, and www.test.smesec.eu build the public infrastructure part of the project. These parts provide a centralized, cloud capable platform to customers offering a low effort entry to the system. Customers concerned about privacy and confidentiality of their data may build their own instances of the software controlling all data flow to the project.

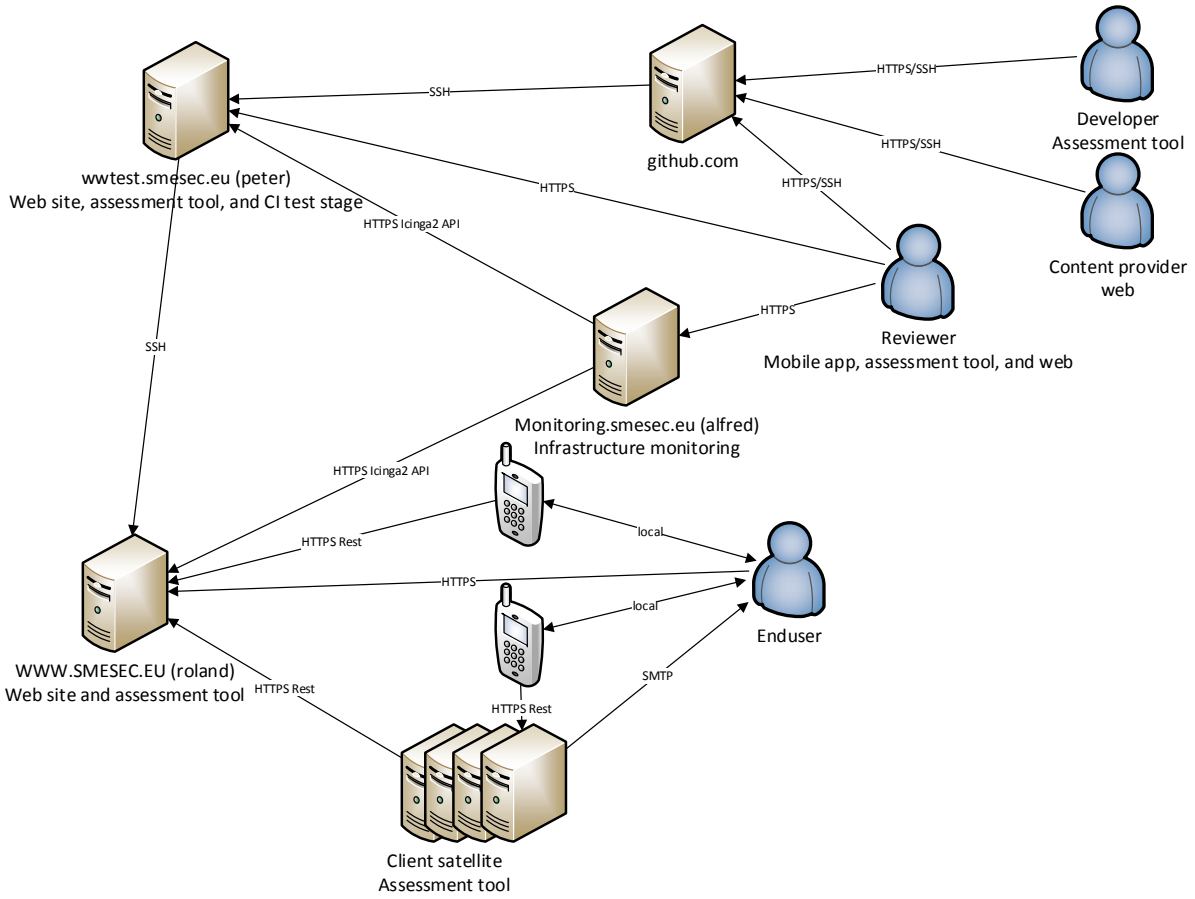

Figure 9. CySec solution architecture

SMEs may use the system with or without any infrastructure and free of charge. All sources are open. Deployment can be done but is not required. Minimum platform is a computer and a web browser such as Firefox.

2.3.6 FORTH EWIS solution architecture

The solution proposed by FORTH provides to the system administrator the option to create an infrastructure of honeypot VM instances namely sensors. These sensors are able to monitor and capture potential attack attempts and transfer that information in a central database in real time. Moreover, the DDoS detection system will be able to detect Denial of Services attacks and provide the appropriate alerts. The whole system is formed by two distinct components. The honeypot VMs and the control panel used for management purposes. The control panel is responsible for the command and control of all the honeypot sensors and the visualization of the attack incidents. Additionally, a component which has access to the central database where all the attack records are stored exists.

2.3.7 IBM AngelEye solution architecture

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.

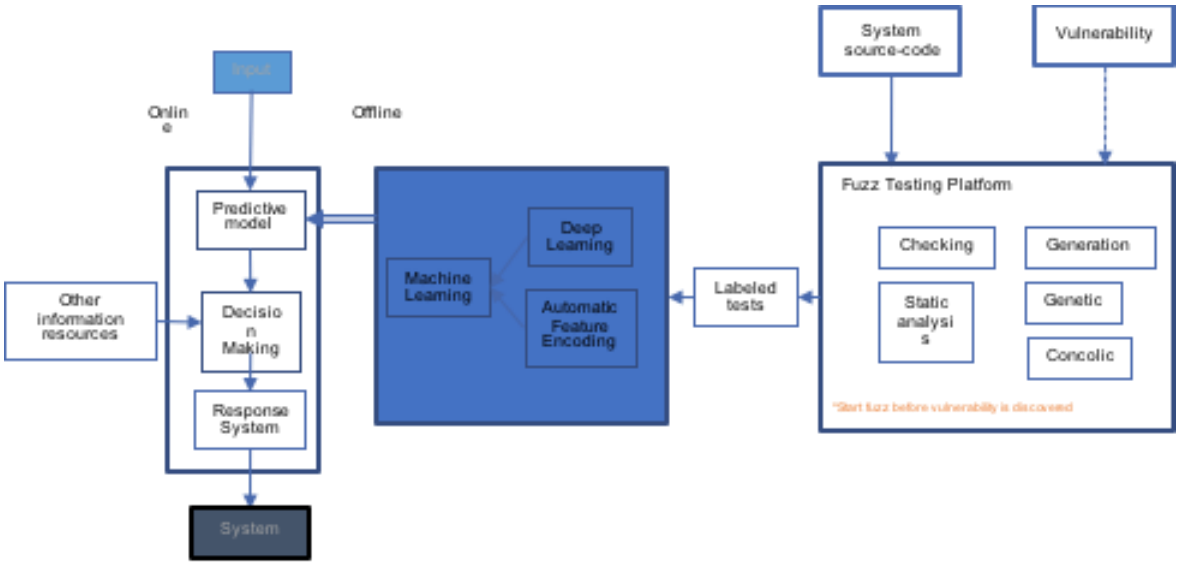

Figure 10. AngelEye solution architecture

2.3.8 IBM Anti-ROP solution architecture

Anti-ROP for binary

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.

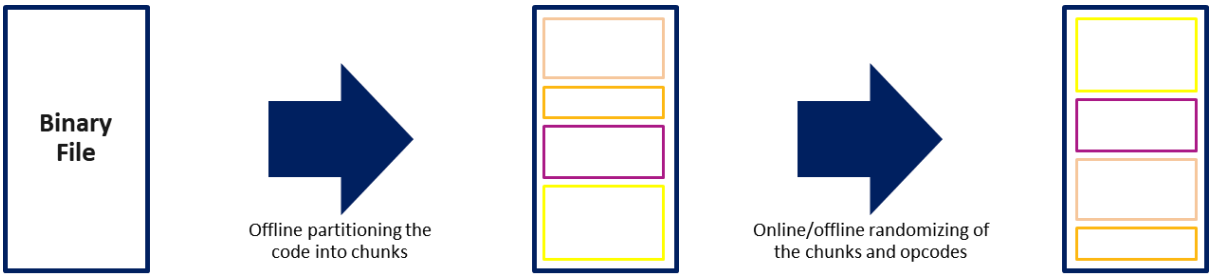

Figure 11. Anti-ROP for binary

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.

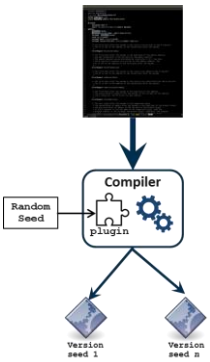

Figure 12. Anti-ROP for source

2.4 Inputs and outputs

2.4.1 ATOS XL-SIEM inputs and outputs

Inputs

The inputs of the XL-SIEM are provided by the agents deployed in the systems under monitoring (e.g. end-user devices, servers, etc.). They collect data and send directly to the XL-SIEM, which analyses it for correlation with the rules defined by the administrators. The agents are built using as basis SIEM agents provided by AlienVault OSSIM [1] (which provides them as open source). Therefore, how the agents obtain and parse the information they collect is supported by plugins. These components of the agents transform the security events they obtain from the target system/device into a normalized format that can later be used for correlation in the XL-SIEM. The formatting is done using as basis regular expressions, so this is a required characteristic for their input (logs and events collected must be provided as regular expressions).

Together with this input format we extended the XL-SIEM to support additional ones (well-known in the area). The objective was to increase its adoption so we added support to the following ones:

- STIX (Structured Threat Information eXpression) [5]: this format is an open source structured language used for exchanging cyber threat intelligence. The format is machine-ready, which allows for a automatic generation, analysis and reaction of the information. STIX also supports collaborative threat analysis, automated threat exchange, etc. The plugin for the XL-SIEM is able to transform this format to OSSIM format maintaining all its information.

- JSON (JavaScript Object Notation) [6]: this is a lightweight format used for data-exchange (in general, not focusing in cyber security as STIX does). The format is language independent so it is widely used in many systems and domains.

Outputs

The outputs generated by the XL-SIEM are of two different types: events and alarms. On the one hand, the events are used for storing/ the information of anomalous behaviour in the system under monitoring or informing the SIEM server. The events are generated by the XL-SIEM agents. On the other hand, the alarms inform in real-time users of malicious activities detected in the logs (which are obtained from the events) according to the rules defined by the administrators. The alarms are generated by the XL-SIEM.

Events:

The events created by the XL-SIEM can be of three different types:

- A string (after normalization) with information of the event to a server;
- A file in CVS format;
- Data in JSOM format.

Alarms:

The alarms created by the XL-SIEM are provided/supported in two different ways:

- Managed in a MySQL database, which can send it either in a JSON format or as a string using a Data Distribution Service (DDS);
- Invoke a Distributed Remote Procedure Call service that provides the information also in JSON format.

Additionally, the events and alarms can always be visualized in the graphical web interface of the XL-SIEM.

2.4.2 Bitdefender GravityZone inputs and outputs

Inputs

Bitdefender tools are used to scan the hard drive, the memory and the network shares for malicious activity. The following scan modes are available:

- Quick Scan is preconfigured to allow scanning only critical Windows and Linux system locations. Running a Quick Scan usually takes less than a minute and uses a fraction of the system resources needed by a regular virus scan. This mode only detects existing malware, without taking any action. If malware is found during a Quick Scan, a Full System Scan task is required to remove detected malware.
- Full Scan checks the entire system for all types of malware threatening its security, such as viruses, spyware, adware, rootkits and others.
- Memory Scan checks the programs running in the virtual machine's memory.
- Network Scan is a type of custom scan, allowing to scan network drives using the Bitdefender security agent installed on the target virtual machine. For the network scan task to work, one

needs to assign the task to one single endpoint in the network and needs to enter the credentials of a user account with read/write permissions on the target network drives, for the security agent to be able to access and take actions on these network drives. The required credentials can be configured in the Target tab of the tasks window.

- Custom Scan allows the user to choose the locations to be scanned and to configure the scan options.

Outputs

Bitdefender Control Center APIs allow developers to automate business workflows. The APIs are exposed using JSON-RPC 2.0 protocol. The API calls are performed as HTTP requests with JSON-RPC messages as payload. HTTP POST method MUST be used for each API call. Also, it is required that each HTTP request have the Content-Type header set to application/json.

2.4.3 Citrix NetScaler inputs and outputs

Inputs

NetScaler platform can accept input through:

- GUI (embedded in the system);
- CLI (command line access);
- Nitro API.

Outputs

- AppFlow (IPFIX) records;
- Stats, command results (through Nitro API).

2.4.4 EGM inputs and outputs

Inputs

As input, the EGM Test As A Service platform (EGM-TAAS) requires two distinct information:

- First, the System Under Test (SUT) information. This contains for instance IP address, protocols, etc. All the data which are required for running the tests and communicating with the SUT. We can provide an encryption key here in some cases, as we would give our public encryption key, for the security testing.
- The second input is a selection of test cases (or whole test suites) by the user. Some tests may not be relevant at all, because the corresponding protocol has no meaning for example. It is, in all cases, basically a set of test cases the user is interested in.

Outputs

With those inputs, we can launch the test and wait for the execution results. Thus, the user will receive:

The test results (pass / fail / inconc) for all test cases which were executed.

- The test reason, in case of failure, the TTCN-3 code contains some sort of log to indicate the user why the test failed. This can be a first indicator to isolating the vulnerability.

- The requirement(s) associated with the test. They are available at any moment but the user will know the reference of the failing security requirement and will be able to read a description of what the failing test-case was trying to verify.
- The exchanges with the SUT, a summary of all messages sent and received by Titan executor are compiled so that the user can see some details about the communication.
- Eventually, the user can access the test logs, but it comes with a certain level of difficulty and it may not be necessary. All relevant parts shall be visually accessible through the web interface.
- Some diagrams and compilations of results are also available for the user to grasp the evolution of his system when checked against the tests' results.

Every inputs and outputs are going through the TAAS web interface which is using a specific format. EGM TAAS is using REST API.

SUT Configuration

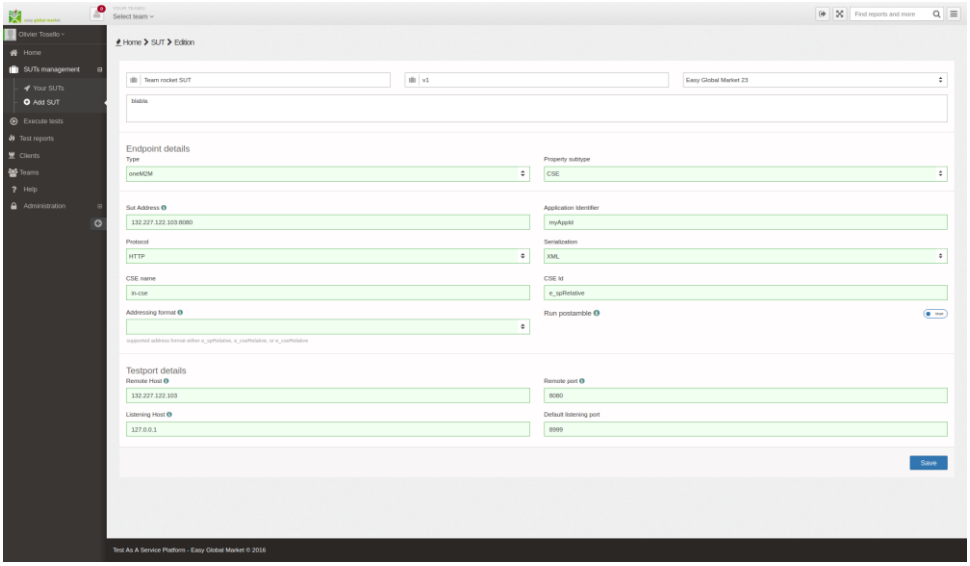

Figure 13. SUT dashboard

The System Under Test configuration is composed of two parts:

- The first (in blank), describe the SUT for the user (name, version, user associated, description). Those fields are not directly relevant for the testing part, but allow clear identification.
- The second part describes the SUT for the test executor. We select a type (here oneM2M) and the fields change according to it. The user must then customize the parameters so that the test execution goes smoothly. We can find IP addresses (v4 or v6), protocols, serialization, ports, etc.

SUT View

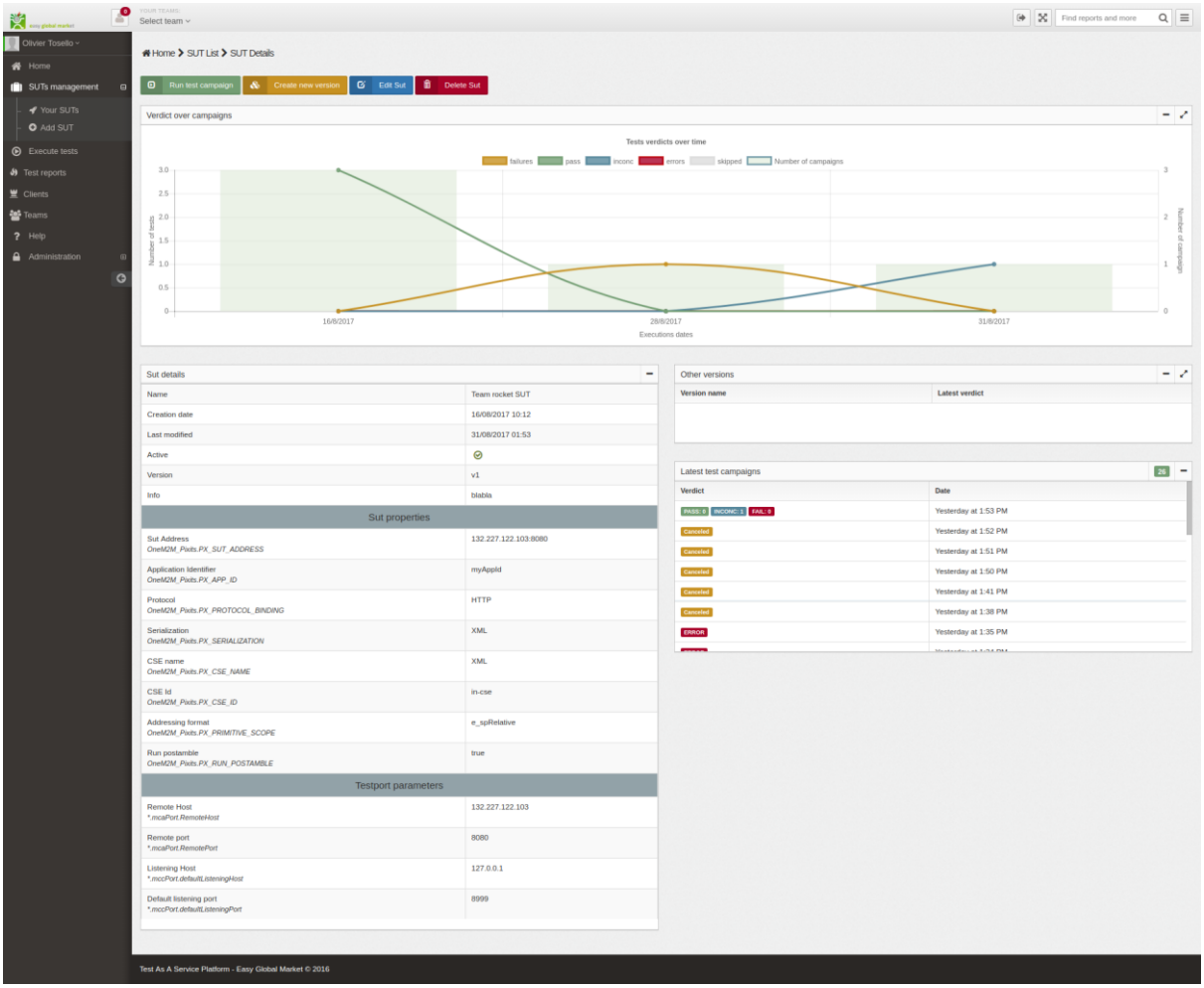

Figure 14. SUT dashboard presenting details

This is the full SUT view; it is basically composed of three parts:

- The evolution of test verdicts over time.
- The detail of the current SUT configuration.
- The latest results.

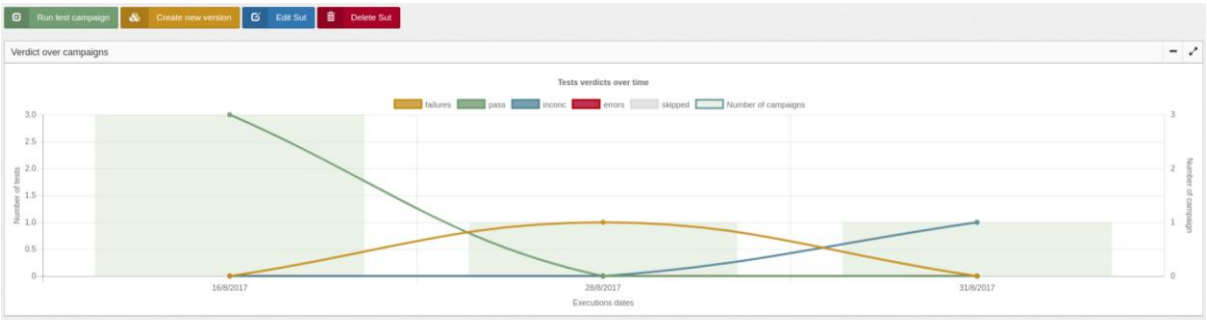

Figure 15. Graphic illustrating tests results

The graphic above shows the evolution we have when testing the current SUT. It displays the statistics of the verdicts and allows the users to determine if we have some kind of improvements over time.

Sut details

Name	Team rocket SUT
Creation date	16/08/2017 10:12
Last modified	31/08/2017 01:53
Active	✓
Version	v1
Info	blabla
Sut properties	
Sut Address *OneM2M_Pixits.PX_SUT_ADDRESS*	132.227.122.103:8080
Application Identifier *OneM2M_Pixits.PX_APP_ID*	myAppId
Protocol *OneM2M_Pixits.PX_PROTOCOL_BINDING*	HTTP
Serialization *OneM2M_Pixits.PX_SERIALIZATION*	XML
CSE name *OneM2M_Pixits.PX_CSE_NAME*	XML
CSE Id *OneM2M_Pixits.PX_CSE_ID*	in-cse
Addressing format *OneM2M_Pixits.PX_PRIMITIVE_SCOPE*	e_spRelative
Run postamble *OneM2M_Pixits.PX_RUN_POSTAMBLE*	true
Testport parameters	
Remote Host **.mcaPort.RemoteHost*	132.227.122.103
Remote port **.mcaPort.RemotePort*	8080
Listening Host **.mccPort.defaultListeningHost*	127.0.0.1
Default listening port **.mccPort.defaultListeningPort*	8999

Figure 16. SUT configuration

The SUR configuration is a quick reminder of all the information we setup during the SUT configuration.

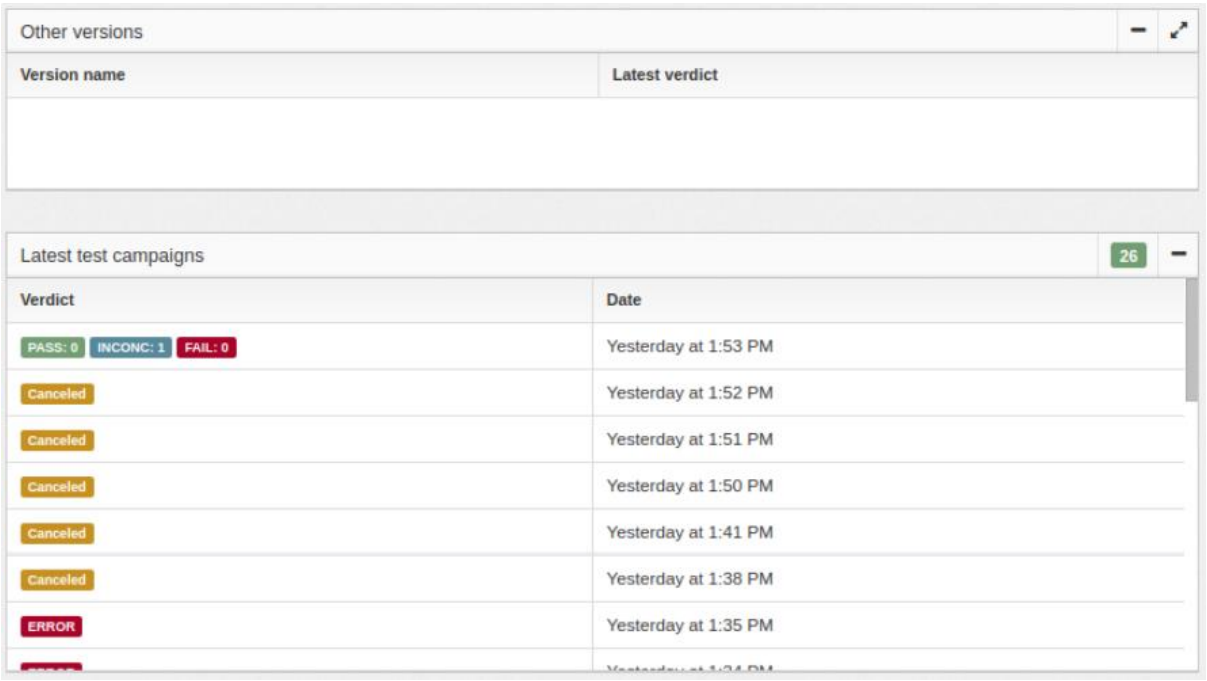

Figure 17. SUT testing results considering version and time

This part shows another way to look at the test results. With version and time information, as well as information about canceled execution.

Test execution

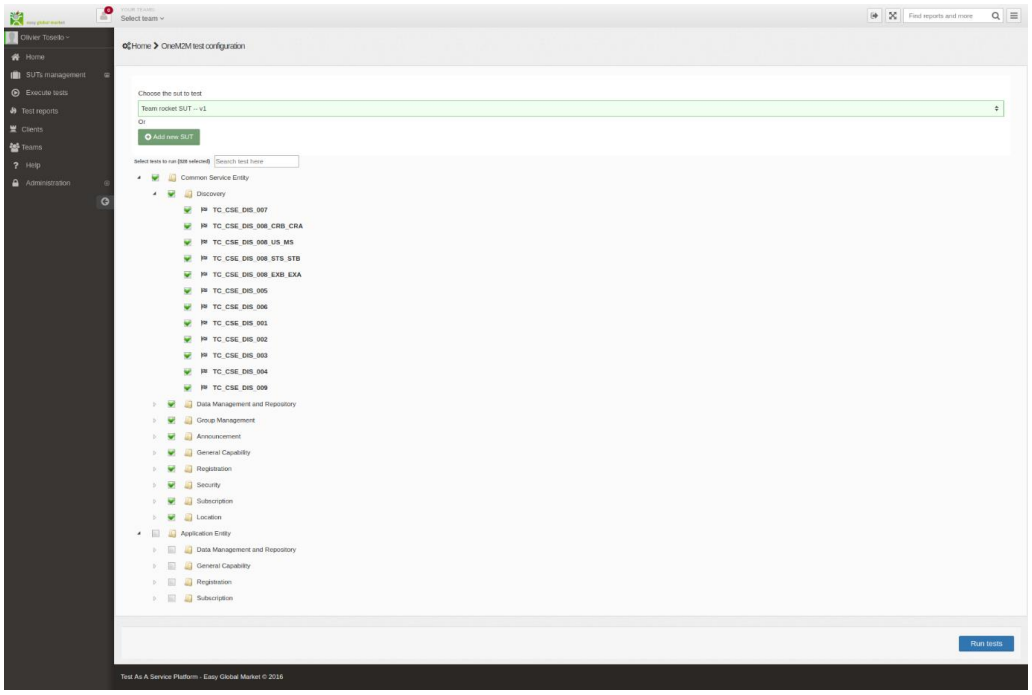

Figure 18. Dashboard illustrating test parameters

The test execution is quite simple, we select (or create) an SUT. Then, we can select (manually test-case by test-case, whole test-suite) or search which tests should be executed. Then we can click on the "run" button and wait for the result.

Test reports

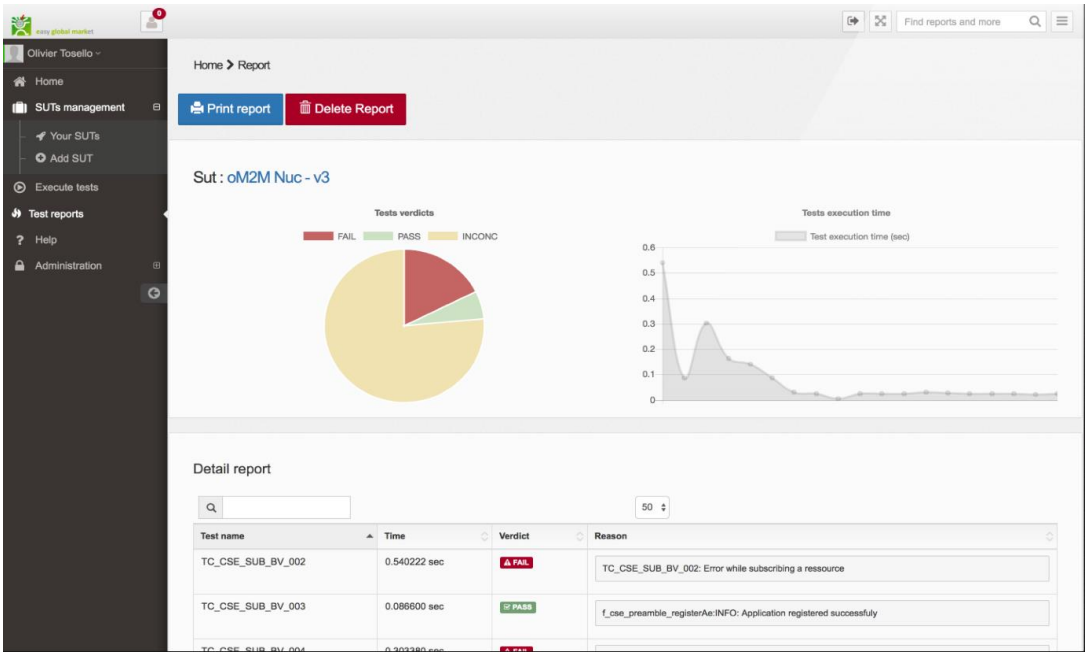

Figure 19. Dashboard presenting the report

The test report is composed by three parts:

- a diagram showing which proportion of test are passing / failing / inconclusive;
- a diagram displaying the time each test-case took, in case a user need more details in order to select test case based on execution time, to exclude the ones taking too long for example;
- lastly, a detailed report for each test case.

Test name	Time	Verdict	Reason
TC_CSE_SUB_BV_002	0.540222 sec	FAIL	TC_CSE_SUB_BV_002: Error while subscribing a ressource
TC_CSE_SUB_BV_003	0.086600 sec	PASS	f_cse_preamble_registerAe:INFO: Application registered successfuly

Figure 20. Details from test report

The detailed report is giving the user four types of information:

- the test-case name, in case we want to check only the failing ones for the next execution. The name of the test-case is used for requirements traceability;
- the time of execution;
- the verdict: pass, fail or inconclusive.
- and, the reason of the failure; written directly in the TTCN-3 code, the corresponding log is extracted and allow the user to quickly know where was the error.

2.4.5 FHNW inputs and outputs

Inputs

Main inputs of the application are as follows:

- Information about the product(s) in question and the company involved entered by the user;
- Tracking information collected over time observing the SMEs involvement into security.

Outputs

Main outputs are:

- Benchmarks about the SME and product maturity level based on CYSFAM;
- Identification of missing gaps and "low hanging fruits" in terms of security. Proposal of measures to improve security based on the CYFAM model;
- Raw data for study;
- Benchmark/Ranking towards other involved SMEs (if data is provided).

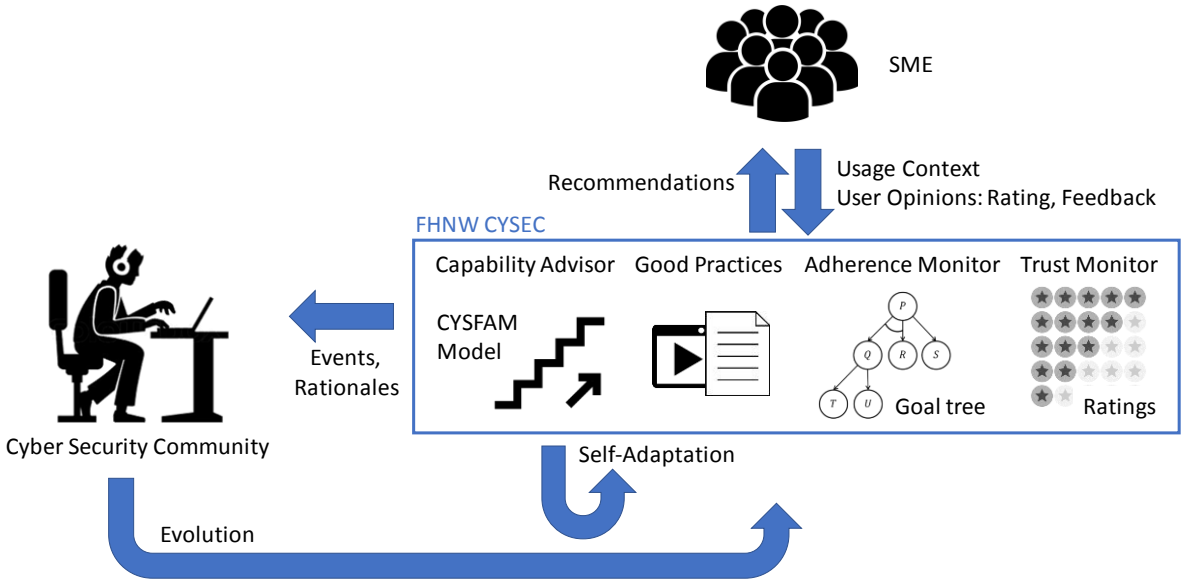

Figure 21. FHNW flow

2.4.6 FORTH inputs and outputs

The control panel of the proposed system is responsible for storing to a local database the users (administrators or simple users) that are authoritative to access and use the honeypot system. The authentication of the users is being performed by the organizations' LDAP service. The attack related information is being kept in PostgreSQL.

The logs produced by the system are also stored is simple text format as well as the configuration files that the system is using for initialization and other operations.

Inputs

The system is configured to monitor a number of unused IP addresses also known as dark space, inside the SME network and receives all the connections and the network traffic destined to that unused

address space. The honeypot sensors then analyse the incoming traffic and log all the activity to a PostgreSQL database.

Outputs

The sensors when a new incident takes place log that information via a REST API to a PostgreSQL database. The information that is logged includes:

- **Connection type**: type of connection occurred.
- **Transport protocol**: transport protocol of the connection ('udp', 'tcp' or 'tls').
- **Connection Timestamp**: date and time when the attack connection occurred.
- **Local host**: IP address of the sensor which captured the specific connection.
- **Local port**: Port number of the local emulated service.
- **Remote host**: IP address of the attacker.
- **Remote Port**: Source port number of the attacker.
- **Connection protocol**: type of service attacked (e.g. smbd).

Logs kept for SSH attack attempts:

- **SSH ATTACKS**
  - **success**: 1 for a successful SSH attack, otherwise 0.
  - **username**: the username used in the attack.
  - **password**: the password used in the attack.
  - **timestamp**: date and time when the attack occurred.
  - **ip:** ip address of the attacker.
  - **url:** url of the downloaded malicious file.
  - **outfile:** local directory of the downloaded file.
  - **input:** commands inserted.

Sensor Management:

The control panel is also the one that can communicate with the honeypot instances. The whole communication process takes place over SSL using certificates. A REST API is used for the two components (the control panel and the honeypots) to interact with each other. Each honeypot instance includes a web service that is responsible to receive and serve the incoming REST requests from the control panel.

The REST API used for the communication of the two components uses the format below:

"https://" [honeypot]:[port] "/honeypot/actions?action=" [function] "&" [query]

Where:

- **honeypot** is the IP address of the honeypot VM;
- **port** is the port where the HTServer is waiting for incoming connections;
- **function** is referring to the desired operation to be performed on the honeypot VM;
- **query** is a list with variables.

Available "functions" currently include the following:

- **aliveAction**: checks whether the honeypot instance is alive and the HTServer is working properly.
- **startSensorAction**: initiates the honeypot services. After that the honeypot is able to monitor and capture potential attacks.
    - Cip: one or more CDIR IP addressed separated by the comma character. Those are the IPs that will be used for monitoring by the respective honeypot instance.
- **stopSensorAction**: terminated all honeypot services. The system will be no longer active.

2.4.7 IBM AngelEye inputs and outputs

The system is composed of two phases. The offline training phase and the online patching phase.

The inputs for the offline phase are:

- A source code or an executable;
- Optional: A test corpus, a dictionary of the file format, known vulnerabilities.

The outputs of this phase are:

- A predictive model;
- An extended test corpus.

Inputs to the online phase:

- A file.

Output:

- A prediction/score of file maliciousness.

2.4.8 IBM Anti-ROP inputs and outputs

Anti-ROP for binary

In this case, the tool receives PE file and output shuffled PE file.

Anti-ROP for source

We have developed a plugin for Clang llvm compiler, thus, the input is C/C++ source code and the output is ELF file.

3 Integration principles for the unified framework

SMESEC is a heterogeneous collection of tools, each one of them contributing to SMEs security in one or several aspects. However, to properly protect an SME, security must be treated as a whole, meaning that all the tools must be integrated into a consistent framework, greater than the sum of its parts.

3.1 Tools classification

The previous section has identified 10 different tools, provided by 7 partners, each contributing to different security areas. The first step in the integration process is to identify the role of each tool. For this purpose, we identified two main categories: real-time tools and offline tools.

Real-time tools are tools that actively block an attacker or alert its presence. They must be continuously running, usually in the background and alerting when an incoming attacked is detected and/or blocked. The logs from these tools can also be used for reporting. Since each tool has access to different types of information, integration is necessary for a unified view.

Offline tools perform security-related tasks that usually run on-demand, for assessing and improving the security of a single file or of the whole system.

Figure 22 presents an overview of the tools offered by each provider, the real-time tools being coloured in red while the offline tools are coloured in cyan.

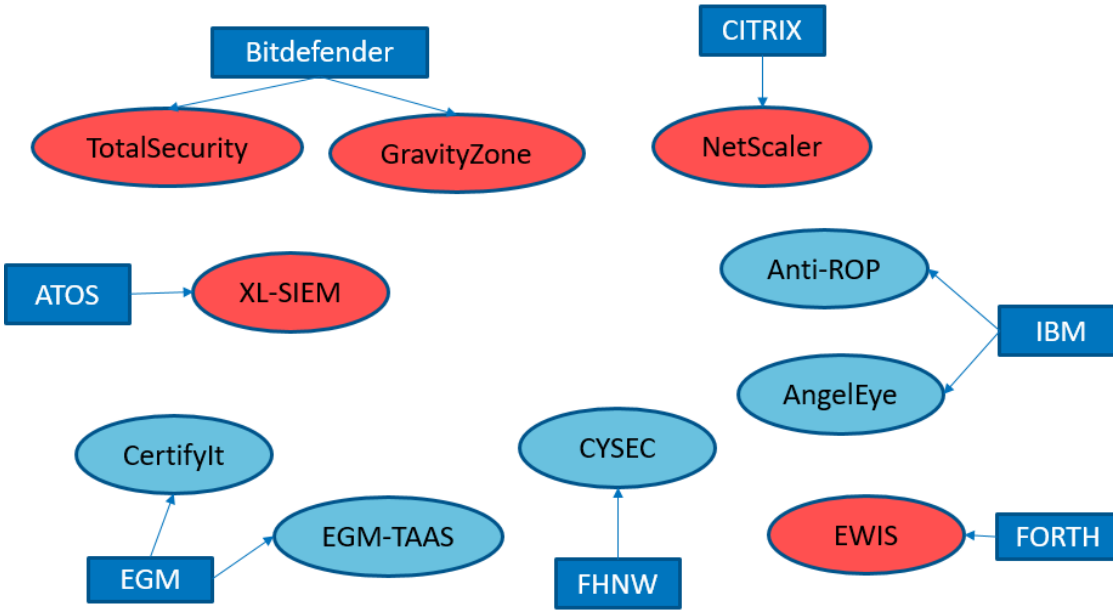

Figure 22. An overview of tools and services providers

3.2 Real-time tools communication model

As stated in the previous subsection, real-time tools are protecting the SME infrastructure by running continuously and monitoring for incoming threats. Figure 23 presents a proposed communication architecture for the online tools.

Bitdefender GravityZone consists of Endpoint Security, that can be deployed on servers, workstations and mobile devices. This component ensures anti-malware protection and monitors systems and network activity. Endpoint Security will report to Bitdefender Control Centre, using internal communication protocols that are already implemented.

Citrix NetScaler has 3 components: the App Firewall, that filters incoming Internet traffic for web applications (and not only), the Unified Gateway, used by mobile devices and the Secure Web Gateway that filters outgoing Internet traffic.

Forth EWIS consists of honeypots that can be deployed in the SME network and they will detect intrusions in the early phase.

The Atos XL-SIEM is a security information management tool that receives real-time events from various sensors, including other tools from the consortium and offers a real-time unified view.

As Figure 23 shows, GravityZone Control Centre, NetScaler and EWIS will provide events to the XL-SIEM. By studying the common inputs and outputs for the aforementioned tools, we have learnt that XL-SIEM can receive events using the syslog protocol, which is also supported for sending events by the events providers. NetScaler emits some events using syslog, while the security-related information is outputted using AppFlow and the Nitro API.

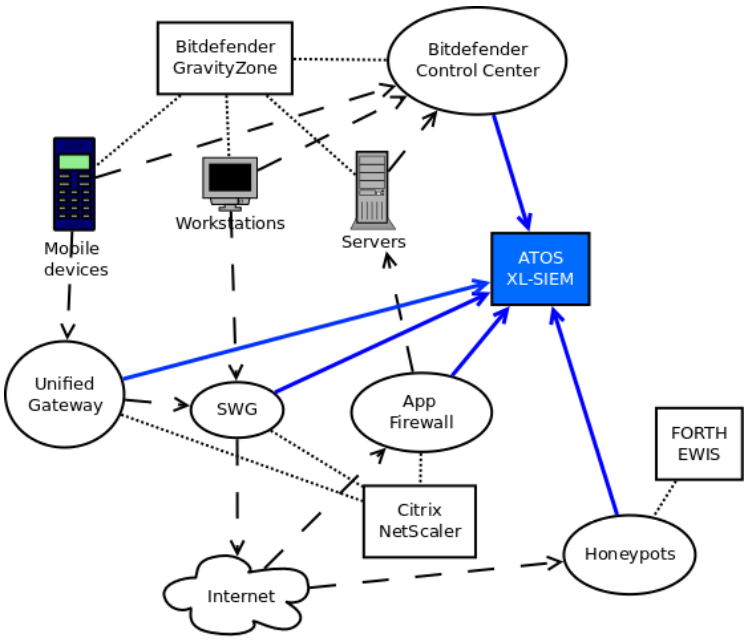

Figure 23. Tools real-time communication model

3.3 Offline tools communication model

The offline tools include all the consortium tools that are not real-time and usually run on demand.

The IBM tools, Anti-ROP and AngelEye are working at files level, by protecting binaries from ROP exploits and offering virtual patches.

EGM tool, Test-as-a-Service provides support for software testing, a crucial step for defeating software vulnerabilities and preventing exploits. Test-as-a-Service provides a generic framework for integrating different tests in a single dashboard.

CySec from FHNW assesses the security for the whole system, based on the information received.

Figure 24 proposes a communication model, centred around TAAS, that can run tests on demand and centralize their results.

The CySec tool, can use tests information from TAAS, and also integrate information from XL-SIEM, as real-time security events can also contribute to the whole system security assessment.

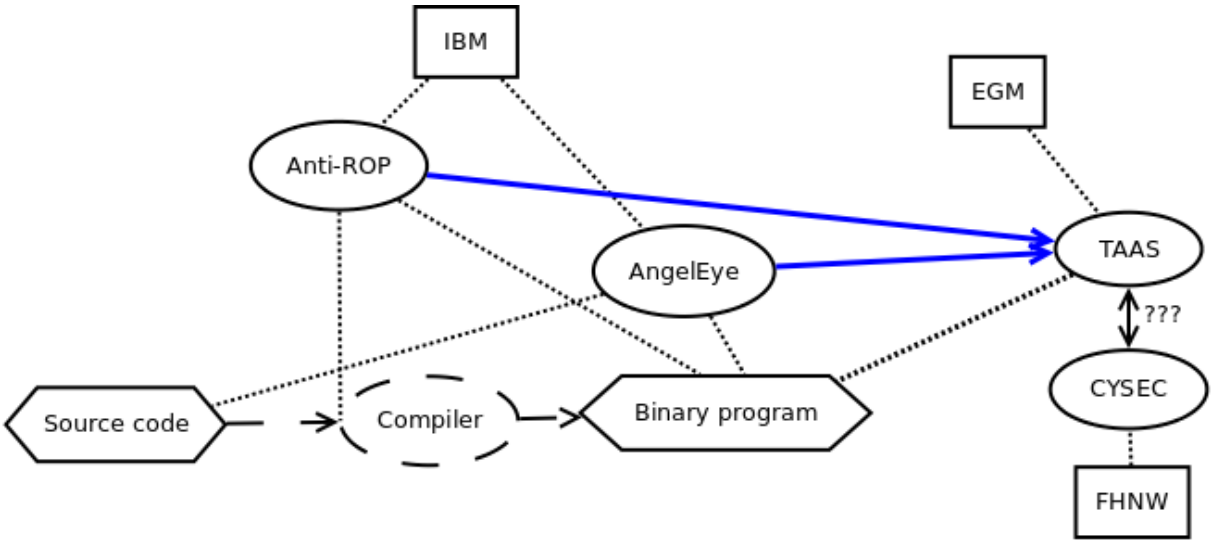

Figure 24. Offline communication model

3.4 Tools automation

Security is the primary goal of SMESEC, but SMEs that consider this framework will also take usability into account. Recurring tasks for these tools should be automated, in order to require minimum human intervention. We are particularly focusing on the following three principles:

- Security tools should mostly work in the background. Users should be aware of them only when they are needed.
- The tools should be easy to deploy. Most of the deployment steps should be done automatically.
- The tools should be easy to update, preferably automatically. If the tools do not update automatically with the latest patches, they become vulnerable and affect the system security.

The following sections will describe each of these principles in details.

3.4.1 Tools that work in the background

The analysis on the SMESEC tools showed that the real-time tools are designed to work in the background, while the offline tools require user interaction.

Atos XL-SIEM is designed to run in the background to detect alarms based on the incoming events.

Bitdefender GravityZone works in background, alerting the users only when a threat is present.

Citrix NetScaler as an appliance, is in the middle between two endpoints, so in this sense, it runs in the background. NetScaler itself is a hardware or software appliance which runs in the "foreground" on the hardware or the hypervisor. Since we care mostly for the software version of NetScaler within SMESEC, it runs on a dedicated VM connecting to several subnets. Conclusively, it runs in the background from usage perspective, but technically it needs its own VM.

FORTH's honeypots operate in the background collecting alerts and storing them in the appropriate databases. The visualisation part of the architecture works in the foreground.

EGM TaaS does not do any background tasks. It's a running server (based on Docker) requiring user interaction in order to perform an action (test execution) manually (or scheduled if necessary).

The users will actively interact with the CySec tool. For that reason, it is not working in the background.

The IBM tools do not work in the background.

3.4.2 Principles for automatic deployment

Atos XL-SIEM does not provide automatic / silent deployment.

The GravityZone Control Centre must be deployed as a virtual appliance. From there, Endpoint Security can be deployed automatically on workstations.

NetScaler VPX (the software appliance) comes as an image for all major hypervisors (XenServer, VMware, KVM, Hyper-V) and as a cloud image for AWS and Azure. In general, the setup process on a hypervisor is straightforward and does not need any technical knowledge. As such, it can be deployed automatically as any other VM image on hypervisors or clouds.

The Core of the FORTH Honeypots is a Virtual Machine Image which can be preconfigured, so it is automated in that sense. The visualisation part is up and running but if we need one per CI then it needs to be reinstalled which means it's not automated.

Test as A service platform (EGM-TAAS) is available at two levels:

- The first one is online, as a web service. A client can connect to the services and execute some tests. In this case, if a client wants to install the web service, he must have a Linux based machine with Docker. After that, it is merely a script that will automatically download the required images and launch them. The Docker images are on a cloud, using EGM credentials to access them. It can be manually deployed on other servers.
- The second, in case of private networks, is available as a hardware. EGM will purchase and install the Test as a Service platform on a device which will be sent to the client. This hardware component will be used as an internal server and allow all the private network to use the web service internally (in case of confidentiality / privacy / certification issues). In this case, at EGM a manually process will be implemented while setting-up a dedicated hardware

and installing the Docker images directly on the hardware instead of querying them from an online cloud.

Within the SMESEC project, the first case will apply, where we deploy the TaaS with Docker directly by requiring an update from the cloud where the images are stored. A web interface that will be installed with a script, where the test execution is "Dockerised" (and the images are automatically download and used from the web service).

CySec is a web platform. One may deploy an own instance on premises or in a cloud (VM), but this is not necessary.

IBM tools can be deployed automatically using a VM or Docker image.

3.4.3 Principles for automatic update

The update process is an important part for any framework, especially for a security framework. There are two solutions to automate it for each component. The first one involves the component handling its own update, meaning that it is able to retrieve patches from the update server, apply them then restart, while maintaining the active protection. The second solution involves a 3rd party update tool that will download the patches, turn off the component, patch it, then restart it. This solution introduces a downtime for the component, where the protection is temporarily turned off. We must ensure that each component, either falls into the first category (self-updating), either it is safe to turn off, patch and restart.

The update feature is not included in the Atos XL-SIEM. But it can be shutdown, patched and restarted it.

Bitdefender products have the auto-update feature, the only requirement being Internet connectivity. Particularly, virus definitions get updated every few hours or even hourly.

Updates for the NetScaler VPX come in the form of a downloadable image, which can be deployed through the GUI or command-line tools. Minor version updates are not so frequent (typically 1-2 months) and contain bug fixes or optimizations but also new features. Typically, it is desired to install these updates, but not critical. To deploy the update, the appliance needs to restart. There are though high-availability setups to prevent service disruption.

Forth's Honeypots do not auto-update but can easily be turned off, patched and restarted.

EGM TaaS has two parts, for the private on hardware TaaS, it requires updates from the public one. The action will have to be ordered by the user as a maintenance task. In SMESEC case, we are more on a public cloud situation, where we will update the tests directly into the servers. The same goes for the interface in case of new functionalities. In any case, since EGM TaaS is a testing tool, it does not endanger the system if we have to turn it off.

CySec is a web platform. The Platform itself receives automatic updates.

IBM tools do not acquire automatic updates.

3.4.4 Operations that need to be made user-friendly

SMESEC framework deployment on an SME environment should not require a high technical background. For this reason, any operations related to installation, updating, maintenance or interpreting the result should be made in a user-friendly manner.

The installation stage requires each of the consortium tools to be installed separately. Some of the tools must be deployed on workstations, while other tools work as virtual appliances. All of these tools should be easy to deploy and the standard configuration must cover most cases encountered in the SME security requirements.

The update process should be performed automatically in most cases, without user interference or assistance. Missing security updates can introduce further security issues and we want to avoid this possibility. When update is not performed automatically, the update instructions must be clear enough to be executed even by non-technical users.

The maintenance operations should occur as seldom as possible and with minimal human intervention.

In case a threat is detected, the information provided to the user should be precise and helpful for taking the most appropriate action. This information should include at least the type of threat, the risk level, the affected systems and the mitigation options.

3.4.5 Tools integration timeline

Sections 3.2 and 3.3 discussed the communication model for the real-time and for the offline tools, emphasizing the point where each tool must be deployed and what type of data must be sent.

For a functional prototype, all these integrations must be handled by both parties. For instance, if tool A has to send data to tool B, the following steps must occur:

- Provider for tool A implements sending the data.
- Provider for tool B implements receiving the data.
- Provider B gets access to an instance of tool A for testing the input data.
- Provider A examines interface / dashboard / logs from tool B in order to make sure that the sent data is correctly interpreted by tool B.
- If provider A sends data to more tools, regression tests must occur in order to ensure the data is still sent correctly.
- If provider B receives data from more tools, regression tests must occur in order to ensure the data is still received correctly.

A final integration test will also be performed after each pair of tools are integrated, to ensure that all the components work together.

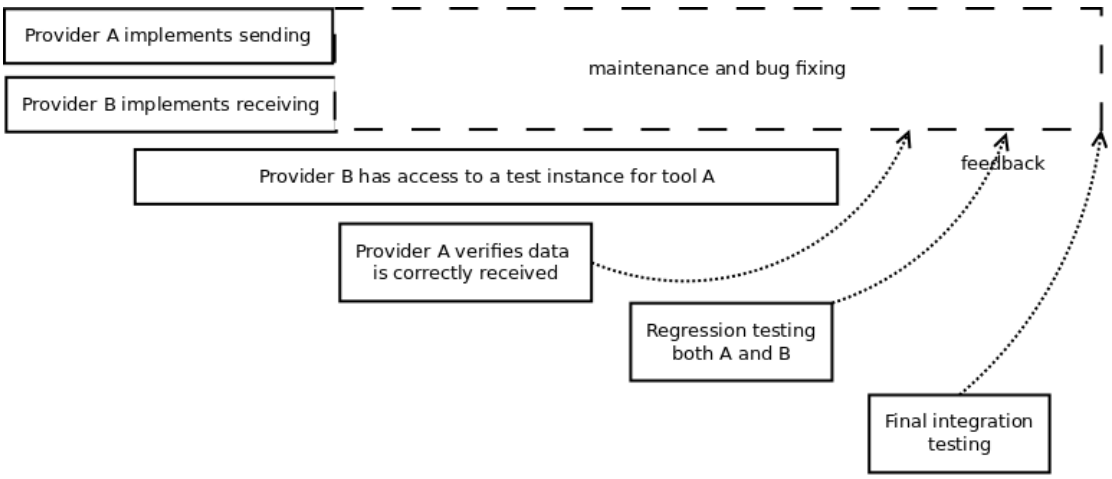

Figure 25. Timeline for tools integration

After deciding what data needs to be sent, both tool providers can implement the communication protocol. Usually tools that send data already implement this feature, while tools that receive the data need to implement the capabilities to integrate it.

At some point, the tools providers need to set-up a common test platform. This platform needs to be available until the final integration testing. During the development phase, it should support unit testing. After the development phase, the integration should be tested by both tool providers. Particularly, provider A should check that the data from his tool is correctly received and interpreted.

Tool A may send data to several other tools, while tool B receives data from several other tools. After each integration, the codebase changes so regression testing should be performed in order to verify that the original integration is not broken.

Finally, after all tools integration is finalized, a final integration test will be performed. This test is more complex since it checks the correct integration for each pair of tools, simultaneously.

Although the development phase will end earlier, the development teams should be available during the entire period for maintenance and bug fixing.

4 Basic principles for innovation

The products integration follows the principle that the whole is greater than the sum of the parts. The SMESEC framework will be innovative by combining multiple security products, offering unified protection rather than just a suite of tools.

The real-time tools will innovate by providing events for the SIEM component. A security threat might not be detected by any tool individually, but several tools may output useful information that will be correlated by XL-SIEM. The first innovation in this scenario is the implementation of the communication protocol. Since the syslog protocol is open and widely used, the tools will be loosely coupled and will be able to work in different context. The second innovation is the ability to generate security-related events, even if they don't necessarily mean detecting a threat. The tool providers will need to think on a bigger picture, where partial clues about possible threats might be dwelled into by other tools, in an integrated environment. From the SIEM point of view, the innovation means handling large loads of events from heterogeneous sources. Event correlation algorithms and heuristics will also be developed.

The offline tools will also innovate through integration. By implementing common interfaces, they will generalize better and will provide the ability to integrate with other tools in the future. The integration will also provide access to a wider range of targets to be tested by the test tools. The security assessment tools will also receive more data, from which they can learn more elaborate models.

5 Conclusions

D2.2 – "*SMESEC security products unification report*" presents the results of the analysis on the following topics:

- **Technical features and characteristics of each and every solution** that will be part of the unified SMESEC framework;
- **The basic principles for a common integration**;
- Elements of **enhanced innovation**.

At 1st, the document collects (as part of Task 2.2) relevant and essential information about each and every SMESEC solution on a fourfold dimension: (1) solutions' overview to understand their role within the SME protection, (2) description of technical characteristics to understand their complementarity, (3) detailed description of each solution's architecture to learn about integration options and (4) overview of data flow in a form of input / output. The feedback collected is useful for sketching a very high-level integration and understanding how both the individual products and their integration under a unified SMESEC framework matches the security of the pilot use cases.

The second key-output of this deliverable is covering basic principles for designing the common architecture following the next aspects: backend mechanism, deployment and update implementations and an integration timeline.

Thirdly, the focus is on innovation enhancements and their role in matching the SMEs requirements. Also, the innovation progress is expected to develop key-differentiators of SMESEC unified framework which will bring competitive advantage within the market of cybersecurity solutions for SMEs.

The information has been collected from SMESEC partners using targeted questionnaires. The results have been further analysed in the other two deliverables of WP2 and specifically:

- D2.1 explains in more detail the requirements of the 4 SME pilots.
- D2.3, among others, contains a detailed explanation of the risk assessment process that has been followed for the analysis of the four pilot use cases.

The key-remarks of this deliverable are:

- The solutions providers are key-actors of the market of cybersecurity solutions with proven and successful track-records.
- The SMESEC partners provide key-security solutions covering different requirements which proves the complementarity among solutions and contributes to a consistent unified framework.
- The technical analysis illustrates that there are similarities among the solutions which facilitate the common integration.
- A unified SMESEC framework will provide added-value to all individual solution, multiplying the benefits for SMEs.

References

[1] AlienVault OSSIM. Link: https://www.alienvault.com/products/ossim. Last time visited: November 21st 2017

[2] EsperTech. Link: www.espertech.com. Last time visited: November 21st 2017

[3] Chunhui Li, Robert Berry "CEPBen: A Benchmark for Complex Event Procesing Systems". Technology Conference on Performance Evaluation and Benchmarking 2013. pp 125-142

[4] Apache Storm. Link: http://storm.apache.org/. Last time visited: November 21st 2017

[5] STIX. Link: https://oasis-open.github.io/cti-documentation/. Last time visited: November 21st 2017

[6] JSON. Link: https://www.json.org/. Last time visited: November 21st 2017