Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

SMESEC

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

D4.9 Overall Pilot alignment and integration process report

Document Identification			
Status	Final	**Due Date**	31/05/2019
Version	1.0	**Submission Date**	24/06/2019

Related WP	WP4	**Document Reference**	-
Related Deliverable(s)	D3.3, D4.2, D4.4, D4.6, D4.8	**Dissemination Level (*)**	PU
Lead Organization	ATOS	**Lead Author**	Jose Francisco Ruiz
Contributors	WoS, Scytl, UoP, GridPocket, FORTH, CITRIX, BD, EGM, FHNW	**Reviewers**	Francisco Hernandez (WoS) Christos Tselios (CITRIX)

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
Francisco Hernandez, Olmo Rayón	WorldSensing
Jordi Cucurull, Noemí Folch	Scytl
Michal Burdzy	GridPocket
Kostas Lampropoulos	UoP
Ciprian Oprisa, Ovidiu Mihaila	BitDefender
Manos Athanatos	FORTH
Hamza Baqa	EGM
Christos Tselios	CITRIX
Jose Francisco Ruiz, Pablo Barrientos	ATOS

Document History			
Version	Date	Change editors	Changes
0.1	31/05/2019	Jose Francisco Ruiz (Atos)	Creation of template, sections and executive summary
0.2	10/06/2019	Use case partners Jose Francisco Ruiz (Atos)	Contribution to Section 3 Section 2
0.3	15/06/2019	Tool owners	Contribution to Section 4
0.4	20/06/2019	Jose Francisco Ruiz (Atos)	Review of the document, minor modifications and contribution to Section 5
1.0	24/06/2019	ATOS	Quality Review + submission to EC

Quality Control		
Role	Who (Partner short name)	Approval Date
Deliverable leader	Jose Francisco Ruiz (ATOS)	24/06/2019
Technical manager	Christos Tselios (CITRIX)	24/06/2019
Quality manager	Rosana Valle (ATOS)	24/06/2019
Project Manager	Jose Francisco Ruiz (ATOS)	24/06/2019

Table of Contents

List of Figures

List of Acronyms

Abbreviation / acronym	Description
AI	Artificial Intelligence
API	Advanced Persistent Threats
AWS	Amazon Web Service
BEST	BitDefender Endpoint Security Tools
CS	Content Switching
DMZ	Demilitarized Zone
DDoS	Distributed Denial of Service
EWIS	Early Warning Instruction System
HTTP	Hypertext Transfer Protocol
HVI	Hipervisor memory Introspection
IDS	Instruction Detection System
KVM	Kernel-based Virtual Module
LB	Load Balancing
MBT	Model based testing
ML	Machine Learning
NAM	National Association of Manufacturers
OVA	Open Virtual Application
QA	Quality Assurance
RAE	Risk Assessment Engine
SSH	Secure Socket Shell
SUT	System Under Test
TaaS	Test-as-a-Service
UI	User Interface
VIP	Virtual IP

Executive Summary

This document provides information about the processes and updates done in the SMESEC Framework and tools for their integration in the use cases.

As the tools were very technical and SMEs had to integrate them as new we went through a series of iterations that helped us all (both tool owners and use case partners) to refine the processes and technical assets of the project. We had three iterations: the first finished at M12, the second at M18 and the final at M24. Each iteration had different objectives and results and played the role of baseline for the following one.

Regarding the tools, it was necessary to have several different levels of adaptation. Some tools only had to undergo minor modifications mostly on the client side in order to adapt to the new target system (e.g. technical capacity), while others needed additional improvements and modifications for updating the integrated communication model or for validating their functionality through a specific acceptance testing process according to the technologies of the pilots, etc.

For the use cases, the work of integrating the SMESEC Framework required specific work, analysis of their system and how they wanted the integration to work and adaption of their systems for using the technologies of the tool. We had several meetings and workshops between all the use cases and tool owners in order to have all the possible feedback and align the needs of each of them.

This deliverable presents first the process of integration of SMESEC in the use case partners, covering the different stages and goals of each iteration. Following we show a description of the adaptation of the pilots and experience of the use case partners and finally the results and comments of the refinement and updates of the tools of SMESEC to the pilots.

1 Introduction

1.1 Purpose of the document

This deliverable describes the work done in WP4 for aligning and refining the SMESEC Framework and tools to the pilots. It is composed of three different elements:

- the process we followed for the integration of the tools, internal components (e.g. authentication), etc. in the use cases.
- the adaptation done in the use cases for integrating SMESEC.
- the updates, refinement and changes done in the tools for facilitating the integration in the use cases.

Finally, we present the conclusions of these activities and the future work for the next year of the project.

1.2 Relation to other project work

The activities presented in this deliverable are complementary to the work done in WP3 (updating, adaption and refinement of the SMESEC Framework) and described in other WP4 deliverables (integration of SMESEC in the pilots). Also, the output of this work will be used in WP5 for the validation of the use cases.

1.3 Structure of the document

This document is structured in five major chapters

Chapter 1, this one, presents the introduction, objectives and structure of the deliverable.

Chapter 2 presents the process we followed for the integration of SMESEC in the pilots and description of each iteration.

Chapter 3 describes the updates and changes the use cases performed in their pilot for the integration of SMESEC.

Chapter 4 presents the adaptation and updates done in the tools using the feedback of the pilots for facilitating the integration in the use cases.

Chapter 5 shows the conclusions of these activities and the basis for the final refinement of the SMESEC Framework and the validation.

1.4 Glossary adopted in this document

- **SMESEC Framework.** Unified framework providing cybersecurity solutions, dashboard with the data of the status of the system and internal components (e.g. authentication).

2 Process for adapting the SMESEC Framework to the pilots

This section presents the process we followed for adapting the SMESEC Framework to the use cases. As the project is a three-years-one we planned to do different iterations with specific objectives in each of them. This way, we planned to do an integration of tools as is shown in Figure 1.

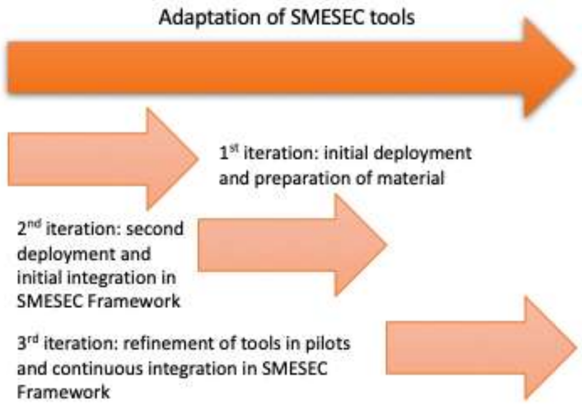

Adaptation of SMESEC tools
1st iteration: initial deployment and preparation of material
2nd iteration: second deployment and initial integration in SMESEC Framework
3rd iteration: refinement of tools in pilots and continuous integration in SMESEC Framework

Figure 1. Iterations of integration of SMESEC tools

So, we have been working this and the previous year in three iterations with the final goal of having all tools and internal components of SMESEC integrated into the pilots according to the needs and specification of each use case. As the internal components were going to be developed after having an initial version of the architecture we worked in parallel with the initial integrations of the tools. Finally, we focused this year in the initial part in the integration of the authentication component and the final refinement and deployment/configuration of the clients or agents of the tools. More specifically, following we describe each of the three iterations we did.

1st iteration

The first iteration corresponded to the first year of the project. After finalizing (i) the requirement elicitation and analysis, (ii) the initial architecture design and (iii) the compilation of all provided tools, we did an exercise of selecting which tools would be implemented in each of the pilots. This was done by discussing with each pilot about the needs they have and what they would like to accomplish at the end of the project, bearing in mind not only the technical requirements but also the specific business perspective of each company participating in the pilot. Once we identified the tools which were necessary to be integrated we specified the actual tool integration order, bearing in mind the initial estimation of updates and refinement necessary in each case. This information was obtained by having specific calls of each tool provider with the use case partners. These meetings helped the tool owners to understand the technologies use cases were using and how much effort would take to adapt the tools to them.

After having an idea of necessary adaptations, the tool owners started working in them. In parallel, other tools that were identified as easier for integrating were selected as the starting point. Once the use case partners were integrating, configuring and understanding the tools they were providing feedback for improving the integration and improve the robustness and user experience.

In parallel to this process we had an initial version of the architecture and started working in the internal components of the SMESEC Framework. The initial implementation was ready by M12 and we performed testing of the integration in order to check everything was working as intended.

2nd iteration

The 2nd iteration was planned from M12 to M18. The main goal was to continue refining the tools already integrated, start integration of the ones planned and for the tool owners to start integrating the common components of the SMESEC Framework (e.g. authentication, dashboard, etc.).

The refinement of the existing tools was done for having a better understanding of how they work and the information they provided. This way, in order to extend some functionalities and provide better use the tool owner had to refine the tools according to the comments of the use case partners.

In addition, use case partners started integrating the next planned set of tools in their systems. This process had an easier start because all the available feedback from the first iteration together with a common process we identified for this work was taken into consideration. The process consisted in having specific workshops of the tool owners with the use cases of the project, explaining the process of deployment and configuration and showing them demos, videos and material, they could use for the integration. This was a very interesting and important action as we learned that the material the tool owners had originally was good for cybersecurity experts but not for the type of users of SMEs. Therefore, tool owners had many feedback about how to provide good and correct information to the use case partners and they would also give even more feedback as they used this material. The material, after several iterations, is now in the training platform and include documentation, examples, videos, etc.

After the workshops the tool owners provided the information to the use case partners and they worked in the integration of the clients or agents in their system. In order to support this work, we had scheduled monthly calls for talking about general issues, so all use case partners could benefit from it and also tool-specific ones, as a webinar, where we supported use case partners in the correct configuration, access to the information, understanding of the information, etc.

Additionally, to this work where the tool owners where updating and refining their tools to help the integration by the use case partners, we were working in internal components of the SMESEC Framework that had to be integrated in the tools. Among others, and as described in D3.3, the more important elements were the authentication component and the integration of the interfaces in the dashboard. Atos provided the infrastructure and technical support for this and the tool owners integrated their systems. The main objective of this work was:

- to have a common authentication process so all tools where working under the same general policy for log-in;
- to have a unified dashboard where all the tools provide their information, so the SMEs can use it as single-entry point.

The result at the end of the iteration was very satisfying as the tool owners had a lot of feedback about their tools at many different stages: deployment, configuration, usability, etc. Also, we had all the tools integrated with the authentication system of SMESEC (Keycloak) and started working in the unified framework for all the tools.

3rd iteration

The last iteration was from M18 to M24. This iteration focused in the refinement of the tools from the configuration phase (adaptation of the tools to the technical requirements of the use cases) and work in a more in-depth integration of the tools with the SMESEC Framework.

Regarding the first part tool owners had several meetings with use case partners for providing information of configuration and usage of the tools. For instance, Citrix ADC went through a series of adaptations and internal component reconfigurations for having better performance and improved usability when operating in the Scytl Pilot. This process provided CITRIX invaluable feedback for improving the overall installation and configuration material (for example we were able to deploy the client in our premises in 15 minutes following the videos and explanations provided, which was a huge improvement from the first iteration). Other tools focused in providing material for the pilots such as the TaaS tool, which had different workshops with the use case partners for providing tests according to the specific needs of each use case partner. Other tools worked in improving the information provided to the use cases as the existing one was not as clear or useful as they wanted.

The feedback was compiled by each tool owner and prepared for when the open call starts, as it will be the best way for evaluating how user-friendly and useful all the material prepared is and how fast can the tools adapt to the different technical and business needs of several different SMEs.

Additionally, to this work the tool owners worked also in integrating more at low level their tools with SMESEC. As we commented in the second iteration that they started working in integrating their dashboards in a unified framework, we also started working in providing data directly to the SMESEC Hub, which will compile all this information in order to make it available for extra functionalities not possible for each individual tool. Therefore, tool owners worked in providing APIs for accessing information of their tools that could be useful from a high-level point of view.

Currently we are working in the minor refinement of the tools in the use cases and the adaption to the SMESEC Framework. We plan to have more iterations of the tools (including the material for installation and configuration) thanks to the open call of the project (where we have different categories with specific objectives).

3 Refinement and update of use cases

This section describes the refinement and update of the use cases for the integration of the SMESEC Framework (and tools) in the three iterations described previously. The objective of this section is to provide an idea of how the work for integrating SMESEC was and the needed adaptations.

3.1 Scytl

The integration of SMESEC Framework for the e-Voting use case took several steps which are described in the following lines:

- Deployment of e-Voting system in AWS: The first step to integrate the eVoting system with the SMESEC framework was the deployment of the voting system itself within the AWS infrastructure. Thus, we initially worked in the adjustment of the deployment scripts (with Ansible) to do this task.
- Deployment of Citrix ADC VPX in AWS: After the eVoting System was deployed in AWS, we had to deploy the Citrix ADC VPX solution. This tool was directly installed using the AMI provided by CITRIX in the AWS MarketPlace following their instructions. Later, they install the appropriate license and configured the service. Instructions for integrating the tool are currently included in the Appendix of D3.4. Due to minor delays in the work, the configuration of the rules to be applied by Citrix ADC VPX was implemented later.
- Configuration of XL-SIEM to report on the activity of Apache, Tomcat and Internal EWIS Honeypot servers: we deployed a XL-SIEM agent within an EC2 instance, as well as the other components configured to send their syslog-based logs to the agent. The XL-SIEM agent was also configured with the appropriate plugins for each type of log to be supported with the help of Atos. We also had to write a custom plugin for the Tomcat logs and modify the one for the Apache Web Server.
- Configuration of the Honeypot at the secure zone: The Honeypot was deployed and configured in the secure zone as a standalone EC2 instance. An OVA image was provided by FORTH. We uploaded it in AWS and deployed it as an instance. This instance is devoted to detect attacks in the secure zone area. The configuration of the HoneyPot was directly done by FORTH by providing them access to the instance via SSH.
- Detailed and optimized internal configuration of a standalone Citrix ADC VPX instance in AWS cloud: all the traffic between the voter's device and the voting server was redirected to pass through this component, which inspects the data ensuring it is correct. We did this by modifying the DNS entry that was referring to the eVoting system. Also, the Citrix ADC configuration was updated by Citrix in order to behave as a client of the eVoting system when requests to the eVoting system were received by it.
- Definition and setup of application level firewall rules and policy files for traffic categorization in the aforementioned Citrix ADC VPX instance: Citrix ADC allows the inspection of the traffic and rejection of it in case it does not comply with the rules defined. After redirecting the traffic to Citrix ADC, the rules were configured from Scytl with the help of Citrix. The API of the eVoting system was considered in order to generate rules based on regular expressions that match the expected HTTP POST requests issued by the Voting Clients. These requests are the

most important because they are directed to the Web Server and, from there, to the Tomcat (which is in the Secure Zone). The rest of the HTTP requests were allowed because they are processed by the WebServer that is in the DMZ sub-network.

- Development of the scripts to generate the samples to train Angeleye: We wrote the required scripts to create many requests for the different voting endpoints and evaluating the response of the voting server, categorizing each request as good or bad depending on the output.
- Generation of most of the dataset of samples to train AngelEye: Using the mentioned scripts we generated millions of samples for training and testing the tool.
- Preliminary setup of AngelEye evaluation client in Apache webserver: an AngelEye tool was adapted by IBM to be deployed in the host of the web server to be run periodically and detect possible attacks from the HTTP POST Requests logged by the server. We deployed and tested the instance with several sample requests in order to ensure that it was working properly.
- Evaluation of the CYSEC tool, giving feedback in terms of usability, functionality and overall user experience: the cloud version of the tool was used to evaluate the level of security of the company and to receive feedback about it.

3.2 WorldSensing

The underlying concept of this pilot, “Industrial Services”, relies on the effective integration of the SMESEC framework within the architecture of a commercial IIoT solution (Loadsensing). As explained in the deliverables D4.5 and D4.6, merging cybersecurity with the IoT domain has been vaguely explored in the past, and it still remains as an open issue to be solved before successfully meeting the requirements of some important market niches, such as critical infrastructure monitoring.

Thus, and to be able to address the expected challenges linked to the effective SMESEC adoption with adequate guarantees, Worldsensing undertook the project implementation from the very beginning following a conservative strategy: the incorporation of a limited number of security tools per technology layer of the product architecture, aiming to minimize undesired interconnection problems among them (D4.6).

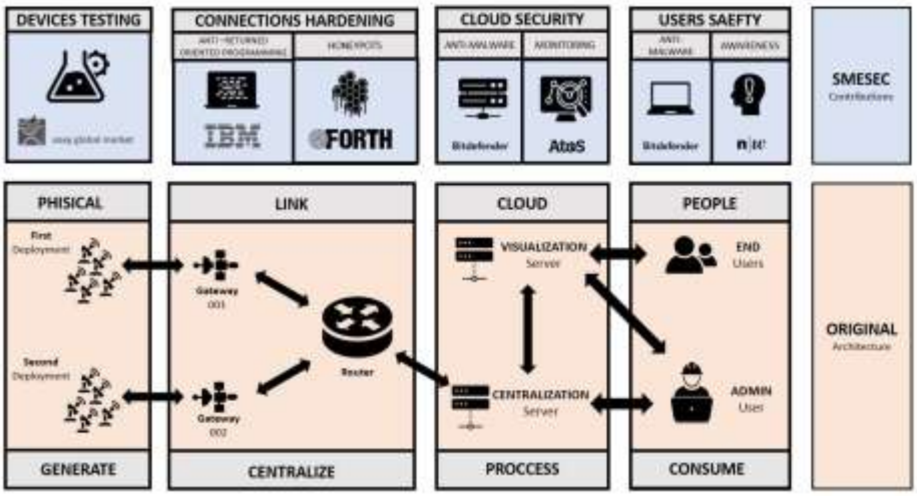


Figure 2. Original Loadsensing architecture overview with the SMESEC

With this approach in mind, the pilot has represented a significant step forward on the security level of Worldsensing products. It should be pointed out that cybersecurity aspects had not been taken into account before, and therefore, SMESEC has demanded a change of mentality at a corporate level: promoting rapid cultural changes like this always entails a conflict, which has been progressively mitigated and addressed during the second year of the project. In any case, a significant part of the internal work directly linked to the pilot has been devoted to modify path dependent internal procedures of the Worldsensing development team to add security elements in the new releases of Loadsensing technology.

Having said that and moving to a more technical description of the work done so far, it can be noted that the adoption of the different SMESEC security tools has not been straightforward as initially expected. The internal lack of knowledge existing at Worldsensing and the poor integration degree of some of the solutions in the early stages of the the project, when the SMESEC framework was just a prototype, have slowed down some parts of the technical work hindering the full pilot up and running before month 24.

Also, the antivirus regular operation and interconnection with the XL-SIEM has been quite demanding in terms of personnel effort with multiple and sequential corrective actions after the first deployment not completely followable from an SME perspective. This has been even worse for some other tools like Citrix ADC VPX provided by CITRIX, which from our point of view remains as a complex solution that cannot be transposed to standard SMEs without the adequate external technical support.

All in all, the implementation has made necessary the active involvement of the technical team of the company in a coordinated way with the internal cybersecurity manager to secure Loadsensing through the SMESEC framework. This has been achieved thanks to the unquestionable interest of Worldsensing to merge IoT capabilities and the security domain. However, and based on our experience, the SMESEC solution should ease the installation process to increase the attractiveness and facilitate its adoption among the SMEs collective with limited cybersecurity knowledge.

3.3 University of Patras

University of Patras' participation in SMESEC project was excellent opportunity to fortify not only its sense.city service but also the overall private cloud infrastructure (UOP private cloud). From the beginning of the project, it was quickly identified that the security status of the sense.city platform was at critical level. The experts of SMESEC consortium performed security assessments revealing to the UOP team a large number of open issues, vulnerabilities and lack of security processes of this pilot. Furthermore, when completing the CYSFAM model that was distributed across all project's pilots, the sense.city platform achieved the lowest score.

The first months in the project were a "wake-up call" for the UOP team, which until then had completely focused on the development of the sense.city service and did not pay and kind of attention to security. The worst aspect about this situation was the fact that the team had implemented some sporadic security tools and rules for its infrastructure and services, maintaining a completely false impression of their security levels. With SMESEC UOP team not only realized that their security was at a critical low, but also that their approach was wrong, since it was not based on any kind of security plan. Furthermore, non-technical operations like emergency response plans, response teams, training etc. were completely absent.

On the other hand, the sense.city service has been created by the NAM Group of the Electrical Engineering Department of University of Patras. The service is not a product of a company and part of its infrastructure is provided by the University, which has its own security rules and procedures. Thus, the security planning for this pilot had to be adjusted to the very specific peculiarities of a free-as-it-is service that is not an actual market product. For example, it was not possible to have a 24/7 CERT, and we could only apply security tools and components to the private cloud but not to the university's network. With these requirements in mind, the security planning of the sense.city's platform was created based on its key objectives. These objectives were the following.

a) Protection against untargeted attacks: Not being a market product of a company with sufficient budget to invest on cybersecurity, sense.city is not capable of investing on something more than protection against untargeted attacks. Also, we believe that the chance of someone explicitly targeting the sense.city is highly unlikely. Achievements by M24: Sense.city hasn't had a major cyber incident in the two years that is participating in SMESEC and it is already extending the use of some of SMESEC tools to its operational ones hosting actual services and platforms. This is a proof that the technical administrators of sense.city service trust the SMESEC solutions to support its security against untargeted attacks.

b) Increase awareness: Within SMESEC project the UOP team put cybersecurity as one of their highest priorities and has started working to apply the necessary security solutions and also train/educate all people involved in sense.city service (developers, IT, public servants etc.). Achievements by M24: Sense.city developers have implemented more testing and QA (quality assurance) phases in their work. Also, the team responsible for the infrastructure management now performs more often security and vulnerability checks and is trying to enforce various security practices like plans for updating and patching, response and mitigation actions etc. Furthermore, the creation of training courses as well as the transformation of the CYSEC solution to a more user-friendly tool will allow the UOP team to not only promote cybersecurity awareness inside its own institute but also to external collaborative parties like the municipalities and their employees.

c) Create a market product: One of the business priorities of the NAM Group is to turn sense.city into a market product. SMESEC has been a great opportunity for the UOP team to work with security professionals to increase sense.city's security levels and create a high-quality product that can reach the market. Achievements by M24: By M24 new features have been added to sense.city platform for supporting people with special needs. These features have increased the value of the service and also draw the attention of other public services (public protection agencies) except from municipalities. UOP is currently discussing with many stakeholders, exploring various diverse ways to transfer the sense.city to the market. Whatever the final form of the service may be, the requirements for strong security and reliable functionality will be critical for the success of the product. SMESEC has already provided a good basis for building our security plans and implementing the first set of security measures. Even after the end of the project UOP aims to take advantage of its connections with the consortium partners and explore additional technical and business collaborations.

For securing its infrastructure and sense.city service, UOP selected the following SMESEC security tools and solutions. GravityZone (Antimalware-Antivirus) from Bitdefender, XL-SIEM from ATOS, EWIS (intrusion detection) and Cloud Security from FORTH, a Code analysis tool for javascript from IBM, and the TaaS tool from EGM. By M24 GravityZone, XL-SIEM, EWIS and Cloud Security are fully integrated, and can offer a general overview of the security status of the UOP infrastructure, while the rest of them are still in the deployment phase already providing early results. For the installation and evaluation of the SMESEC framework, UOP created a new set of nodes inside its private cloud. These new nodes were created to host the SMESEC tools and components.

XL-SIEM: UOP used the cloud version of XL-SIEM since a local installation seemed to be a highly complicated and time-consuming process. To be able to connect and communicate with ATOS cloud that hosted XL-SIEM, a new node was created inside the UOP cloud. In this node the UOP team installed an XL-SIEM agent specially configured for University of Patras. The purpose of this agent is to collect all events from the various SMESEC solutions installed in the smart city pilot and transmit them to ATOS cloud to be presented in the UOP XL-SIEM account.

GravityZone: For this solution UOP created a new node to host the GravityZone platform. After successful integration of the platform, and through a specific process described by Bitdefender's experts and manuals, the UOP team created a set of software agents that were installed across various nodes we wanted to protect inside the cloud. Once these agents were installed, they automatically connected to the GravityZone platform reporting any suspicious activity. Finally, the GravityZone platform was connect to the the UOP XL-SIEM agent reporting any security events that is collecting.

Honeypot: FORTH requested from UOP a new node in order to install its solutions inside the cloud. The honeypot installed in this new node, reporting security events to the EWIS platform and also to the XL-SIEM agent. UOP can see the list of these events in the XL-SIEM and more details about them in the unified SMESEC dashboard in the EWIS tool.

Cloud IDS: One of the solutions specifically requested by UOP inside the context of SMESEC project was cloud security. This solution was provided by FORTH. After some discussions between UOP and FORTH, it was decided to not directly deploy this solution directly in the UOP cloud, since the whole process required the installation of components on operational nodes, and UOP considered this to be risky. To address this issue, UOP setup a new physical machine with the same hypervisor as its private cloud (clone). This resided inside the same network, but outside the UOP private cloud. The solution was also connected to the UOP XL-SIEM agent.

Code analysis: For the IBM code analysis solution, UOP was a special case since its requirements were based on the analysis of javascript code. IBM was able to provide such a solution in the second year of the project. Since this process started late in the project its deployment is still ongoing with some early results already sent to UOP. Currently UOP is preparing a docker container with a copy of the sense.city service to send it to IBM for further analysis.

CYSEC: CYSEC is integrated inside the SMESEC framework and not inside the network of the smart city pilot. Through the SMESEC framework, the administrator can have access to CYSEC, do cybersecurity self-assessment, see the recommendations (in the specific area based on the priorities), and communicate with the relevant staff in the company. The CYSEC tool, was evaluated by UOP during a one-day workshop that took place in Patras, showing promising results.

TaaS: For the Test as a Service solution UOP has provided to EGM all requested information considering the architecture and code of the sense.city platform. Based on this input EGM will produce a set of tests for DDoS attacks and threat analysis.

The integration of SMESEC framework has been a smooth process. Our experience is that all installations were user friendly and easy to configure. Also, all partners have done their best to help the UOP team properly install and configure the selected tools in sense.city service. It must be noted that some of the consortium partners have done more that was required from them e.g. FHNW provided an extensive vulnerability analysis for the UOP infrastructure and also IBM and FORTH provided a javascript fuzzing and a Cloud Security tool respectively, solutions that were not part of their initial SMESEC toolkit. Furthermore, after the successful testing of the selected SMESEC tools, the UOP team has decided to extend the use of some of the tools to operational nodes hosting live services and platforms. In particular, currently we are deploying the GravityZone solution across various operational nodes inside the UOP cloud. Other services that are also examined to be used outside the isolated testing space are the EWIS system of FORTH and the XL-SIEM.

Considering the overall experience of participating in the SMESEC project, we can say that UOP has already increased its security levels. Many nodes are much better protected that they were in the beginning of the project. Also, solutions like the IDS are giving us very good insights on the types of attacks we are dealing with. For the 3rd year of the project our goal is to finalize the deployment of all remaining tools and evaluate what can be also used in our operational nodes. UOP is looking to have a clear improvement of its security status and take advantage of its connections with the consortium partners, exploring additional technical and business collaborations not only until the end of the project but also long after.

3.4 GridPocket

Most GridPocket work about the integration of the SMESEC framework consisted in (1) The replication of a typical GridPocket environment and (2) the installation of the different components of the SMESEC framework.

GridPocket replication environment

GridPocket first created a production environment specifically dedicated to the SMESEC project. The goal was to be able to use the SMESEC framework in the most realistic way in order to observe interests and limitations of such a framework. For this purpose, GridPocket environment was implemented based on four servers of Ubuntu instances mounted on the OVH cloud platform, a configured private network and an IP Failover:

1. Database server

Its role is to store energy data that will be further analysed. We installed MongoDB (NoSQL database) and configured the firewall to only accept connections through the configured private network. Data from a given user was stored in the database after his consent (approximately 6 months of consumption data extracted from a smart meter). The data were also saved on a detachable disk in case of the server became unavailable, so we could easily create a new instance and recover the data.

2. API server

Its role is to provide all the functionalities needed to run our platform such as getting and processing data, authentication, data analysis, etc... Microservices were the format used for our architecture using NodeJS as the server side Javascript runtime. A firewall was configured to accept communication only for microservices in the private network.

3. UI server

It provides all the resources needed for the implementation of our website UI. Website location was created at https://gridvalley.eu. The firewall was configured to access this service only in the private network.

4. Reverse proxy server

It has an interface role between the external network and the private network. When a user browses the website, the reverse proxy routes the request to the right server. The developers need to connect to the

VPN installed on this server in order to access the private network. The used tools were Nginx and Strongswan.

Integration of SMESEC tools

1. XL-SIEM

The installation of this tool was done on our reverse proxy server. This choice was strategically decided because it was the only server connected to the external network. Indeed, other components like BitDefender and IDS SNORT which are installed locally at GridPocket office, need to send data log to XL-SIEM. Thus, a modification to the firewall was necessary to allow this communication. The documentation from ATOS was clear and easy to follow even if the installation process needed some basic knowledge about GNU/Linux system and command lines.

2. BitDefender

For the installation of this tool, we allocated a hardware machine in GridPocket office. Citrix Hypervisor XenServer needed to be installed before a Virtual Machine containing BitDefender. Antivirus clients were installed on GridPocket laptops and were managed by Bitdefender.

3. TaaS

This tool did not require any installation. A conference call was made with EGM to understand respective technologies and needs. The goal was to define the appropriate tests that would be run on the online SMESEC framework. Issues concerning Gridpocket system (upgrade of our API that invalidated the tests) and the online EGM tool were solved in cooperation with EGM.

4. IDS SNORT

We used the Citrix Hypervisor XenServer on our hardware machine in GridPocket office. We installed a second Debian Virtual Machine where this tool would be running. The documentation from FORTH was clear and easy to follow even if the installation process needed some basic knowledge about GNU/Linux system and command lines. Communication with a tool specialist was necessary to complete the installation.

5. Citrix ADC

The documentation of the installation indicated different ways to implement the tool. It required an advanced level concerning systems/networks knowledge. As a result, the constant presence of a Citrix collaborator was required for almost each step. We first choose a main implementation on AWS (Amazon Web Services) cloud computing platform. For strategic reasons, we further decided to remove this installation and installed it successfully on our OVH cloud. A collaboration with a Citrix collaborator was necessary for this installation since OVH provided access to the bare-metal resources through the widely popular OpenStack private cloud solution, which uses KVM as Hypervisor and demands different deployment strategy than common Public cloud environments such as AWS or Microsoft Azure.

6. HoneyPot

In the first place, it was decided that the tool would be hosted on FORTH cloud and then connected to Citrix ADC. Finally, FORTH decided that the tool should be installed on our OVH cloud.

4 Adaptation of the SMESEC tools

This section presents a description of the adaptation of the SMESEC tool in the three iterations for integration in the pilots.

4.1 Atos

Plugins of Honeypot

In the scope of integrating the tools offered by SMESEC, FORTH provides different types of Honeypot. For each type, a new plugin has been developed to capture and correlate their events. More in concrete:

- FORTH Cloud-IDS. This plugin is able to capture 29 different types of events from the logs sent by FORTH Cloud-IDS. Specially important are the logs capable of detecting denial of service attacks, which immediately trigger alarms to the final user.
- FORTH Kippo Honeypot. This honeypot is specially designed to log SSH connection attempts. Because of this, this plugin only detects accepted SSH connections, sending events to the XL-SIEM every time an user accesses to the SSH service enabled by the honeypot.
- FORTH Cowrie Honeypot. Cowrie is a Kippo based honeypot with extended capabilities, included the possibility to allow the possible attackers to execute commands inside the honeypot. The plugin made for this honeypot is capable of logging the SSH connection successes and failures, along with the executed commands inside the honeypot.

Plugins for NetScaler

In order to provide correlation for the events captured in NetScaler, a plugin has been developed to provide integration between NetScaler and the XL-SIEM. This plugin is still subject to further changes and we expect to capture more types of events to provide more information to the users. At this moment, the plugin captures the following events:

- Executed commands.
- Failed requests.
- Socket timeouts.

Pilot-specific plugins: Scytl

In the scope of SMESEC, Scytl had some special needs regarding capturing some specific logs that were triggered by them in their web server.

For giving Scytl the ability to see and correlate those logs, two different plugins have been jointly developed by Atos and Scytl:

- Tomcat web server. The plugin created capture events that are raised in the following situations:
  - A user logs into the system.
  - A user successfully modifies an election process.
  - A user votes in the system.
- Apache web server. This plugin captures the following events present in Scytl’s infrastructure:
  - GET or POST requests that are proxied by the apache web server.
  - The steps made by a user when is voting.

Pilot-specific plugins: WorldSensing

In order to support IoT devices, changes were needed in the XL-SIEM agent, that previously consumed a big amount of memory and disk, that usually are not available in IoT devices. As stated in deliverable 3.4, in section 3.1.1 [1], functionalities and integrations were removed from the XL-SIEM agent. This allowed it to be installed in environments with memory and disk capacity restrictions.

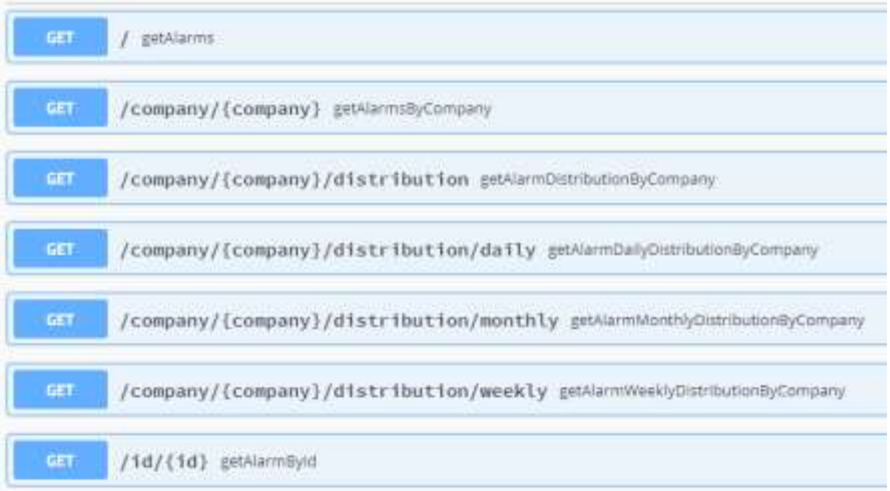
API of the XL-SIEM

In order to provide data for creating different charts in the SMESEC Framework dashboard without the need of including many different iframes, an API has been designed and implemented to provide this data to the SMESEC Framework.

This API supports query data of events and alerts present in the XL-SIEM database with some filters.

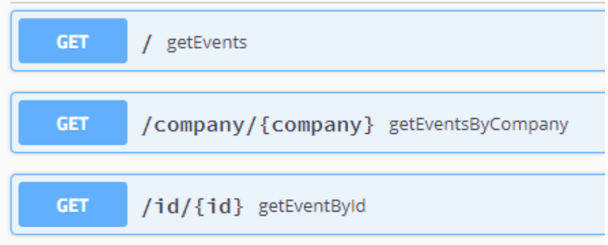
Specifically, the API provides the following endpoints:

- /api/v1/alarms



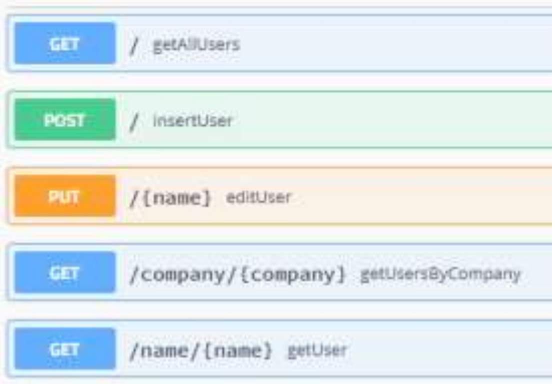
GET / getAlarms
GET /company/{company} getAlarmsByCompany
GET /company/{company}/distribution getAlarmDistributionByCompany
GET /company/{company}/distribution/daily getAlarmDailyDistributionByCompany
GET /company/{company}/distribution/monthly getAlarmMonthlyDistributionByCompany
GET /company/{company}/distribution/weekly getAlarmWeeklyDistributionByCompany
GET /id/{id} getAlarmById

- /api/v1/events



GET / getEvents
GET /company/{company} getEventsByCompany
GET /id/{id} getEventById

- /api/v1/users



GET / getAllUsers

POST / insertUser

PUT /{name} editUser

GET /company/{company} getUsersByCompany

GET /name/{name} getUser

RAE OpenID Authentication

For this project, single sign on has been enabled on the Risk Assessment Engine in order to support the OpenID [2] specification. This allowed the tool to be integrated seamlessly with the SMESEC Framework.

For implementing this authentication method, we have used the open source library provided by Mozilla, mozilla-django-oidc [3].

Frontend of the XL-SIEM

The SMESEC Framework aims to provide a unified experience to the users so it’s easier to manage the cybersecurity in the company. With that goal, the interface of the XL-SIEM was redesigned to provide a look and feel in line with the SMESEC branding created by FHNW.

Colors, fonts and some layouts were changed, as shown in the following screenshots:

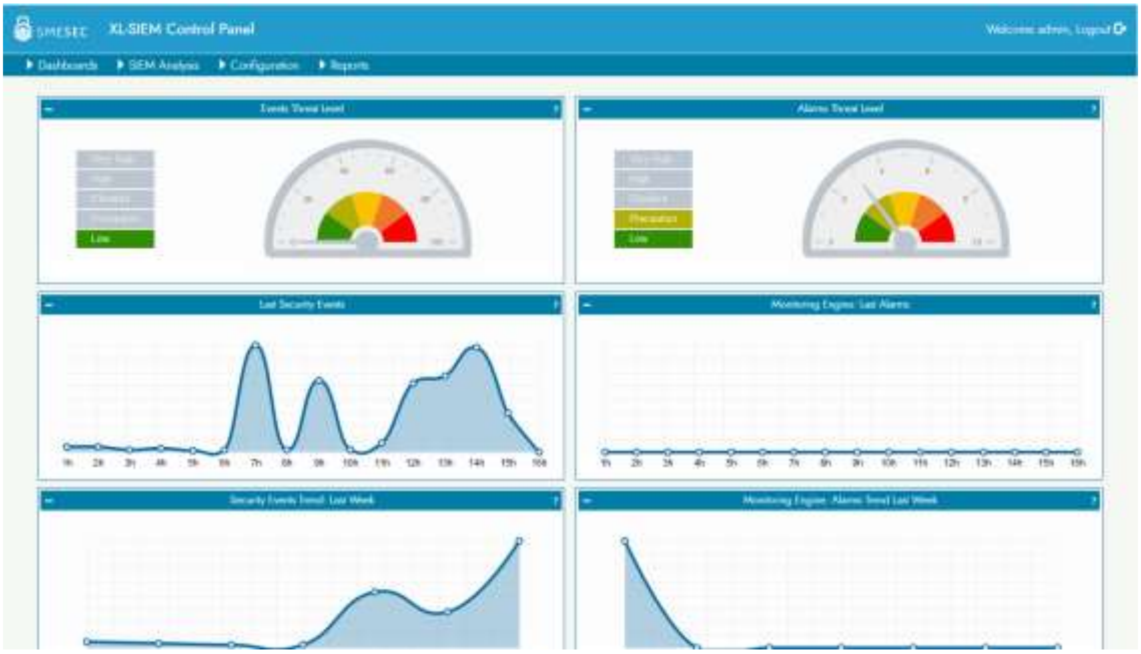


Figure 3 - XL-SIEM Dashboard after

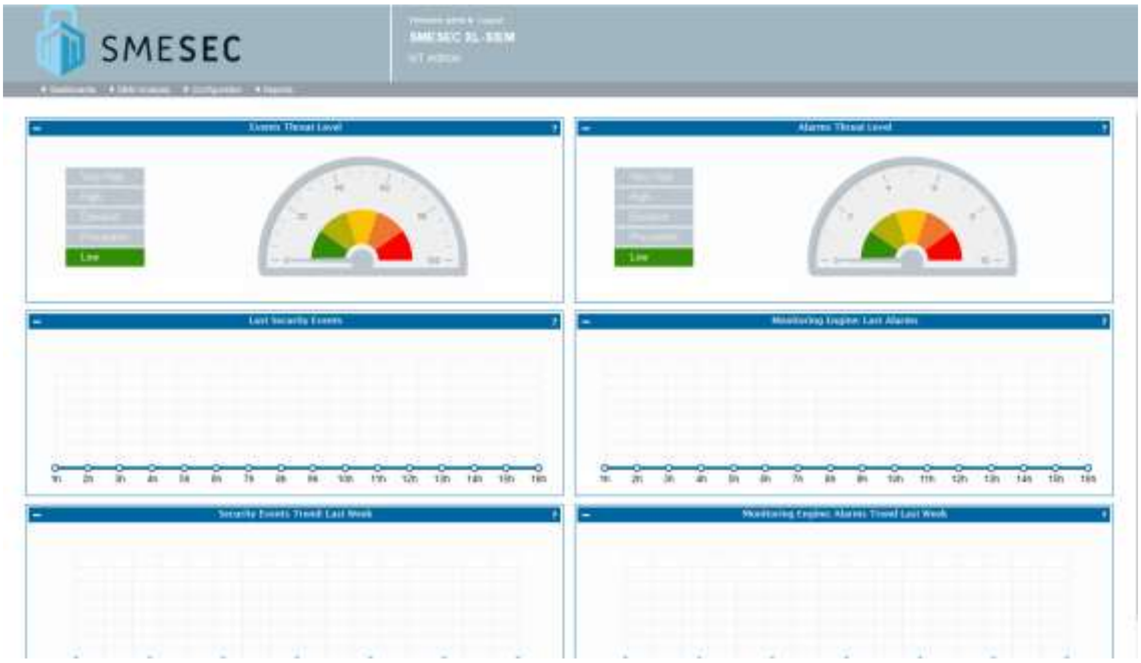
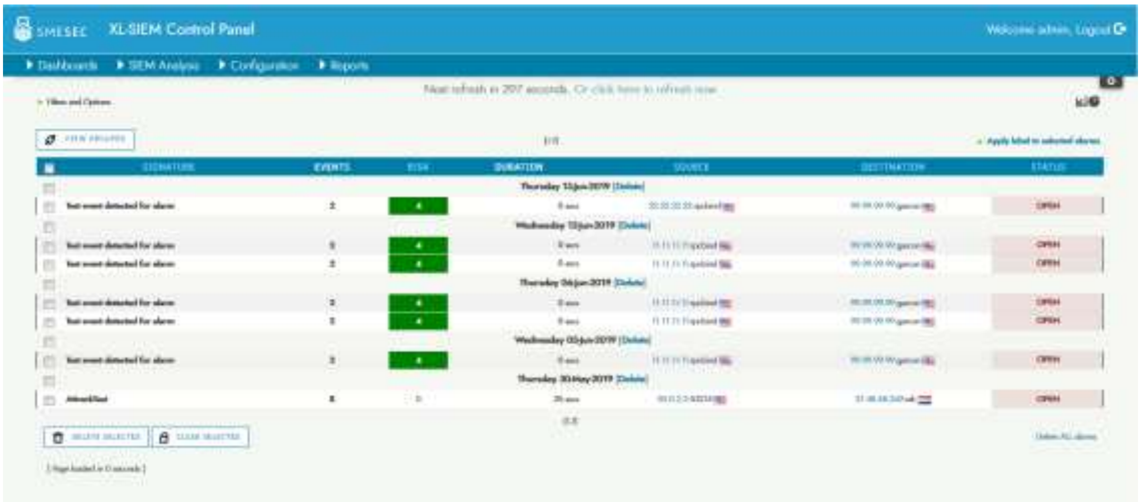


Figure 4 - XL-SIEM Dashboard before

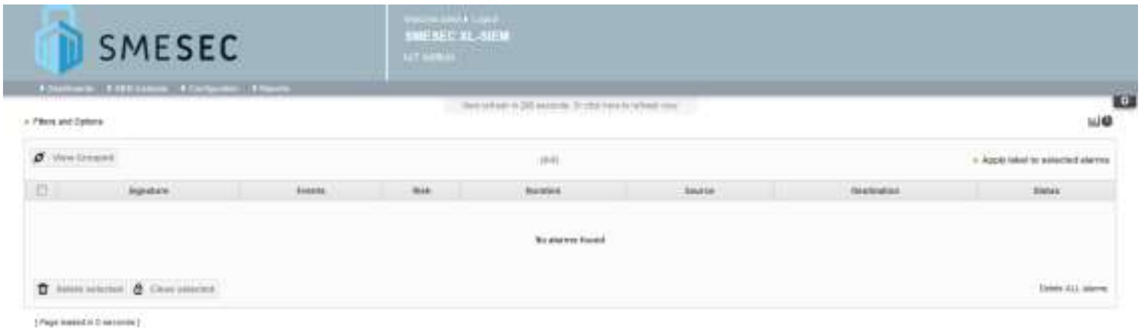


Next refresh in 297 seconds. Or click here to refresh now

☐	SIGNATURE	EVENTS	RISK	DURATION	SOURCE	DESTINATION	STATUS
☐				Thursday 13-Jun-2019 [Delete]			
☐	Test event detected for alarm	2	4	0 secs	[illegible]	[illegible]	OPEN
☐				Wednesday 12-Jun-2019 [Delete]			
☐	Test event detected for alarm	2	4	0 secs	[illegible]	[illegible]	OPEN
☐	Test event detected for alarm	2	4	0 secs	[illegible]	[illegible]	OPEN
☐				Thursday 06-Jun-2019 [Delete]			
☐	Test event detected for alarm	2	4	0 secs	[illegible]	[illegible]	OPEN
☐	Test event detected for alarm	2	4	0 secs	[illegible]	[illegible]	OPEN
☐				Wednesday 05-Jun-2019 [Delete]			
☐	Test event detected for alarm	2	4	0 secs	[illegible]	[illegible]	OPEN
☐				Thursday 30-May-2019 [Delete]			
☐	[illegible]	8	0	26 secs	[illegible]	[illegible]	OPEN

[Page loaded in 0 seconds]

Figure 5 - XL-SIEM Alarms table after



☐	Signature	Events	Risk	Duration	Source	Destination	Status
				No alarms found			

[Page loaded in 0 seconds]

Figure 6 - XL-SIEM Alarms table before

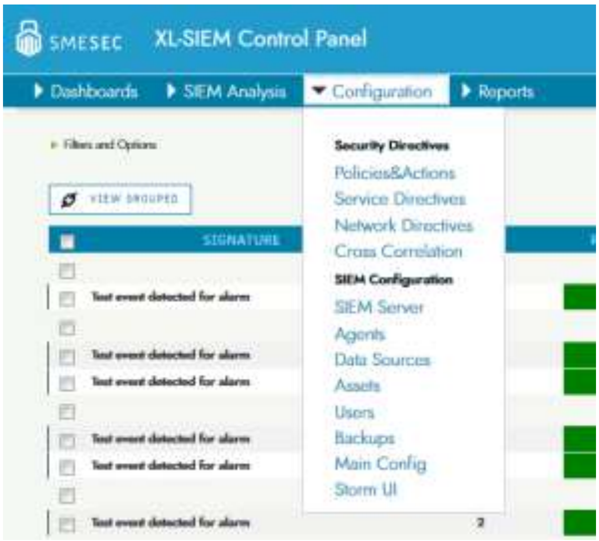


Figure 7 - XL-SIEM menu after

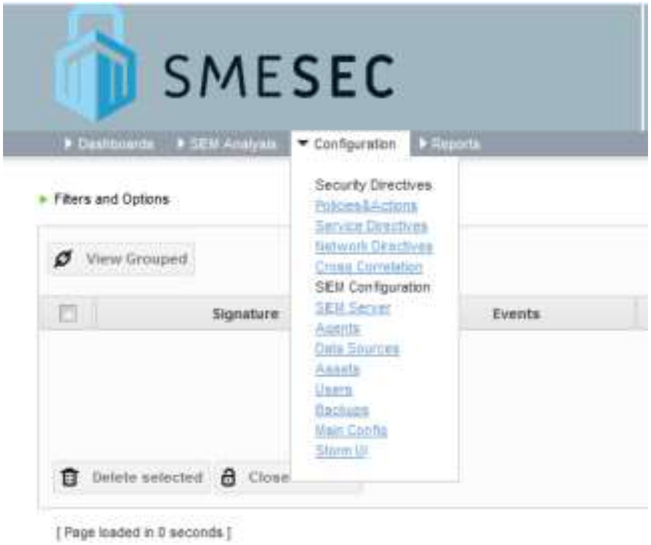


Figure 8 - XL-SIEM menu before

Still, the work in this section is not completed yet. Another iteration will be made in order to assure that the XL-SIEM offers the correct experience to the customers of the SMESEC Framework.

Frontend of the RAE

Also, the Risk Assessment Engine interface has been modified to fit the SMESEC branding. Below, there can be found screenshots of different screens of the tool:

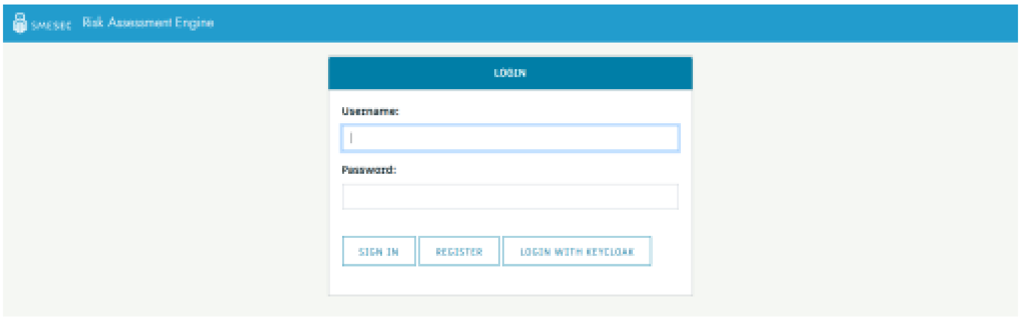
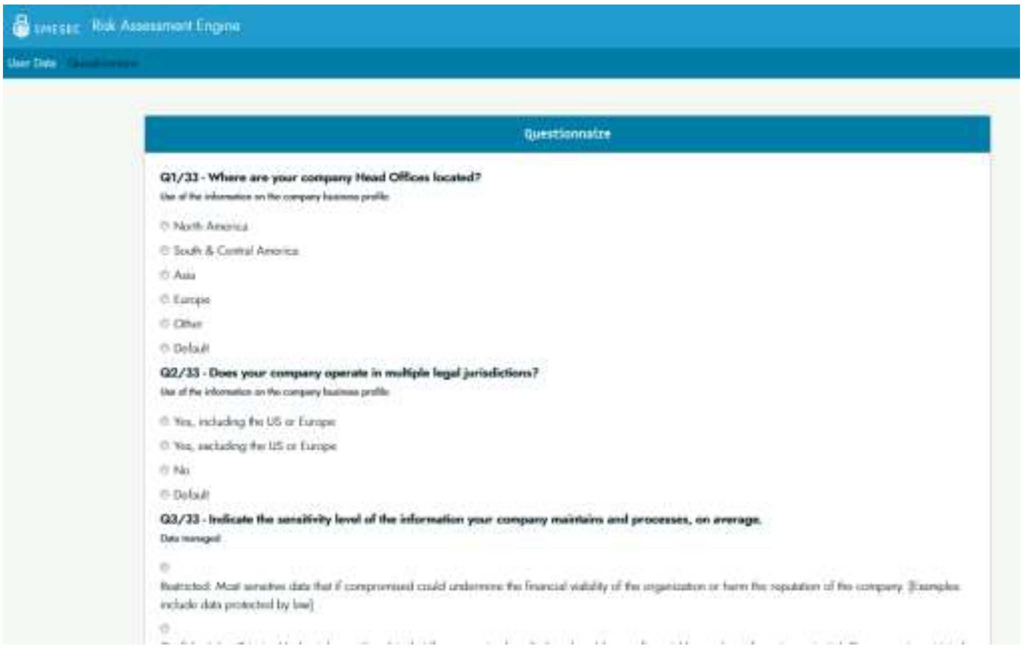


Figure 9 - RAE login page



Q1/33 - Where are your company Head Offices located?

Use of the information on the company business profile

○ North America

○ South & Central America

○ Asia

○ Europe

○ Other

○ Default

Q2/33 - Does your company operate in multiple legal jurisdictions?

Use of the information on the company business profile

○ Yes, including the US or Europe

○ Yes, excluding the US or Europe

○ No

○ Default

Q3/33 - Indicate the sensitivity level of the information your company maintains and processes, on average.

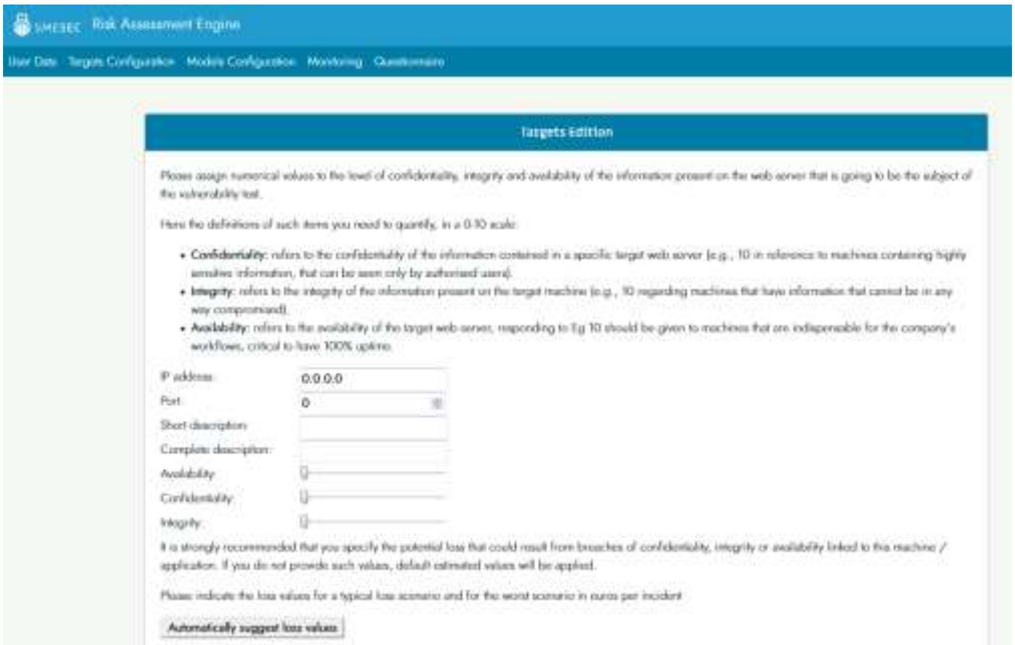
Data managed

○

Restricted: Most sensitive data that if compromised could undermine the financial viability of the organization or harm the reputation of the company. [Examples include data protected by law]

○

Figure 10 - RAE questionnaire page



Please assign numerical values to the level of confidentiality, integrity and availability of the information present on the web server that is going to be the subject of the vulnerability test.

Here the definitions of such items you need to quantify, in a 0-10 scale:

- **Confidentiality**: refers to the confidentiality of the information contained in a specific target web server (e.g., 10 in reference to machines containing highly sensitive information, that can be seen only by authorized users).
- **Integrity**: refers to the integrity of the information present on the target machine (e.g., 10 regarding machines that have information that cannot be in any way compromised).
- **Availability**: refers to the availability of the target web server, responding to: Eg 10 should be given to machines that are indispensable for the company's workflows, critical to have 100% uptime.

IP address: 0.0.0.0

Port: 0

Short description:

Complete description:

Availability:

Confidentiality:

Integrity:

It is strongly recommended that you specify the potential loss that could result from breaches of confidentiality, integrity or availability linked to this machine / application. If you do not provide such values, default estimated values will be applied.

Please indicate the loss values for a typical loss scenario and for the worst scenario in euros per incident

Figure 11 - RAE targets edit page

4.2 FORTH

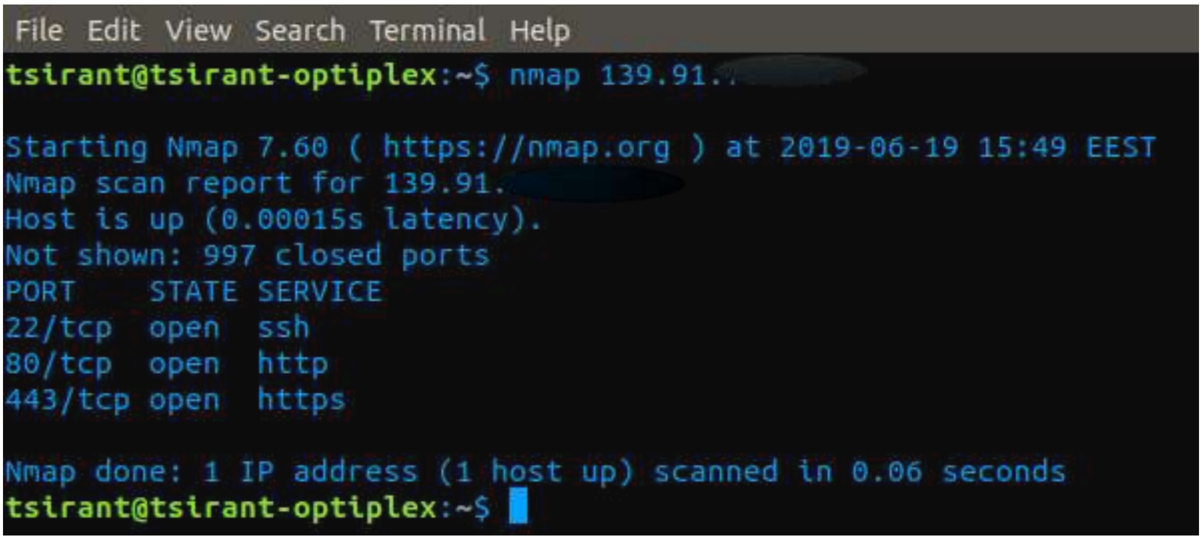
Forth Cloud IDS

Our cloud IDS implementation was tested and working with the Xen hypervisor. In order to support the deployment activities of the SMESEC framework, FORTH updated the Cloud-IDS solution and its ability to natively support the KVM hypervisor. To achieve that, we created a new instance of the KVM hypervisor onto a fresh Linux installation. Afterwards the appropriated configurations for the network bridges, which enable the monitoring of the communications between the VMs, was successfully concluded. Moreover, the installation of an IDS system (snort) in the hypervisor that is able to monitor all the traffic, using specific rules that can detect attacks between VMs, was done. Finally, cloud IDS communicates with the XL-SIEM and the SMESEC dashboard.

For each Cloud-IDS installation the snort rules have to be adjusted depending the size of the network traffic for each SME.

Forth IoT-Honeypot

In order to be able to protect the WoS' pilot in Patras, FORTH created a new flavour of our Honeypot solution, able to emulate an IOT gateway. Firstly, we analysed the already deployed IoT GW and we constructed a custom configuration for our SSH honeypot, that is able to emulate the same ports and similar functionality to the ones of the deployed IOT GW. This implementation allows us to return the same results for our honeypot and the GW to a potential network scan of an attacker.



```
tsirant@tsirant-optiplex:~$ nmap 139.91.[redacted]

Starting Nmap 7.60 ( https://nmap.org ) at 2019-06-19 15:49 EEST
Nmap scan report for 139.91.[redacted]
Host is up (0.00015s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
tsirant@tsirant-optiplex:~$ █
```

Figure 12. Example of a network scan of our iotHoneypot

In addition, changes were made in the default settings of the virtual environment of the honeypot, such as default users, hostname and more in order to better mimic the IoT Gateway after an attacker "successfully" compromises our honeypot. We configured the output connections of our honeypot for both the syslog output to the XL-SIEM and our backend databases for communications with the SMESEC dashboard.

Finally, we deployed specific rules that are able to detect volume based (D)DoS attacks. To accomplish that we installed an IDS system (snort) and a custom ruleset which can detect spikes in the network traffic and report the DDoS attacks to the XL-SIEM and our unified dashboard.

Forth EWIS Dashboard

Numerous changes have been applied to FORTH EWIS Dashboard in order to integrate to SMESEC Framework. We enabled Keycloak authentication sign in, in order to have single sign in the unified dashboard. Also, changes were applied to the css of FORTH EWIS Dashboard to line up with the general branding and give a unified experience to the user.

For functionality reasons changes were made to various screens of FORTH EWIS Dashboard.

In Home screen the user can monitor the traffic of all the honeypots activated per hour for the Last Day, Last Week, Last Month Grouped or Stacked. Furthermore, there is a notification when new traffic is captured as well as a new Screen "Summary Traffic" where basic plots are demonstrated for the Last Hour. Moreover, a new functionality has been added prompting to the user when new events arrive while he is connected to our platform.

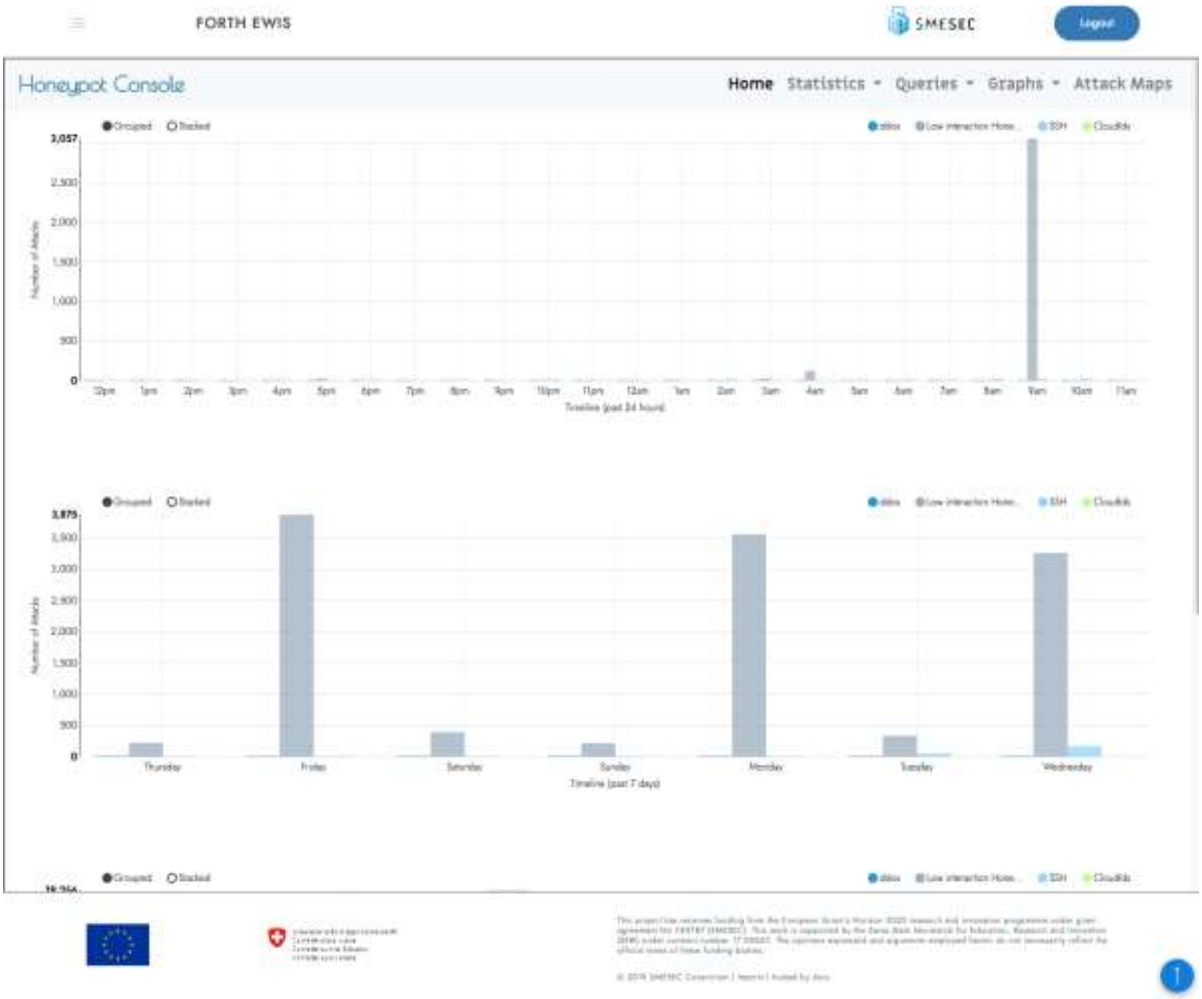


Figure 13. New interface of Honeypot

The screens IP/Port Statistics, IP Look up, Port Look up, Attack Maps have enriched their content with data from SSH, Low interaction Honeypot and DDOS traffic. New screens were added to monitor the traffic captured from IOT Honeypot and Cloud IDS Honeypot.

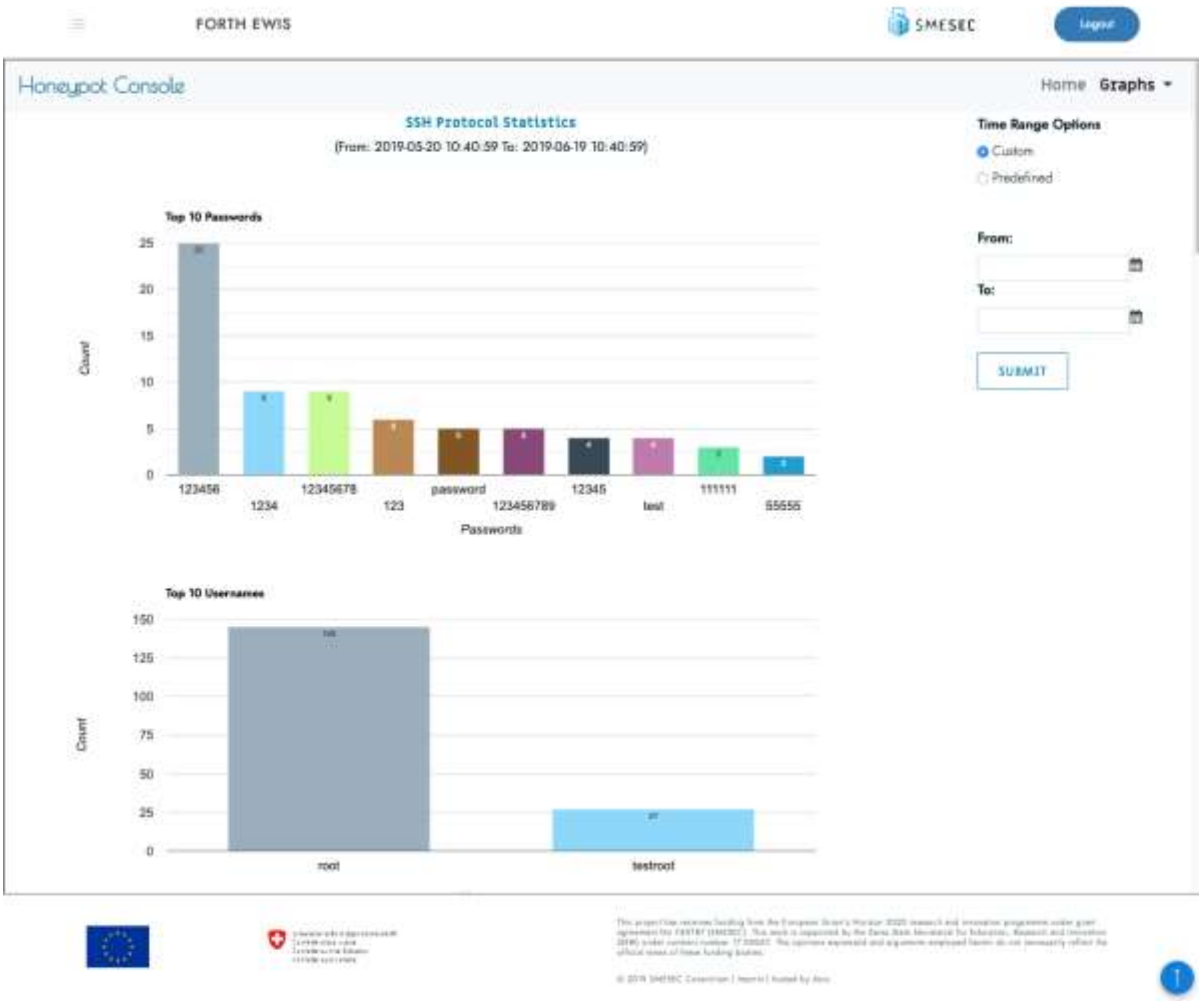


Figure 14. New interface for traffic capturing

4.3 EGM

TaaS existing product before SMESEC project

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. It is based on MBT (Model based testing) for generation of test cases. 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. Figure 15 and Figure 16 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.

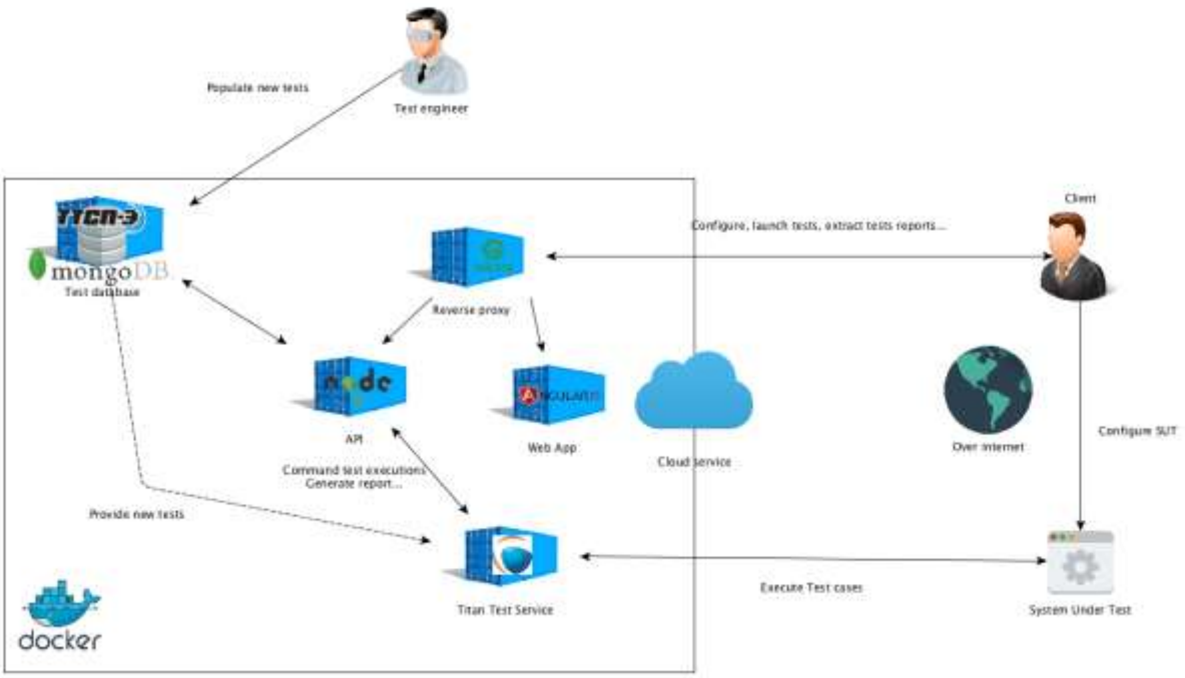


Figure 15. EGM TaaS Architecture

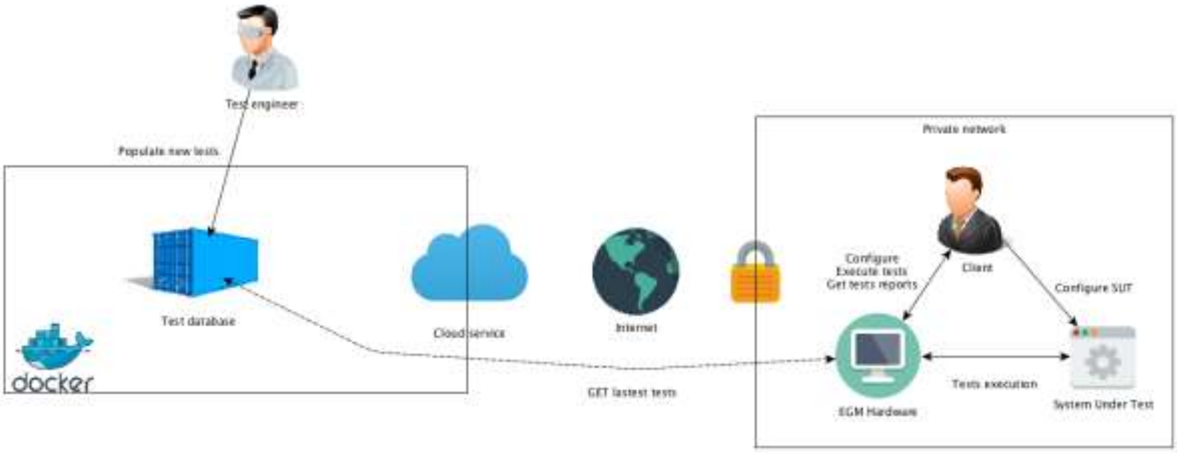


Figure 16. EGM offline testing

TaaS integration with SMESEC project

To fully meet the SMESEC project technical requirements, described in D2.1, some further configuration (add or adapt some architecture blocks) is required:

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

- Add two test services (API and LoRa) to meet the pilot's requirements: The TaaS, as described in the previous section, implements a micro-service architecture, where each test service (oneM2M, API, semantic validation …) is represented as a micro-service in the global architecture. A test coordinator service is also implemented, which play the role of the orchestrator between the different services, if any interaction is needed (for example, any test service with the reporting service). Figure 17 shows what we have described. Both test services (API and Lora) has been described in the deliverable D3.4.

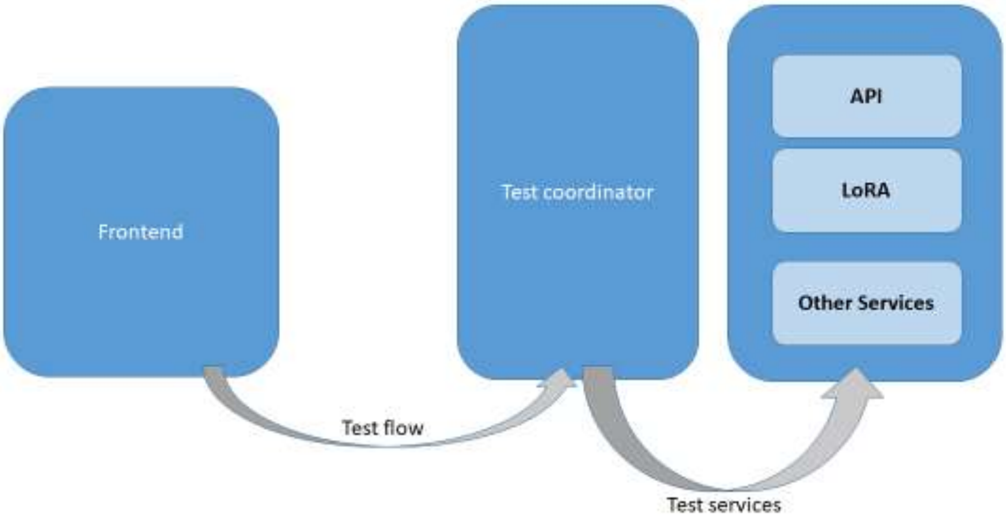


Figure 17. TaaS Micro service Architecture

4.4 CITRIX

Integrating Citrix ADC to the SMESEC Framework

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

The full spectrum of Citrix ADC security and protection features efficiently protects web applications from Application Layer attacks. An ADC appliance allows legitimate client requests and can block malicious requests. It provides built-in defences against denial-of-service (DoS) attacks and supports features that protect against legitimate surges in application traffic that would otherwise overwhelm the servers. An available built-in firewall protects web applications from Application Layer attacks, including buffer overflow exploits, SQL injection attempts, and cross-site scripting attacks.

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

Load Balancing (LB) virtual server

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

Content Switching (CS) virtual server

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

SSL virtual server

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

Delivering extended CyberSecurity services in the freemium subscription

Under the auspices of SMESEC project, Citrix had pledged providing thousands of USD worth of licenses for some of their most popular services delivered through Citrix ADC. However, as it turned out, small SMEs are reluctant or even incapable of paying such a premium, regardless of the significant cybersecurity boost it provides. To tackle this situation, we have focused on providing similar services through selective and meticulously deployed LB, CS and SSL virtual servers that efficiently protect the SME servers from most application layer attacks, while always **remain in the Free-Tier offering of Citrix ADC**. In the unlikely case that the cybersecurity requirements of an SME exceed the provided solution, a license which unlocks additional features must be purchased. The deployed solution is based on Citrix ADC Express and its only functional limitations are 20Mbps throughput and 250 concurrent SSL connections. Virtual server functionality is not compromised or affected, therefore cybersecurity protection resembles to the one of the full-blown Citrix ADC packet.

Pilot Adaptation and generic deployment blueprints

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

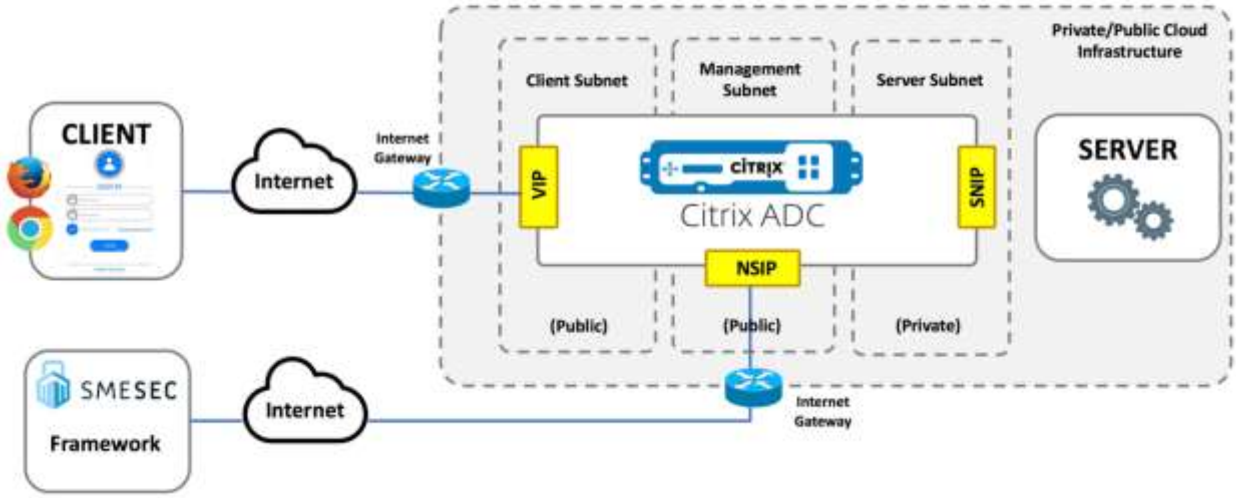


Figure 18. High-level network topology for Citrix ADC deployment in SMESEC Pilots and Trials

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

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

Citrix ADC Aggregator

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

Design

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

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

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

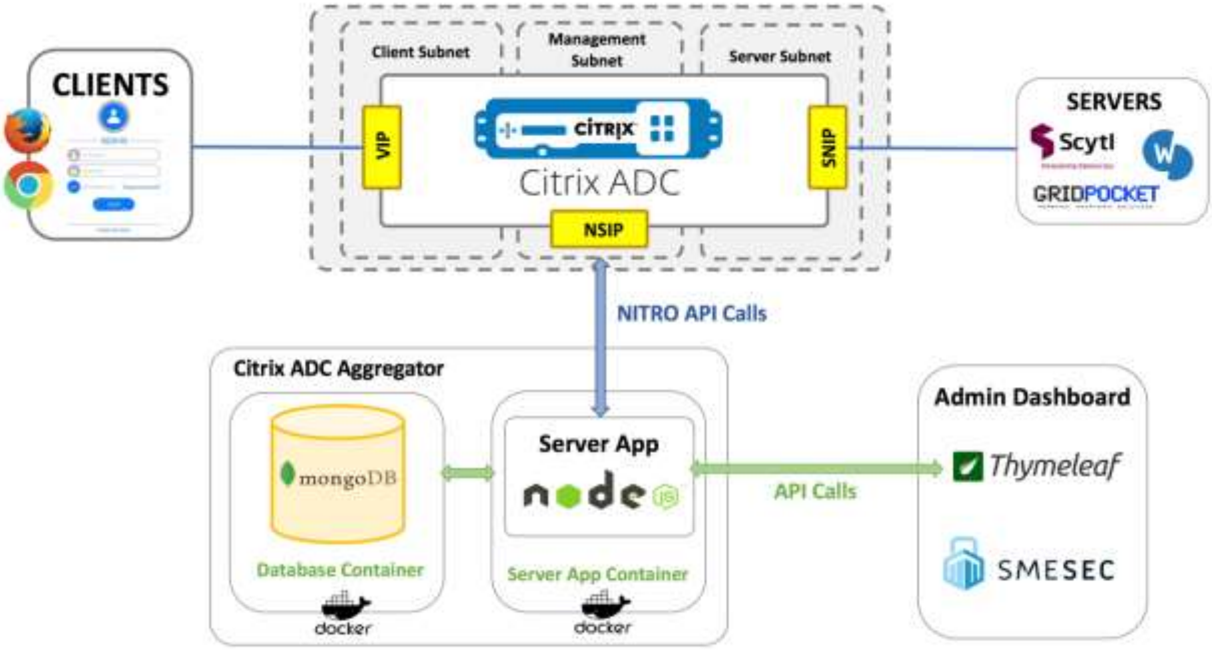


Figure 19. Deploying Citrix ADC Aggregator

Business Opportunity for Penetrating the SME Cybersecurity Market

Using Citrix ADC Express in AWS, MS Azure and Google Cloud Engine, Cybersecurity costs are now limited to the cost of maintaining an active Citrix ADC VPX instance and do not exceed 200 Euro per month for up to 20Mbps throughput and 250 concurrent SSL connections, depending on the instance size.

4.5 BitDefender

GravityZone is a business security solution built from ground-up for virtualization and cloud to deliver security services to physical endpoints, mobile devices, virtual machines in private, public cloud and Exchange mail servers [4].

GravityZone is one product with a unified management console available in the cloud, hosted by Bitdefender, or as one virtual appliance to be installed on SMEs' premises, and it provides a single point for deploying, enforcing and managing security policies for any number of endpoints and of any type, in any location.

The unique architecture of GravityZone allows the solution to scale with ease and secure any number of systems. GravityZone can be configured to use multiple virtual appliances and multiple instances of specific roles (Database, Communication Server, Update Server and Web Console) to ensure reliability and scalability.

The GravityZone solution includes the following components:

- GravityZone Virtual Appliance with the available roles:
  - Database.
  - Update Server.
  - Communication Server.
  - Web Console (Control Centre)
- Report Builder Virtual Appliance with the available roles:
  - Database.
  - Processors.
- Security Server.
- HVI Supplemental Pack.
- Security Agents.

Within the SMESEC security framework, Bitdefender GravityZone offers protection and security at the endpoint level, providing the best solution in class against internal and external attackers.

The Architecture is composed of two main components, the management console called GravityZone and the endpoint agents called Bitdefender Endpoint Security Tools (BEST).

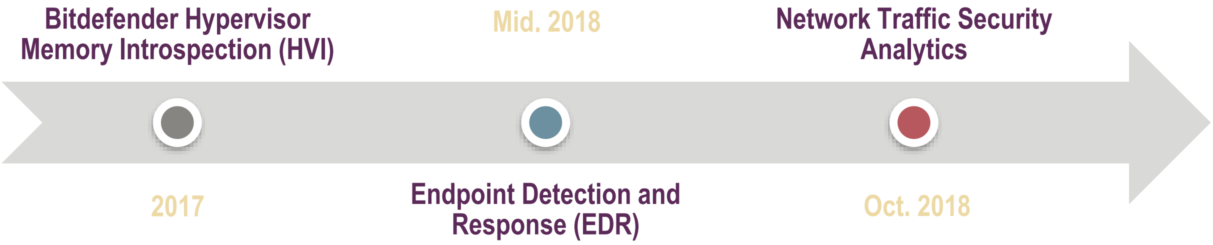
The BEST agent monitors the systems and reports back to the management console. The management console, receives the events, stores them into the database and forwards the events to the XL-SIEM.

Along the implementation of the SMESEC framework, the Bitdefender GravityZone went through multiple updates and significant additions in terms of unique innovations.

Implemented and refined across 2017, the Bitdefender Hypervisor Memory Introspection (HVI) is a ground-breaking solution that detects suspicious activities by working directly with raw memory at hypervisor level – a level of insight from which malware cannot hide. The HVI protects virtual machines in data-centres against advanced and sophisticated threats that signature-based engines cannot defeat. HVI enforces strong isolation, ensuring real-time detection of attacks, blocking them as they happen and immediately removing the threats.

Whether the protected machine is a server or a workstation, HVI provides insight at a level that is impossible to achieve from within the guest operating system. By operating at the hypervisor level and leveraging the hypervisor functionalities, HVI overcomes technical challenges of traditional security to reveal malicious activity in datacentres.

By working alongside any endpoint protection (EPP) solution, it provides an unprecedented layer of defence for the most notorious Advanced Persistent Threats (APTs) targeting SMEs and start-ups.


Bitdefender Hypervisor Memory Introspection (HVI)
2017
Mid. 2018
Endpoint Detection and Response (EDR)
Network Traffic Security Analytics
Oct. 2018

Within the last couple of months of SMESEC implementation, BD GZ integrates layered next-gen endpoint protection and easy-to-use EDR platform to accurately protect SMEs against even the most elusive cyber threats. It offers prevention, automated detection, investigation and response tools so even the customers with tight budgets can protect their digital assets and respond to these threats. More, new technologies refinements are conducted each month, these being integrated with the constant updates or additions.

The Bitdefender Network Traffic Security Analytics accurately detects breaches and provides insights into advanced attacks by analysing network traffic. It lets organizations quickly detect and fight sophisticated threats by complementing pre-existing security architecture – network and endpoint – with specialized network-based defence.

It uses AI (Artificial Intelligence) / ML (Machine Learning) and heuristics to analyse network meta-data in real-time and to accurately reveal threat activity and suspicious traffic patterns. With flexible deployment options, Bitdefender Network Traffic Security Analytics is a plug-and-play, out-of-band solution, that focuses on outbound traffic and enables analysis over longer periods of time to accurately detect the most sophisticated malware and APTs with high fidelity.

5 Conclusions

This deliverable presents the work done and the process followed in the three iterations done till M24 in the project for the integration of the SMESEC Framework, and more specifically, SMESEC tools in the pilots and the impact the feedback had for refining them.

The process we followed for integration was a continuous process with specific goals and work in each of the iterations. The first one focused in initial integration of some of the planned tools for each use case (after their analysis for fulfilling the requirements), together with work in the design of the internal components of the SMESEC Framework. The second iteration continued the integration of tools as planned, refinement of the initial ones already integrated (focusing in usability) and start with the integration of common components of the SMESEC Framework such as the authentication component. The third iteration focused in the refinement of tools from the point of view of usability, material for integration and configuration, and their integration in the rest of common components of the SMSEC Framework such as the unified dashboard and APIs for communication.

The use cases had to follow a specific approach, each of them their own, for integrating the SMESEC tools. Bearing in mind some of the SMEs were lacking technical experts that could integrate these tools we had several workshops and meetings in order to provide the necessary support for the correct deployment, configuration and use. This was a critical exercise as we expect from the SMEs to be in a similar situation and, therefore, this was a very useful training for our objective. The material, work and functionalities of the SMESEC tools were hugely improved with the three iterations, having more work in the second and third one.

The integration of the SMESEC components was also an interesting work for the tool owners. Each of them have an unique method for authentication in their tools so they have to adapt them all to the common one used in SMESEC. Also, providing an API with some of their data was a good exercise as the tools are now more open and available for data sharing than they were before. This will open new business opportunities in the project that were not foreseen at the beginning of it.

The refinement of the tools was also done at interface level, using all a common look and feel (SMESEC look & feel) that improves greatly the usage of the tools.

Finally, this work will continue in the next year as we will continue refining the tools in the open call (where several SMEs with very different needs and business elements) will use the SMESEC Framework and the validation of the pilots of the project.

6 References

[1] **SMESEC D3.4** SMESEC products integration on the Unified Architecture , section 3.3.1.

[2] **OpenID Connect specification.** https://openid.net/specs/openid-connect-core-1_0.html

[3] https://github.com/mozilla/mozilla-django-oidc

[4] https://www.bitdefender.com/support/how-to-deploy-gravityzone-in-nutanix-1780.html