Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

SMESEC

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

D4.1 Preliminary integration report on e-Voting SME pilot

Document Identification			
Status	Final	**Due Date**	30/11/2018
Version	1.0	**Submission Date**	30/11/2018

Related WP	WP4	**Document Reference**	D4.1
Related Deliverable(s)	D2.1, D3.1	**Dissemination Level (*)**	PU
Lead Organization	SCYTL	**Lead Author**	Noemi Folch
Contributors	CITRIX, FORTH, ATOS	**Reviewers**	Christos Tselios (Citrix) Papa Niamadio (GridPocket)

Keywords:

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.

security, system, design, architecture, integration, WP4, requirements, goals, innovation, use case, e-voting, protection, defence, management.

Document Information

List of Contributors	
Name	Partner
Noemí Folch	SCYTL
Jordi Cucurull	SCYTL
Pau Julià	SCYTL
Manos Athanatos, Christos Papachristos, Sotiris Ioannidis	FORTH
Christos Tselios	CITRIX

Document History			
Version	Date	Change editors	Changes
0.1	11/10/2018	Jose Fran. Ruiz (Atos)	Table of contents template for all use case partners
0.2	30/10/2018	Noemi Folch (Scytl)	Use case information
0.3	2/11/2018	Manos Athanatos (Forth)	Forth contribution
0.4	5/11/2018	Christos Tselios (Citrix)	Citrix Contribution
0.5	15/11/2018	Christos Tselios (Citrix)	First review
0.6		Noemi Folch (Scytl)	Update with comments
0.7	29/11/2018	Papa Niamadio (GridPocket)	Second review
1.0			FINAL VERSION TO BE SUBMITTED

Quality Control		
Role	Who (Partner short name)	Approval Date
Deliverable leader	Name (Organisation short name)	dd/mm/yyyy
Technical manager		
Quality manager		

Project Manager		

Table of Contents

List of Figures

List of Acronyms

Abbreviation / acronym	Description
API	Application Programming Interface
AWS	Amazon Web Services
BCP	Business Continuity Plan
CLI	Command Line Interface
CWE	Common Weakness Enumeration
DB	Database
DMZ	Demilitarized Zone
DNS	Domain Name System
DRP	Disaster Recovery Plan
Dx.y	Deliverable number y belonging to WP x
EC	European Commission
EC2	Elastic Compute Cloud
EWIS	Electrical Wiring Interconnection System
FE	Frontend
FTP	File Trasnfer Protocol
GDPR	General Data Protection Regulation
GHDB	Google Hacking Database
GSLB	Global Server Load Balance
HA pair	High Availability Pair
HTTP	Hyper Text Transfer Protocol
HTTPS	HTTP Secure
IP	Internet Protocol
IT	Information Technology
JSON	JavaScript Object Notation
MIP	Mobile Internet Protocol
NetBIOS	Network Basic Input/Output System
NTP	Network Time Protocol

Abbreviation / acronym	Description
OWASP	Open Web Application Security Project
PKCS	Public Key Cryptography Standards
RCP	Remote Copy Protocol
REST	Representational state transfer
ROTI	Report of Test and Inspection
SIEM	Security Information and Event Management
SMB	Server Message Block
SME	Small Medium Enterprise
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SSH	Secure Shell
TelNet	Telecommunication Network
TFTP	Trivial Files Transfer Protocol
VM	virtual machine
VP	Vice-President
VPC	Virtual Private Cloud
VPN	Virtual Private Network
WAF	Web Application Firewall
XML	Extensible Mark-up Language

Executive Summary

This deliverable describes the preliminary efforts carried out during the first 18 months of the project for integrating the SMESEC Framework in e-Voting SME pilot (pilot I). This report has been generated at an early stage of the SMESEC Framework development. Therefore, the framework itself has not been integrated, but some of its tools instead. But still, the results obtained so far already validate the objective of improving the security of the e-Voting platform at infrastructure level.

This document describes in deep the specific e-Voting use case within SMESEC project, with all the tools implemented so far, and the business plan; and also, the next steps to be completed, comprising the framework integration as well as the initial validation plan.

In summary, this document describes the existing first version of e-Voting SME pilot. A final document will be provided on month 24, with the details of the final version of the e-Voting pilot, which will comprise the integration of the final SMESEC Framework.

1 Introduction

1.1 Purpose of the document

This is the first deliverable of WP4 “Integration of SMESEC security framework to e-Voting, Smart City, Industrial Services and Smart Grids pilots” related to e-Voting pilot. The role of this WP in the SMESEC project is to adapt the SMESEC security framework prototype to the different pilots proposed in the project.

Specifically, D4.1 provides a detailed description of the preliminary efforts carried out to integrate the SMESEC security framework into Pilot I (e-Voting pilot), with the aim of hardening, monitoring, preventing and detecting attacks and ensuring the availability of the election process.

1.2 Relation to other project work

As described before, this document covers the preliminary efforts carried out to integrate the SMESEC security framework into e-Voting pilot. This is the preliminary baseline on which other deliverables and Work Packages will be based, and more specifically:

- D4.2 will describe the final integration of e-Voting SME pilot
- WP5 will evaluate the integration work described in this document and D4.2

1.3 Structure of the document

This document is structured in 6 major chapters

Chapter 1 presents an introduction to the use case and its integration with SMESEC

Chapter 2 presents an update of requirements and needs of the use case, with respect to D2.1

Chapter 3 presents a description of the use case: architecture, diagrams, usage, roles, users, scenarios, threats and impact, cybersecurity training and awareness status

Chapter 4 presents the integration of SMESEC Framework: process, tools integrated, testing, feedback, updated use case with integration of the framework

Chapter 5 presents the next steps: integration of the business in the SMESEC Framework, usage of tools, needs of training, testing and validation plan

Chapter 6 presents the conclusions: experience of the initial integration of SMESEC, fulfilling of objectives, user-friendliness, general feedback and thinks to improve for the next version.

2 Update of requirements and needs

There has been no update in the requirements with respect to those identified in the document D2.1 "SMESEC security characteristics description, security and market analysis report", section 2.3.2 "List of requirements for e-Voting Pilot".

3 Description of the use case

Scytl's Online Voting product is a secure solution that enables voters to securely and easily cast their v ote from any location or device with a stable Internet connection. This solution enfranchises all voters, including remotely located ones, while ensuring privacy and results' integrity.

The voting solution also provides the following remarkable features:

- End-to-end encryption: For protecting the voter's privacy even when facing the election servers, the votes are encrypted on the client side. Thus, when the vote is received it is already encrypted and the server cannot know the votes' contents. The votes' contents are not decrypted until the end of the election.
- Mixing and decryption: In addition to being encrypted in the voters' device, votes cannot be decrypted until they are anonymized. To do so, a mixing process is used that shuffles and decrypts the votes to which the voter identity has been previously removed, making it impossible to link the mixed vote to the identity of the voter who has cast it. Only when votes are mixed can they be decrypted. To ensure this, the decryption process requires the participation of several members of the electoral board. At the beginning of the election, the election key that allows the decryption of the votes is divided into shares and stored in smartcards (one for each member). Therefore, only by gathering together a minimum threshold of the board (with their smartcards) is it possible to decrypt the individual votes.
- Key roaming: Since voters usually do not have cryptographic keys in their computers or in card identities (or they do not know how to use them), a mechanism called key roaming is implemented. This mechanism provides the voter with a keystore that contains personalized keys to be used to sign the votes. These keystores are protected with a derivation of the voter's credentials.
- Immutable logs: The voting back-office generates immutable logs for all critical operations, i.e. logs that are cryptographically protected against manipulation.

3.1 Architecture and design

The following diagram shows an overview of the online voting solution architecture, representing the main components of the system and their interaction with the different actors:

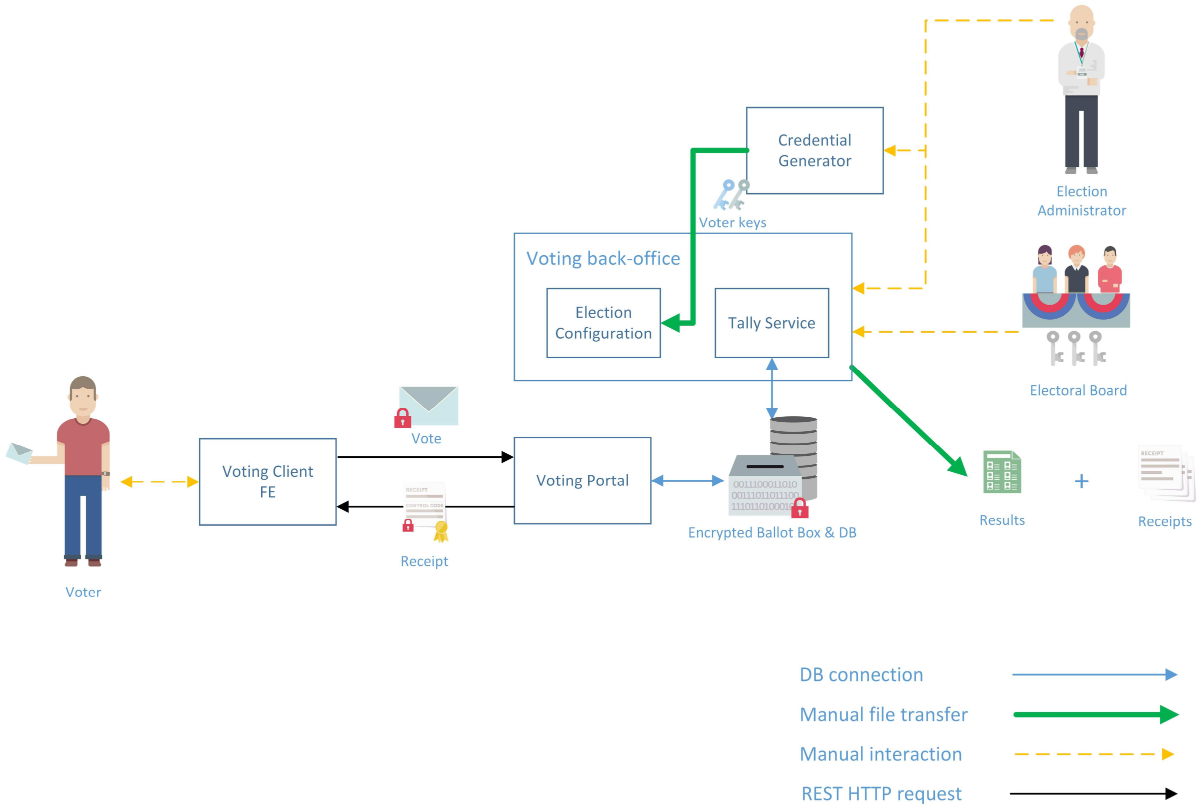


Figure 1: online voting solution architecture

There are two types of modules, those residing on the backend (Credential Generator, Voting back-office, Voting Portal and Receipts back-office) and the ones that are located on the client-side (Credential Delivery Portal, Credential Delivery back-office, Voting Client and Receipt Website). The interaction of these types of modules, hence the communication between the backend and the frontend is performed via HTTP-REST APIs. The remaining interactions are conducted manually or through the database.

3.1.1 Credential generation

The credential generator is a command-line tool that creates a pre-defined number of voter credentials and their corresponding voter keys. To generate credentials for an Election, the Credentials Generator requires a series of information, such as the Electoral census, the number of credentials to be generated, the expiration time of the credentials, etc. depending on the election.
The Credentials Generator is automatic. After requesting the parameters or input files and validating the configuration, it will start generating credentials in the output folder configured before.

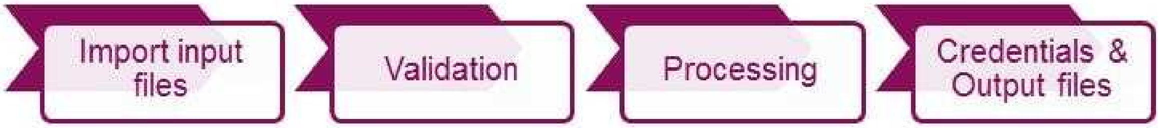


Figure 2: Credential generation process

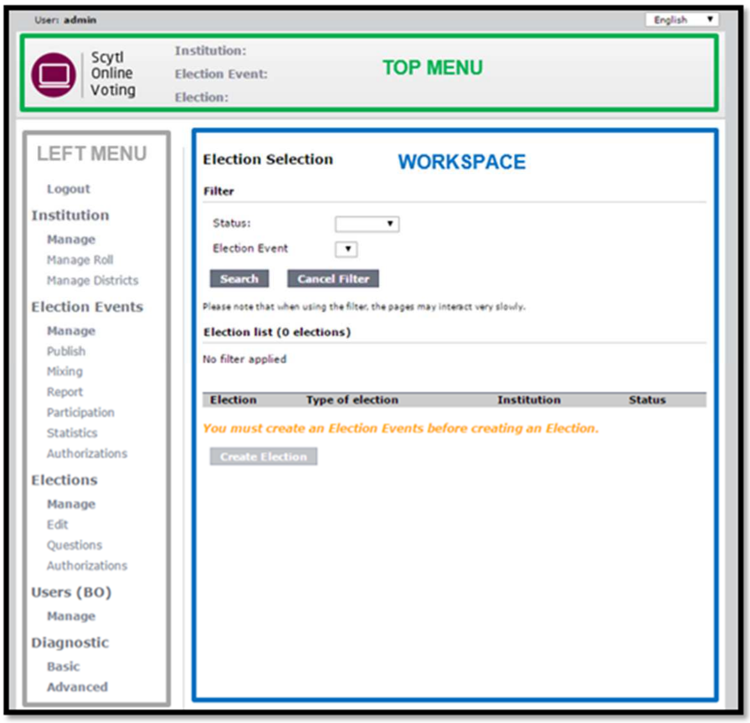
The credentials and voter keys are stored in a number of files:

1. List of voter credentials: file that contains the voter credentials to be delivered to the voters. There is one version in plain-text and another one encrypted for the credential delivery (see below).
2. XML with voter authentication and voter keys: files to load the voter identities with protected passwords and voter keys to the voting back-office. This data is used to authenticate the voters and, later, to provide the voters with their voter keys (which are stored within PKCS#12 keystores protected with a derivation of the voter credentials). A PKCS#7 file is also generated to guarantee the authenticity of the credentials and it is used at the time of importing them to the Online Voting System database via the Voting back-office.

This information is directly uploaded to the corresponding modules using the output files. The credentials assignation, authorization, activation and delivery will depend of
the project and can vary in function of the project necessities.

3.1.2 Voting back-office

The voting back-office is the component of the system that is used to configure, manage and finalize the election. It is a web application that offers a web interface to the administrators (see image below) and it is composed of the parts described in the next subsections. A REST-API with partial functionality is also being implemented. This component is only accessible through a private network (usually via VPN) that is only accessible by the system administrators.



Election Selection

Filter

Status:

Election Event

Please note that when using the filter, the pages may interact very slowly.

Election list (0 elections)

No filter applied

Election	Type of election	Institution	Status

You must create an Election Events before creating an Election.

Figure 3: Voting platform back-office

The voting back-office generates immutable logs for all critical operations, i.e. logs that are cryptographically protected against manipulation.

The following picture shows some inner details of the component:

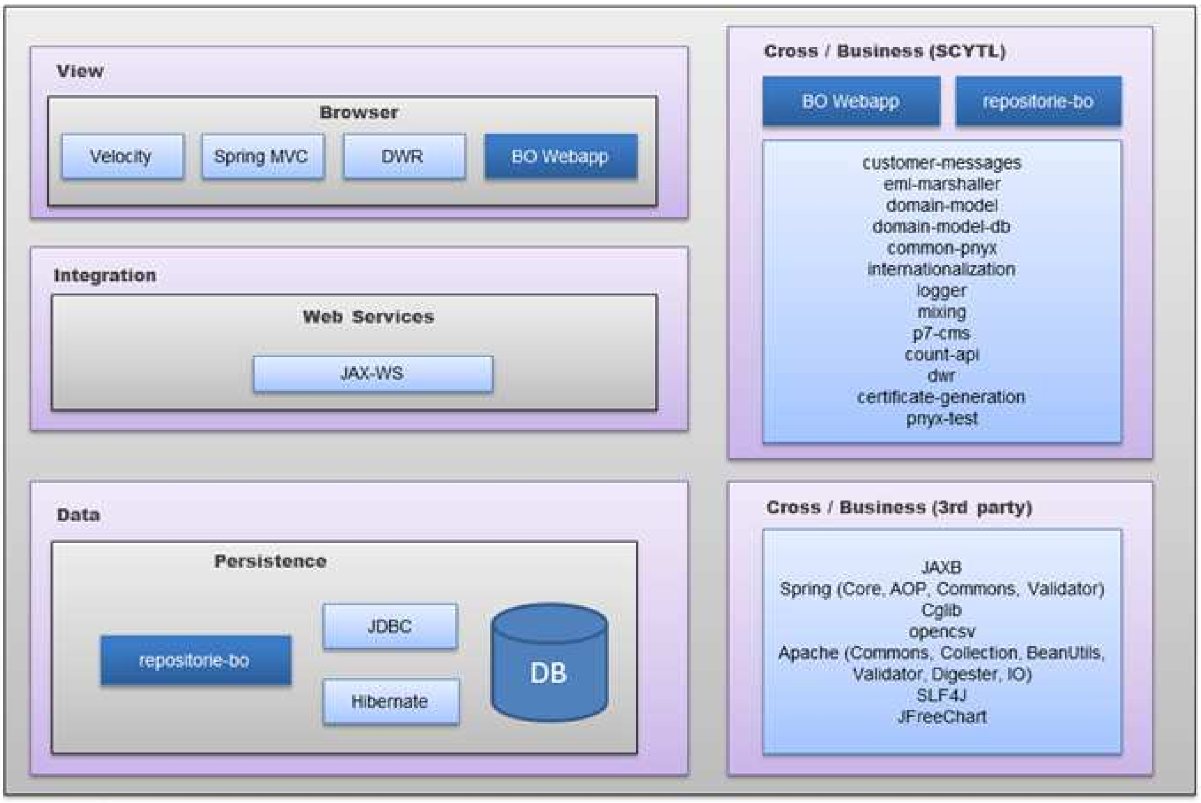


Figure 4: Voting platform back-office Architecture

The next diagram represents the functional location of the main modules of the Online Voting Platform BackOffice.

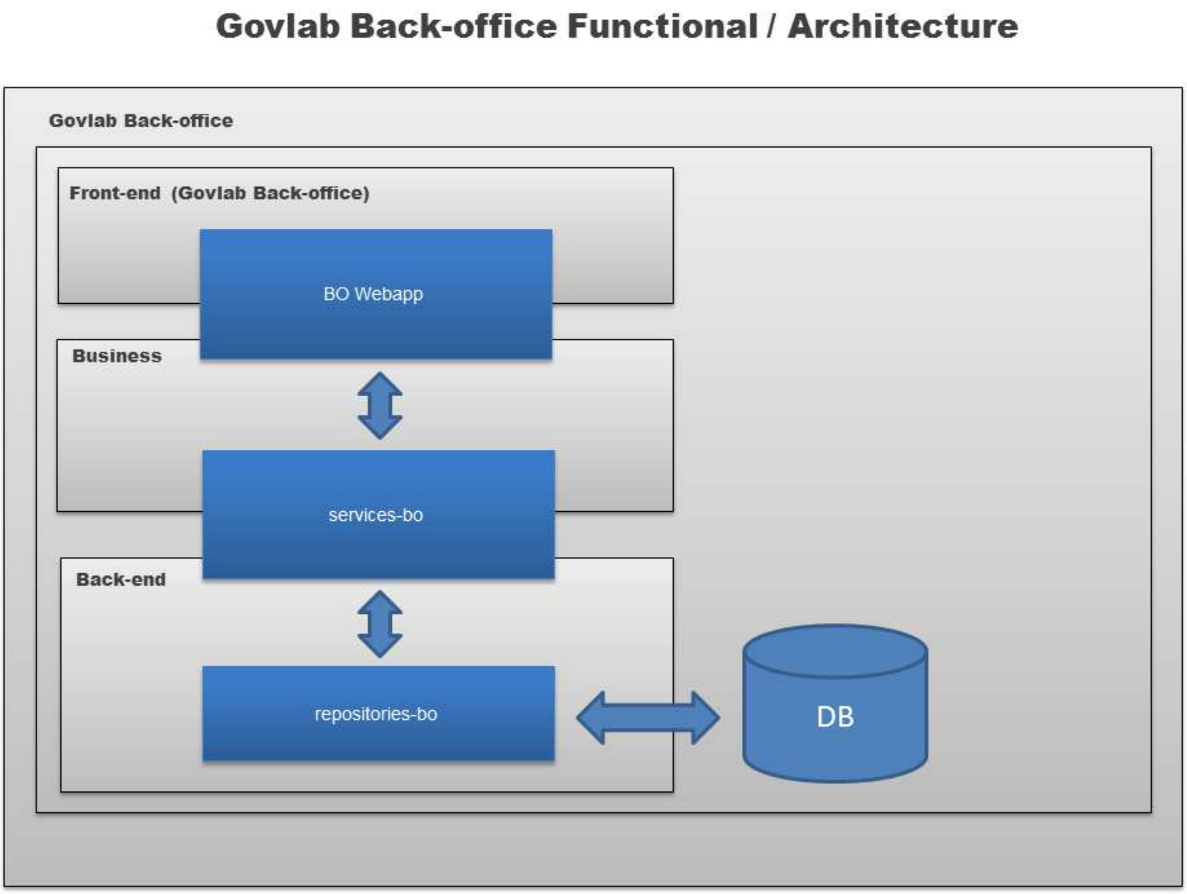

Govlab Back-office Functional / Architecture
Govlab Back-office
Front-end (Govlab Back-office)
BO Webapp
Business
services-bo
Back-end
repositories-bo
DB

Figure 5: Online voting platform Functional / Architecture

3.1.3 *Election Configuration*

The election configuration part allows the election administrators to configure and manage the election. The following actions should be performed to configure a typical election:

- Generation of institution
- Generation of Election Event
  - Configuration of Election Event
  - Generation of Electoral Board Key (used to encrypt the votes and distributed in shares in smartcards)
  - Generation of Administrator Board key (used to sign the election configuration and distributed in shares in smartcards)
- Configuration of electoral roll (import list of voter credentials)
- Generation of election
- Publish the election

The election configuration service makes use of a database to store and share the data with the Voting Portal and the Tally Service.

3.1.4 *Tally server*

The tally server is the part of the voting back-office that allows the election administrators to finalize the election by verifying, shuffling, decrypting and tallying the votes and publishing the election results (see figure below). The main steps are:

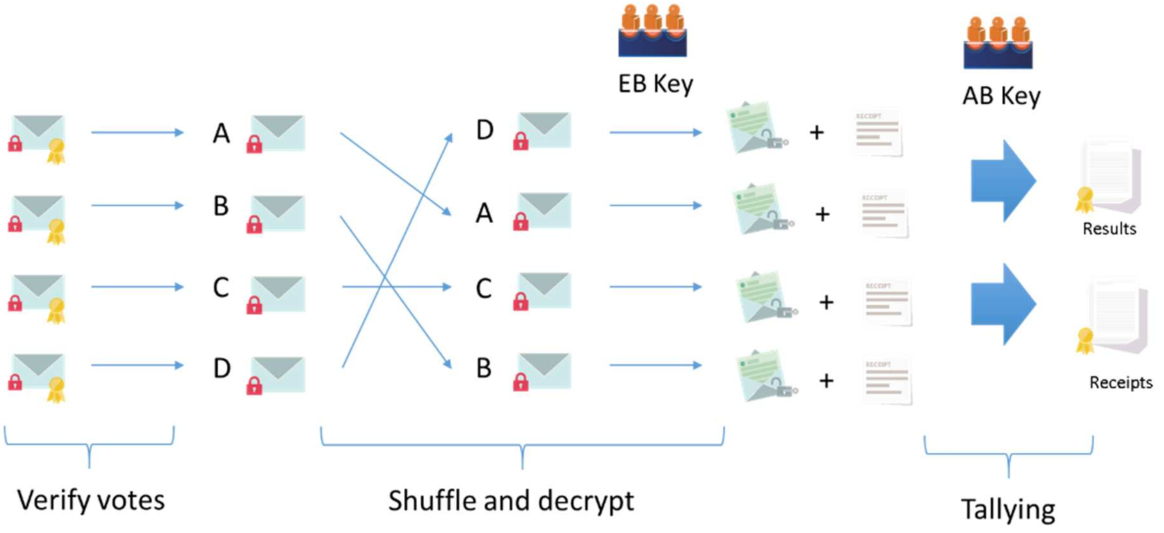


Figure 6: Tally server steps

- Verify votes: the signatures of the votes are verifying and it is checked that the voters that issued them are present in the electoral roll. Afterwards, the vote signatures are removed to separate the vote contents from the voter identity.
- Shuffle votes: the votes are shuffled to prevent that decrypted votes could be related to voter identities.
- Decrypt votes and receipts: votes and receipts are decrypted. Despite not represented on the picture for simplicity, vote and receipts are separately shuffled again.
- Tally votes: decrypted votes are counted in order to compute the election results.
- Publish results and receipts: results and receipts are published in a website, thus anybody can check them and voters can be sure their votes were processed.

3.1.5 *Voting Portal*

The voting portal is the component that, together with the Voting Client component, allows voters to cast a vote during the election. It also provides authentication facilities if no third-party authentication service is used. This component is connected to a network accessible to voters (usually Internet).

3.1.6 *Voting Portal backend*

The component is implemented as a web application and offers a REST-API. The following operations are supported:

- Authentication: Used by the voter to authenticate to demonstrate its identity. A JSON web token and an authentication token are returned in exchange.
- Provision of voter keys: Used by the voter to request the keystore that contains her set of voter keys.
- Cast a vote: Used to cast a signed and encrypted vote. Once received, it is checked that the voter is allowed to cast a vote and the vote is stored in the ballot box. A signature of the hash of the vote receipt is returned as a proof that the vote has been registered by the system.

3.1.7 *Voting Client FE*

The voting client is developed as a set of HTML and JavaScript files which run on the client side and more specifically in the voter's web browser. The voting client implements most of the cryptographic operations performed to protect the vote, achieving the end-to-end encryption previously mentioned. This component presents a graphical interface to the voter and interacts with the Voting Portal through the REST API. The most relevant operations performed are:

- Authentication: Requests the voter credentials and performs the corresponding derivations in order to authenticate the voter in front of the Voting Portal. In case of using an Identity Provider, it redirects the browser to this service.
- Vote encoding: Encodes the voter selections in a ballot.
- Vote receipt generation: Randomly generates a voting receipt that is delivered to the voter if the vote cast is successful.
- Vote encryption and signature: Encrypts the vote and its receipt with the election key and signs the encrypted vote with the voter key. Afterwards, the vote is sent to the Voter Portal.
- Receipt signature validation: Once the vote is cast, a signature of the hash of the receipt is returned, and this is validated before delivering the receipt and signature to the voter.

3.2 Scenarios of application

The current scenario of application is the voting system with the back office, database and voting portal components deployed. To be more specific, Netscaler and the external EWIS HoneyPot protect the REST HTTP requests that the voters issue to the Voting Portal backend. Then, XL-SIEM monitors the infrastructure logs (i.e. syslogs) of all the components deployed. And the internal EWIS HoneyPot is used to detect potential attacks at internal level.

3.3 Cybersecurity threats and impact

The list of potential threats has not been modified since its definition in the document D2.1 “SMESEC security characteristics description, security and market analysis report”, section 2.3.3 “E-voting Pilot Potential Attackers and Threats”.

3.4 Cybersecurity training and awareness status

At Scytl there is a Security department, lead by the Director of Security and Data Protection Officer, and with 4 security analyst and researchers.

The Scytl Security department is reporting to the VP of Research & Development. Security is always a core priority in Scytl operations and products; by this way, not only the Security department is performing Security functions. The Security department is defining and leading the security activities, which are also coordinated with other departments, mainly the IT Department and the Testing Department.

The Security department is the core of the Security activities performed at Scytl, which can be executed by the Security Department itself, the IT Department - following the security policies defined by the Security department, or the Testing Department - executing the security tests defined by the Security department.

3.4.1 Security strategic areas

The functions performed by the Security department are the following:

3.4.1.1 Definition of Security Policies

- General Security Policies

Scytl Security Department defines all the Security policies affecting Scytl internal operations. These policies cover access control management, network security, IT rules for end-users, personnel security, and change management among others.

- Secure Software Development Lifecycle

Scytl Security Department defines and supports the implementation of a Secure Software Development Lifecycle, affecting the Development department, Testing department, Consulting, and Security department itself.

- Physical Security

Security department is also responsible of physical security, including access control to Scytl offices, restricted areas, and intrusion detection systems (alarms).

3.4.1.2 Security training and awareness

The Security department is in charge to promote and organize security trainings when the training necessity is identified.

They are also responsible to make Scytl personnel and partners aware of most recent security threats, attacks, and any security tendency which could be useful for Scytl activities.

3.4.1.3 BCP and DRP maintenance

The security department is in charge of developing the BCP and DRP of Scytl facilities and datacenters. They are also responsible to:

- Test the plans periodically, as required in the plans.
- Update the plans in case any change occurs.
- Provide to election software project managers the BCP and DRP templates for election projects.

Furthermore, the Security Manager is the replacement of the VP of Research & Development as the responsible of the Crisis response team in the Business Continuity Plan and the Disaster Recovery Plan.

3.4.1.4 Risks assessment

When a risk situation is detected (new threats affecting Scytl, potential security attacks, security control opportunities, organization or infrastructure changes affecting security) the Security department is in charge to perform a risk assessment regarding this risk situation.

This risk assessment could be formal or informal according to the importance or complexity of the situation, and its objective is to inform to the VP of Research & Development about the potential mitigation controls and residual risks, to take a decision.

3.4.1.5 Incident management

Any security incident detected by IT personnel or any other people reporting a security incident is notified directly to the Security Manager.

The Security Manager is in charge to coordinate the required activities to solve the incident.

3.4.1.6 Security Research

The Security Department is always up-to-date about security tendencies and new security attacks, while they are researching new security controls which could be applied to Scytl environment and products.

Scytl Security Department is often speaking in prestigious conferences and congresses about electronic voting, security, and cryptography. They have published several articles in scientific magazines and registered several patents related to Scytl technology.

3.4.1.7 Security and Cryptography consultants

Scytl security department is in charge to define the security requirements and cryptographic protocols related to the Scytl election software; by this way, the Security department is acting as a consultant defining the requirements which shall be implemented by the software developers.

Furthermore, Scytl Security department is also involved in any security decision affecting election projects.

3.4.1.8 Privacy managers

Scytl Security manager is also Scytl Privacy manager. In addition to considering “privacy” as a Security dimension in any Security activity previously described, he is acting as a Privacy officer by guaranteeing the compliance of the privacy laws (GDPR, Spanish Data Privacy laws and any other international privacy law related to subsidiaries and election projects).

Security functions of IT Department: Scytl IT Department is in charge of **Technical Security** in two different situations: Scytl internal Security and technical security for election projects.

3.4.1.9 Scytl Internal Security

IT Department is in charge of applying the Security Policies defined by the Security Department.

These policies are covering technical issues like:

- Antivirus and antimalware policy.
- Firewall and communications filtering.
- Network segmentation and permissions.
- Network passwords policy.
- Application passwords policy.
- Account management.

- Hardening of users’ laptops and PCs.
- …

3.4.1.10 Technical Security for election projects

In addition to the IT functions required for the IT department in any election project (technological architecture, installation and configuration, network set up…) the IT department is in charge to apply the technical requirements related to security which are established by the Security Department.

These security technical requirements are covering issues like:

- Hardware security hardening (if required), e.g. voting kiosks.
- Software hardening, following the hardening guides provided by the Security Department, involving the operating system, web servers, application servers, database engines, and any other basic software required for the use of the election software.
- Logs monitoring, after applying the traceability configuration required by the Security Department, IT Department is monitoring the logs and raising alerts to the Security Department when required.
- Network security enhancement.

3.4.2 Security functions of Testing Department

Scytl Testing Department is integrated by testing engineers performing **Security testing**.

In any software project, there are 3 types of Security test according to its schedule:

- **Continuous security tests**: They are executed daily through automated tools to ensure the software is not containing bugs, programming defects, and it is resistant to most known security attacks (like the specified in OWASP top ten, CWE/SANS 25, GHDB…). Continuous integration tools related to bugs, style, code analyzers, and web security scanners are used.
  In addition to the automated tests, our Secure Software Development Lifecycle includes manual secure testing performed by the Security testing engineers.

- **Security tests on releases and deliveries**: Before releasing a new version of the product, and before the delivery a service, a full set of Security tests is performed, including: 1) automated security test covering the application security 2) using web security scanners, automated security test covering the infrastructure security 3) using vulnerability scanners, and manual security tests 4) using web proxy applications to perform spidering, request tampering, request sequences and repetitions,…

- **External Security tests**: Full security audits are performed by independent third parties, including black box or white box approaches, Analysing the application and its source code, required and managed internally or requested by customers.

3.5 Business opportunity

Scytl is the worldwide leader in secure electronic voting, election management and election modernization solutions. Its solutions incorporate unique cryptographic protocols that ensure maximum security, transparency and auditability in all types of elections. Scytl's groundbreaking electoral security technology is protected by international patents and it enables organizations to electronically carry out all types of electoral processes in a completely secure and auditable manner, positioning the company as the global leader in this industry.

Within SMESEC, Scytl will be able to update its security solutions with more efficient mechanisms. The proposed real-life experimentations will evaluate the SMESEC framework for the e-voting use case. The identified most cost-effective cyber-security mechanisms will be integrated on the commercial offer of Scytl to provide more functionality and lines of protection for Scytl's clients.

SMESEC will provide the security layer for hardening, monitoring, attack detection and prevention as well as a method to ensure the availability of the election process. The integration of both technologies will provide a joint solution that will allow entities with limited budget to implement secure online voting processes with the highest levels of security, availability and transparency. Moreover, SMESEC will address the requirement for last minute code and service modifications to meet the peculiarities of each specific voting process.

4 Integration of the SMESEC Framework

4.1 SMESEC-enhanced business pilot

The electronic voting system integrated with the SMESEC Framework has been deployed in an environment composed of several networks with different security privileges and policies. The different components and setup can be seen in the following picture:

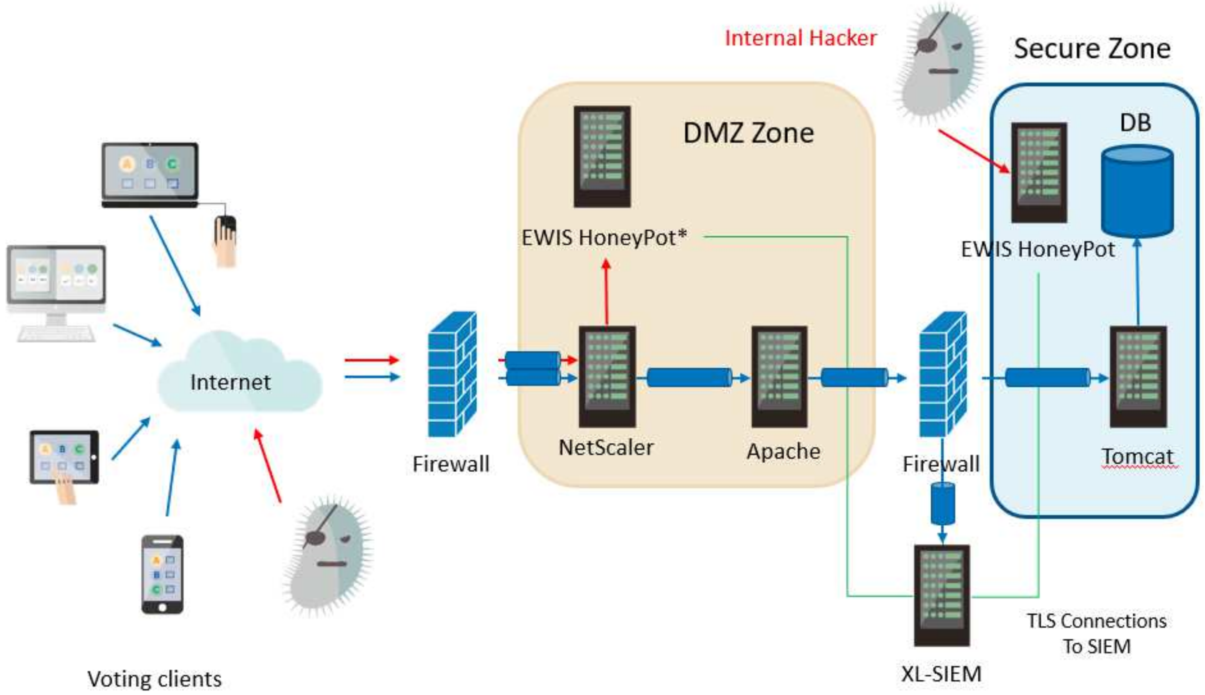


Figure 7: Online voting pilot components and setup

The voting system is composed of three main components: the web server (Apache), the application server (Tomcat) and the database (DB). The web server is deployed in the DMZ network, which is accessible through Internet. The application server and the database are deployed in the Secure Zone network, which is not directly accessible through Internet. In addition, when the voters connect to the system, a Javascript Voting Client is locally executed in their computers.

The integrated SMESEC Framework components are the NetScaler, the XL-SIEM and the different instances of the EWIS HoneyPot. NetScaler is used as an application firewall, thus it is configured to be the first element that process the incoming connections that arrive from the Javascript Voting Clients in Internet to the web server. The EWIS HoneyPot that is installed in the DMZ Zone is used as a system to receive redirected connections rejected by NetScaler, i.e. connections that NetScaler have determined that are not compliant with the voting REST API. The EWIS HoneyPot that is installed in the Secure Zone is a regular honeypot system used to attract attackers that are trespassing into this private network. And, finally, the XL-SIEM is an agent, deployed in a dedicated subnet, that listens for Syslog connections from the other components deployed, e.g. web server, web application server, database, Netscaler, etc. The syslog of these components is forwarded to this agent that, in turn, forwards it to the external XL-SIEM server.

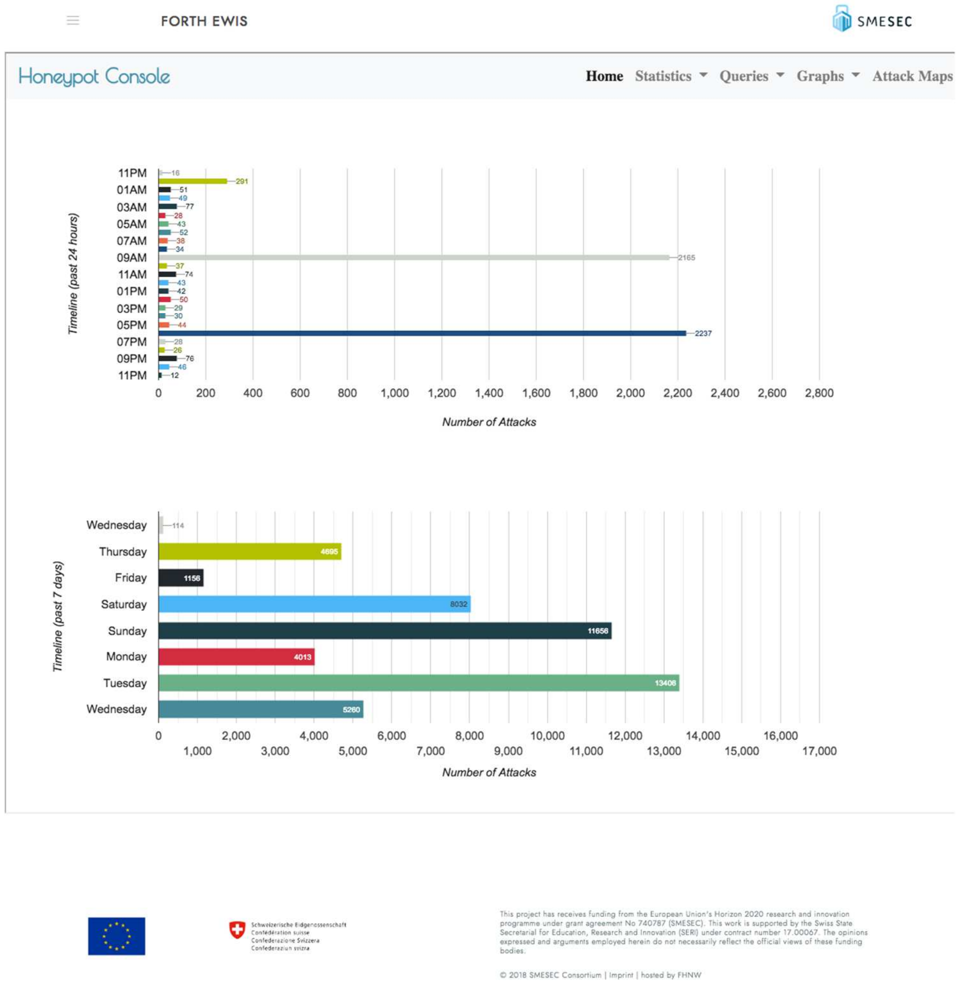
The components shown in the picture have been deployed in the EC2 of Amazon Web Services, although the same scenario is valid to be deployed in physical networks. From a perspective of the SMESEC Framework, the components selected can be used both in virtual and physical environments.

4.2 Process and tools integrated

The tools integrated in this use case were NetScaler, EWIS HoneyPot and XL-SIEM:

- NetScaler VPX is a virtualized node deployed in premises, which provides advanced network data manipulation features and capabilities. This tool offers several functionalities related to the filtering of network data. In our case it is used as a Web Application Firewall (WAF) to filter the HTTP REST connections issued by the Javascript Voting Client. The deployed node analyses the content and the format of inbound connections and accepts or rejects them. NetScaler was integrated in our system by deploying a virtual machine instance in AWS that contained the software. This instance was configured to have three network interfaces in three different subnets: client, server and management. The tool was configured and managed through the management network interface. Then, all the traffic to be filtered arrives through the client interface and is output through the server interface, which is located inside the same subnet of the web server.
- EWIS Honeypot: The honeypot is a software used to receive and register attacks. It works as an Early Warning Intrusion Detection System, luring attackers into attacking it instead of the production system. In that way, we are able to gain insight about the ongoing attacks in the network and provide valuable information to the system administrator. Currently, there are two instances of the EWIS Honeypot that have been deployed, one in the FORTH premises and another in the Secure Zone area of our scenario. The first instance, deployed in FORTH is connected to the NetScaler instance that is deployed in our premises. When an external connection arrives, it is firstly examined by NetScaler, then, based on ruleset created for Scytl, if it is not a valid request it is forwarded to the honeypot, for further examination and interaction with the potential attacker. If it is a valid request it is forwarded to the e-voting production system. The setup of this behaviour was directly configured by CITRIX and FORTH. The honeypot that is deployed in the Secure Zone, is a honeypot that offers a set of services and tries to attract attackers to explore them. Both honeypots were deployed using a virtual machine provided by FORTH and directly configured by them. The honeypots

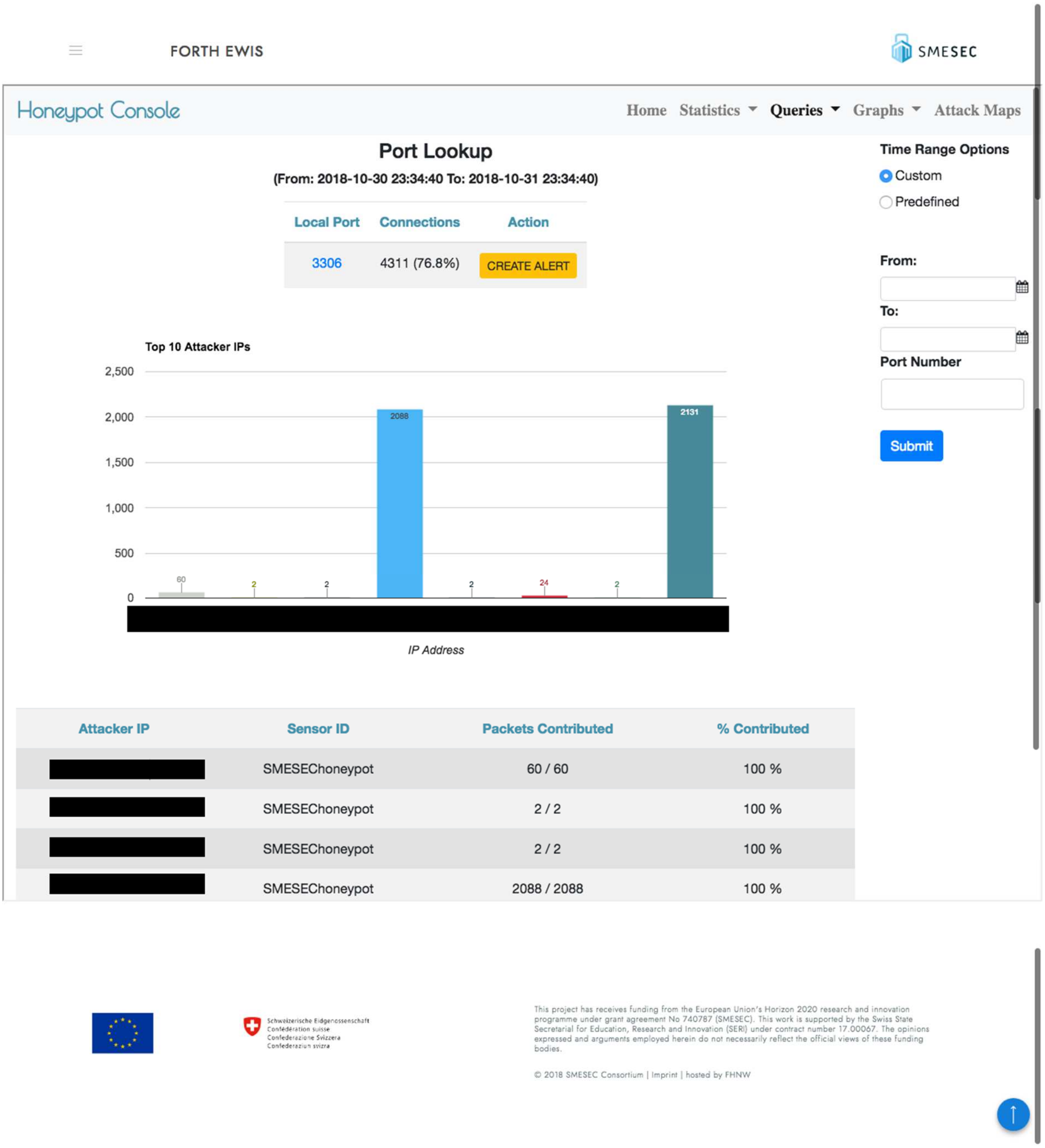
included in the solution are able to emulate a variety of services, which are common targets for cyberattacks. So, we are able to detect attacks against FTP, TFTP, HTTP, HTTPS, TELNET, DNS, SMTP, MS Windows RPC, SMB, SSH, DNS, NTP, SNMP, NetBIOS and more. The events detected by the honeypots are sent to the ATOS cyber-agent in the Scytl premises and to FORTH's databases. Moreover, the data collected to FORTH's events' databases are visualized by the EWIS Visualization platform. In that way, the system administrator can receive more detailed reports for each event captured, focusing on specific services, IP or date ranges. General statistics, can also be used, for assessing the overall cybersecurity state of the organization. Finally, EWIS has already been unified and incorporated in the SMESEC Framework and can be accessed in a unified way by all the users. In **Error! Reference source not found.**, we can see a view with general attack statistics for the last 24h and the last 7 days. This view is taken from the incorporated SMESEC framework. In Figure 9 information about the top10 IP addresses that target MySQL service can be seen.



This project has receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 740787 (SMESEC). This work is supported by the Swiss State Secretarial 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 these funding bodies.

© 2018 SMESEC Consortium | Imprint | hosted by FHNW

Figure 8 EWIS general attack statistics, as seen in SMESEC platform



[redacted]

Port Lookup

(From: 2018-10-30 23:34:40 To: 2018-10-31 23:34:40)

Local Port	Connections	Action
3306	4311 (76.8%)	CREATE ALERT

Time Range Options

● Custom

○ Predefined

From:

To:

Port Number

Submit

Attacker IP	Sensor ID	Packets Contributed	% Contributed
[redacted]	SMESEChoneypot	60 / 60	100 %
[redacted]	SMESEChoneypot	2 / 2	100 %
[redacted]	SMESEChoneypot	2 / 2	100 %
[redacted]	SMESEChoneypot	2088 / 2088	100 %

This project has receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 740787 (SMESEC). This work is supported by the Swiss State Secretarial 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 these funding bodies.

© 2018 SMESEC Consortium | Imprint | hosted by FHNW

Figure 9 EWIS specific statics for a specific service (MySQL) based on the tcp port

- XL-SIEM: This is a tool that collects events produced in the system and allows the study of them and also the generation of alarms based on the data collected. This tool is composed of a server that receives and process the data and also of agents that collect the data in place. In our case, we just have installed one agent in a dedicated subnet that collects the syslog of the different components of our scenario. The tool was provided as an installation package for Ubuntu/Debian. We deployed an instance of a Debian 9 in AWS and we installed the package with the instructions provided. Then, each server was configured to send its syslog data to the agent.

Integration Process

As already mentioned, NetScaler is an application delivery controller that provides flexible delivery services for traditional, containerized and microservice applications from your data center or any cloud. It features unmatched security, superior L4-7 load balancing, reliable GSLB, and increased uptime. In addition, it facilitates cost control by efficient capacity scaling capabilities, even allowing capacity sharing across instances on-premises and in the cloud. NetScaler VPX can be deployed on both public and private cloud infrastructures. Since Scytl Voting Server was delivered through the Amazon Web Services (AWS) ecosystem, we opted for deploying a co-located instance of NetScaler VPX and configure it accordingly. Upon its configuration, NetScaler would provide dedicated security services to the associated Scytl Voting server, through its Secure Web Gateway service functionality.

The actual NetScaler deployment on AWS is a well-documented process, analysed in full detail in D3.2. In addition, D3.2 also provides a step by step guide for configuring Citrix Secure Web Gateway service. For additional information regarding these processes, readers are encouraged to refer to the aforementioned sources.

The Citrix NetScaler appliance is available as an Amazon Machine Image (AMI) in AWS marketplace and enables customers to leverage AWS Cloud computing capabilities and use Citrix NetScaler load balancing and traffic management features for their business needs. The virtualized flavor of NetScaler, namely NetScaler VPX instance, supports all the traffic management features of a physical appliance, and can be deployed as standalone instances or in HA pairs.

For the specific use case, we opted for deploying a standalone instance in Scytl's predefined Virtual Private Cloud (VPC) domain, located inside a specific AWS Region. NetScaler was deployed in a special availability zone as illustrated in Figure 10

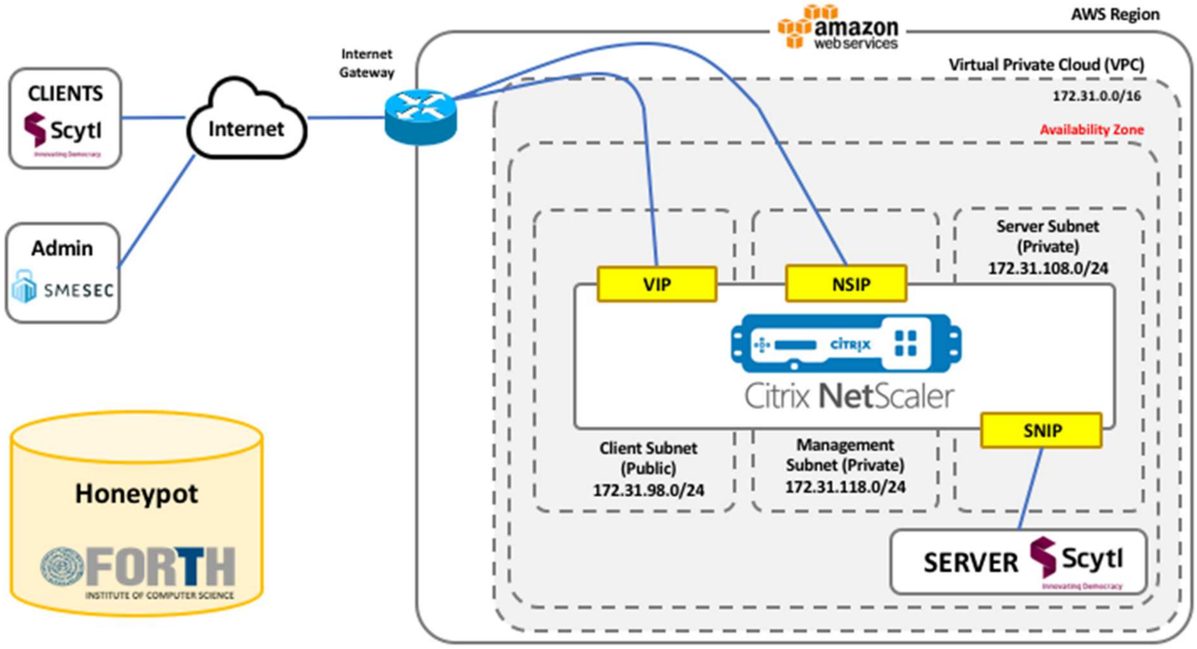


Figure 10:Initial Node Deployment

For properly deploying NetScaler and enabling the full spectrum of its features, three distinct network interfaces must be available. These interfaces are linked to different subnets to avoid data leakage and each plays a different role in the overall NetScaler functionality. More specific, each NetScaler instance requires at least three IP subnets:

- A management subnet
- A client-facing subnet (VIP)
- A back-end facing subnet (SNIP, MIP, etc.)

Citrix recommends three network interfaces for a standard NetScaler instance on AWS installation and this is the approach we also followed for the specific use case. The Management subnet is a private subnet associated with the NSIP interface and is only used for providing administrative access to the deployed NetScaler node. For executing configuration commands users must obtain access to the NetScaler's Command Line Interface (CLI) through the NSIP and the Management subnet. This dictates certain connectivity of the NSIP to the AWS Internet Gateway preferably via hardcoded routes. This configuration was also conducted in the specific setup, together with additional networking configuration, as illustrated in Figure 11.

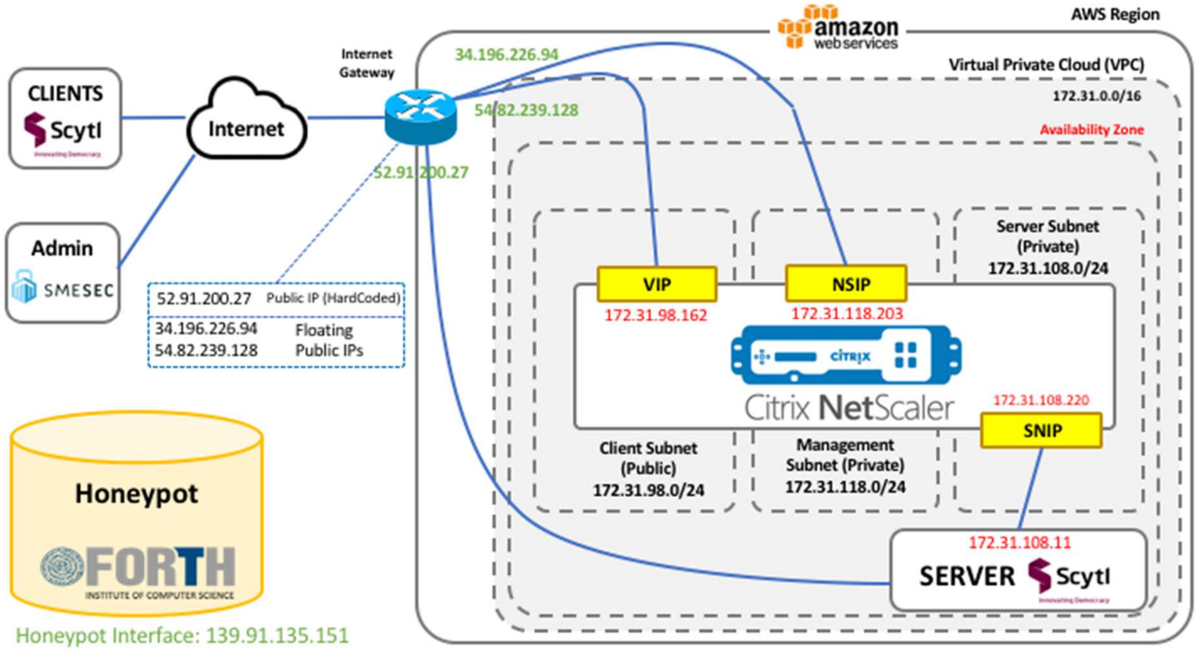


Figure 11: Initial IP/Networking Configuration

However, after the configuration of NetScaler in AWS, and despite establishing proper connectivity with Scytl Voting Server via the dedicated SNIP interface, no traffic traversed the node. The reason was an existing direct line of communication between the AWS Internet Gateway and the Scytl Server which virtually led traffic to bypass NetScaler and all its security features. Once discovered, this problem was tackled and the final networking configuration is presented in Figure 12 .

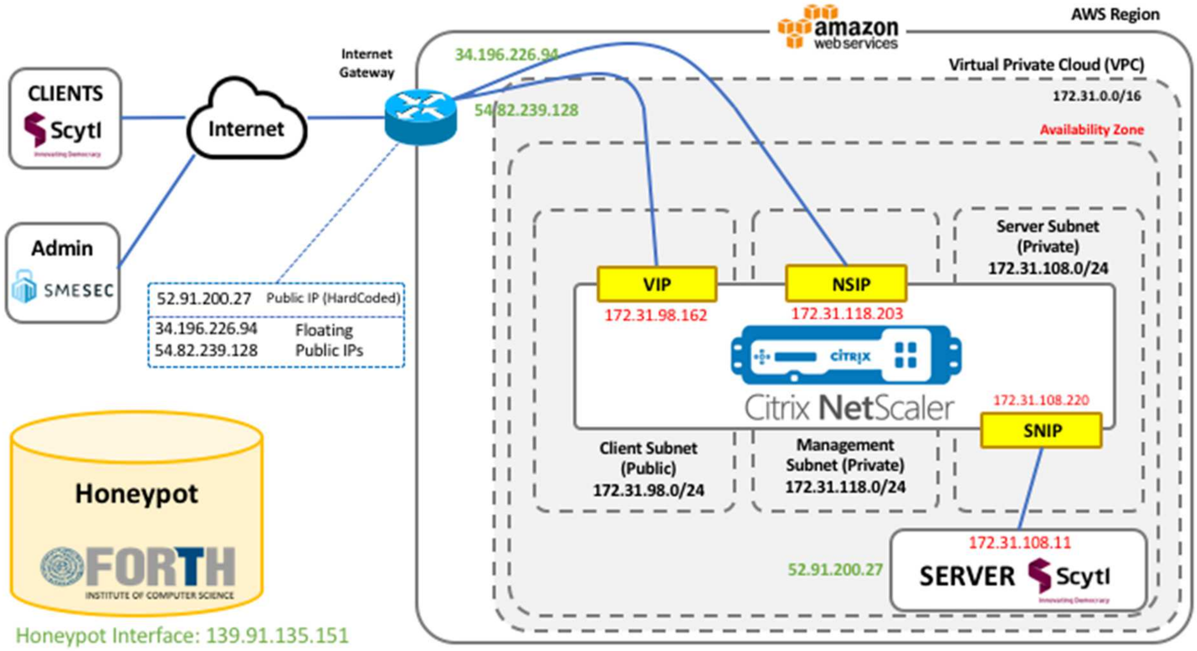


Figure 12: Current IP/Networking Configuration

Figure 6 illustrates the management route used for accessing NetScaler CLI. The SMESEC administrator connects to the AWS Internet Gateway using a .pem file for enhanced security. Once obtain access to the AWS Region, the Internet Gateway routes the management requests to the NetScaler NSIP. This dictates the existence of a dedicated public IP associated with the management subnet of the NetScaler, therefore an additional one must also be allocated for traffic purposes.

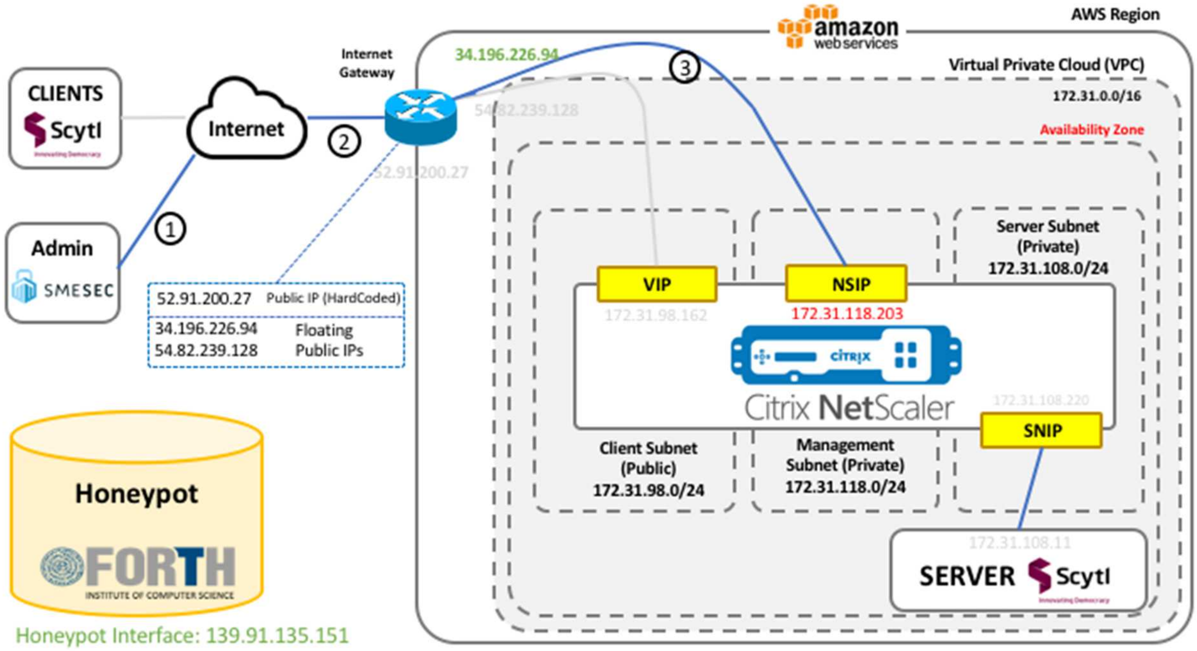


Figure 13: Management Packet Flow

Figure 14 presents the traffic route from Scytl clients towards the Scytl Voting server which traverses NetScaler. As already stated, a dedicated public IP must be allocated for granting access to the AWS region infrastructure in which the Scytl VPC is deployed.

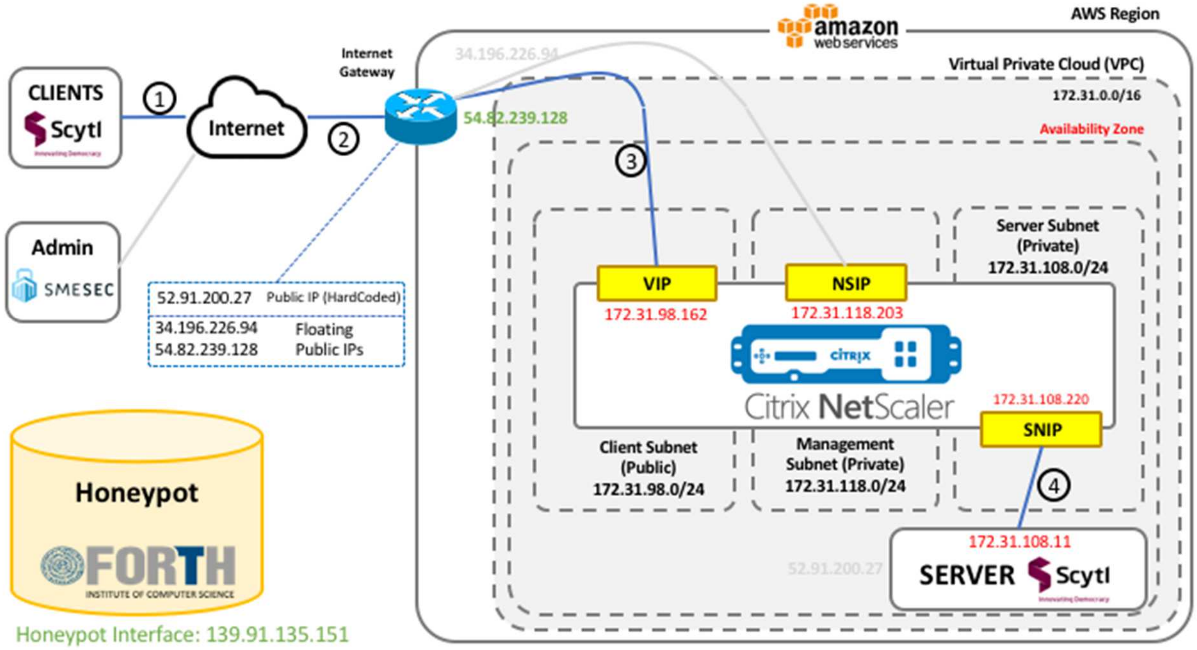


Figure 14: Client Request Packet Flow

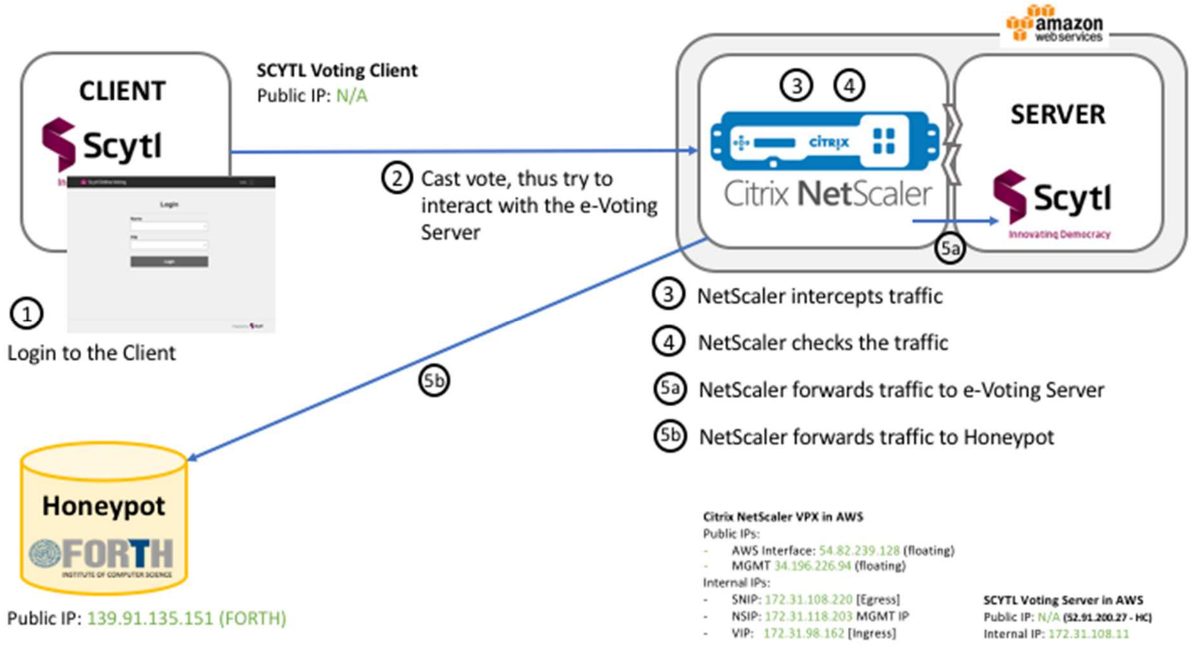

CLIENT
Scytl
SCYTL Voting Client
Public IP: N/A
② Cast vote, thus try to interact with the e-Voting Server
① Login to the Client
amazon web services
③ ④
Citrix NetScaler
SERVER
Scytl
Innovating Democracy
5a
5b
③ NetScaler intercepts traffic
④ NetScaler checks the traffic
5a NetScaler forwards traffic to e-Voting Server
5b NetScaler forwards traffic to Honeypot
Honeypot
FORTH
INSTITUTE OF COMPUTER SCIENCE
Public IP: 139.91.135.151 (FORTH)
Citrix NetScaler VPX in AWS
Public IPs:
- AWS Interface: 54.82.239.128 (floating)
- MGMT 34.196.226.94 (floating)
Internal IPs:
- SNIP: 172.31.108.220 [Egress]
- NSIP: 172.31.118.203 MGMT IP
- VIP: 172.31.98.162 [Ingress]
SCYTL Voting Server in AWS
Public IP: N/A (52.91.200.27 - HC)
Internal IP: 172.31.108.11

Figure 15: Demonstration Packet Flow

4.3 Testing

In order to test the system deployed, we have applied functional tests and security automated tests.

The functional tests consist of configuring an election and test that it is possible to vote and perform the counting process normally.

The security automated tests consist of using an internal tool, ROTI, which simulates different parts of the voting process and tries to execute them automatically. The tool calls and modifies the logics of the Voting Client API in order to perform several security tests against the Voting Portal. These tests try to exploit common attacks to ensure the system cannot be compromised. Some of these attacks are also used to test NetScaler:

C655537 1.7.1 - Send a malformed request: Authentication request

C655538 1.7.2 - Send a malformed request: Certificates request

C655539 1.7.3 - Send a malformed request: Ballot request

C655980 1.7.4 - Send a malformed request: Web Service request

These tests issue an HTTP request to perform one of the actions described by their title with an error in the parameters of the request. Netscaler has to verify the HTTP requests received are compliant with the Voting Client API, thus in these tests it has to detect the modifications and discard the HTTP requests.

5 Next steps

5.1 Integration of business in the SMESEC Framework

The SMESEC Framework provides security at infrastructure level. Thus, its integration is based on adding and configuring certain components when the infrastructure is deployed. The framework management is performed via a web interface that is completely independent of the voting system backoffice. The integration will basically modify the installation procedures of the system, but not the software of the system.

5.2 Training and awareness plan

A plan for training and awareness on the new SMESEC framework must be set-up. The Security Committee will approve a periodic plan of communication and training on security which will include the SMESEC framework use.

- Basic Training: A program for training and continuous awareness of issues related to security of information systems is defined. This training will include the use of new security framework which, due to its usability, is intended to be easy to use with little training.
- Advanced Training: Specific technical training will be given to the staff of the division of information systems and Security department that will integrate SMESEC framework in the online voting platform deployed for real elections

5.3 Initial testing and validation plan

After the final integration of the second prototype is concluded, we will perform a pilot test to validate the usefulness and effectiveness of SMESEC framework in the e-voting use case.

The tests will be conducted at Scytl's premises. An election will be configured so that a set of end users will be asked to access Scytl online voting platform and vote. All users will be able to conduct their tasks from any location. The SMESEC framework will be configured as well. During the election, several tests will be performed in order to verify that the integrity of the system cannot be compromised and that the SMESEC framework effectively protects the system.

After the conclusion of the election, the team will evaluate the results and elaborate the conclusions regarding the functioning of the framework, in terms of usability and effectiveness.

In terms of infrastructure, the pilot will be run on the prototype version deployed in the Amazon EC2 cloud infrastructure.

The different phases of the pilot will be the following:

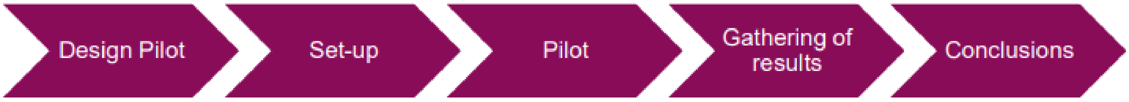


Figure 16: Pilot different phases

a. Phase 1: Design Pilot Test Plan. In this phase, the initial design and conceptualization for the test cases is carried out and validated among all participants in the project consortium.
b. Phase 2: Pilot Set-up. In the second phase, the necessary facilities (both physical and virtual) will be prepared to run the tests.
c. Phase 3: Pilot. During the third phase, the pilot itself will be run.
d. Phase 4: Gathering and analysis of results. In this phase, all results from individual tests will be gathered and consolidated.
e. Phase 5: Conclusions and next steps. In this phase, a workshop will be run to share results and feedback from individual tests and create a prioritized list of recommendations for future development tasks.

6 Conclusions

6.1 Feedback and experience of the initial integration

Most of the tools were deployed as instances of virtual machines, except the XL-SIEM. The installation was not complicated, although some issues were presented:

- Netscaler: 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 rules of the WAF. No written instructions were provided at the moment of integrating the tool, thought we would have preferred it to avoid depending on them for the installation. Such instructions are included in the Appendix of D3.4
- EWIS HoneyPot: We only installed the internal honeypot. In this case, we had to deploy a virtual machine provided by FORTH and all the configuration was performed by them. We had to give access to the machine. Again, we would have preferred to have a manual with the steps required to perform the configuration, otherwise a new deployment depends on them.
- XL-SIEM: The installation of XL-SIEM was straightforward, in this case we had to create an instance of a Debian image and install the packages provided. An instructions manual was provided, thus it would be easy to do a re-installation.

6.2 Fulfilling of objectives

Overall, the results of the integration activities conducted to obtain the first prototype of the e-Voting use case are satisfactory and meet the initial expectations, having obtained the expected results. All the tools that were planned to be integrated in this first iteration have been successfully integrated.

The security framework has not yet been integrated into the e-voting platform, since the first prototype was expected to be ready in the 18th month of the project and, therefore, the system does not yet have the expected usability. This is expected to be solved in the second version of the prototype, in the 25th month of the project. It is also expected that, at that moment, the security framework will integrate all available tools that are into the scope of the project.

6.3 Use in SME environment

SMESEC framework is intended to be easy to use with little training. However, in this first prototype, the integration has not been done to the framework, but to the different tools: the Honeypots, the

Netscaler and the XL-SIEM. For this reason, at the moment both the configuration and the use of these tools can only be done by experts in programming and cybersecurity.

It is expected that this will change in the second prototype, when the different tools are fully integrated into the security framework.

6.4 Improvements for the scenario

The scenario will be improved in the next iteration by adding two components of the voting service: Credential Delivery and the Receipts Website.

6.4.1 Credential delivery

This is a service used to deliver the generated voter credentials to the voters in a secure manner (e.g. via e-mail using a One Time Link (OTS)). The service is implemented through the following components:

- Credential Delivery: it is a web application which offers a REST-API to manage and deliver the credentials.
- Credential Delivery back-office FE: it is a set of JavaScript files that allow a manager to configure and election and import the credentials generated by the Credential Generator module. The file with the credentials is encrypted with a secret key shared between the Credential Generator and the Credential Delivery.
- Credential Delivery Portal FE: it is a set of JavaScript files that allow a user to retrieve their credentials pointed by the OTS file received.

6.4.2 Receipt Website

This is a website where the receipts generated during the election are published. Thus, the voters can check their vote was decrypted or present in the ballot box. The service is enabled at the end of the election.

References

[1] **Deliverable:** SMESEC. *D.2.1 – SME security characteristics description, security and market analysis report.* Oikonomou, George. 2017

[2] **Deliverable:** SMESEC. *D.3.2.– SMESEC Unified Architecture.* Copty, Fady. 2018

Annexes

Enter your text here.