D3.3 Vulnerability Assessment as a Service v1

WP3 – Cyber security risk assessment & Beyond – Sphinx Intelligence

Version: 1.00

SPHINX

A Universal Cyber Security Toolkit for Health-Care Industry

Disclaimer

Copyright message

Document information

Grant Agreement Number	826183	Acronym	SPHINX
Full Title	A Universal Cyber Security Toolkit for Health-Care Industry		
Topic	SU-TDS-02-2018 Toolkit for assessing and reducing cyber risks in hospitals and care centres to protect privacy/data/infrastructures		
Funding scheme	RIA - Research and Innovation action		
Start Date	1st January 2019	**Duration**	36 months
Project URL	http://sphinx-project.eu/		
EU Project Officer	Reza RAZAVI (CNECT/H/03)		
Project Coordinator	National Technical University of Athens - NTUA		
Deliverable	D3.3 - Vulnerability Assessment as a Service v1		
Work Package	WP3 – Cyber security risk assessment & Beyond – Sphinx Intelligence		
Date of Delivery	**Contractual** M18	**Actual**	M18
Nature	R - Report	**Dissemination Level**	P - Public
Lead Beneficiary	HMU		
Responsible Author	Yannis Nikoloudakis	**Email**	nikoloudakis@pasiphae.eu
		Phone	-
Reviewer(s):	Dana Oniga(SIMAVI), Lous Landeiro Ribeiro(PDMFC)		
Keywords	Vulnerability Assessment		

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

Document History

Version	Issue Date	Stage	Changes	Contributor
0.10	21/5/20	Draft	ToC	Ioannis Kefaloukos (HMU)
0.20	28/5/20	Draft	Content	Yannis Nikoloudakis (HMU)
0.30	28/5/20	Draft	Content	Ioannis Kefaloukos (HMU)
0.40	28/5/20	Draft	Content	Yannis Nikoloudakis (HMU)
0.50	10/6/2020	Draft	Internal Review	Dana Oniga (SMAVI), Louis Landeiro Ribeiro (PDMFC)
0.60	12/6/2020	Draft	Address Reviewers' Comments	Yannis Nikoloudakis (HMU)
0.70	12/6/2020	Pre-Final	Address Reviewers' Comments	Yannis Nikoloudakis (HMU)
0.80	16/6/2020	Pre-Final	Quality Control	George Doukas (NTUA) Michael Kontoulis (NTUA)
1.00	16/6/2020	Final	Final	Christos Ntanos (NTUA)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

Executive Summary

This deliverable is a report on the progress of the development of the Vulnerability Assessment as a Service (VAaaS) component. VAaaS is one of the central components of SPHINX, which assesses the vulnerability status of network-enabled devices and services, within the SPHINX environment. The assessment outcome is distributed to several internal SPHINX components to aid them in performing their designated functionality. This report is a versioned document, and this is the first version (v1).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

Contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

Table of Tables

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

Table of Figures

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

Table of Abbreviations

VAaaS – Vulnerability Assessment as a Service

VAS – Vulnerability Assessment Service

CVSS – Common Vulnerability Scoring System

CVE – Common Vulnerabilities and Exposures

DSS – Decision Support System

RCRA – Real-time Cyber Risk Assessment

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

1 Introduction

1.1 Purpose & Scope

The scope of this report is to present the translation of all VAaaS-related requirements (System, User, Functional, Non-Functional), into functionalities and therefore sub-components. The SPHINX project, at the time of authorship of this document, is on the 18th month of its overall lifetime. The VAaaS component's development phase is at the end of its first iteration (M10-M18). The initial design phase has just finished, and the component is under heavy development. Thus, only some of the foreseen functionalities have been implemented. This report will demonstrate its currently developed functionalities and overall achievements, in terms of development.

1.2 Structure of the deliverable

The rest of this deliverable is structured as follows. Section 0 presents the component's role within the SPHINX ecosystem, its design principles, and its technical characteristics. Section 2 presents the component's basic functionalities and their particularities. Finally, Section 3 concludes this report by presenting a short summary and general conclusion deriving from this report.

1.3 Relation to other WPs & Tasks

This report is closely related to WP2 and more specifically to tasks T2.2 (Basis of Legal and Ethical Requirements), T2.3 (Stakeholders' Requirements) and T2.5 (SPHINX Architecture and Detailed Technical Specifications). Additionally, this task (T3.2) is also related to WP6 and more specifically to T6.1 (Definition and specification of the system integration). WP2 has provided technical partners with the architectural and design guidelines for the development of their respective components. Similarly, WP6 has provided partners with the design and development/implementation principles.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

Overview of Vulnerability Assessment as a Service

1.4 Scope of Vulnerability Assessment as a Service

Within a large and complex network environment, such as healthcare infrastructures, hundreds, if not thousands heterogeneous network-enabled entities (devices and services) enter or leave the network. These entities, apart from largely widening the management surface of the underlying infrastructure, also bring several vulnerabilities in multiple layers. This is due to several reasons. Firstly, these entities cannot be efficiently managed and maintained, since the administration has been traditionally been delivered solely by humans, coping to make do with such large amounts of endpoints. Secondly, some of these entities are proprietary devices, such as diagnostic imaging devices, provided with software, or operating systems with well-known vulnerabilities (Windows NT, Windows XP) that are difficult or impossible to update or change. Finally, all devices and services are operated by humans, who do not necessarily have thorough knowledge of cybersecurity-related best practices and thus being susceptible to attacks, such as social engineering, or phishing, etc. Taking all the above-mentioned facts into consideration, it is obvious that all entities within a network must be continuously monitored and assessed in terms of vulnerability, by an automated service, requiring little to none, human interaction. In this respect, VAaaS will continuously monitor existing network-enabled entities within the SPHINX network and assess them on their vulnerability status. Each assessment will produce a detailed machine-readable report and a vulnerability score, based on the standardized vulnerability scoring system CVSS (First, n.d.). These reports will be distributed to several of the SPHINX components, such as the Situational Awareness component.

1.4.1 Design Principles

Taking into consideration the design and software development lifecycle (SDLC) principles narrated in deliverable D6.1, we designed and started the development of the VAaaS component, keeping in mind security and interoperability. The VAaaS component is an isolated application, comprising several sub-components, exposing only its necessary egress endpoints.

1.4.1.1 Stakeholders' Requirements Fulfilment

According to deliverable D2.6, the VAaaS component has four (4) basic functional requirements VAAAS-F-10, VAAAS-F-20, VAAAS-F-30, and VAAAS-F-40. These requirements fulfil some of the functional requirements provided by the stakeholders. The table below illustrates the functional requirement fulfilment between the VAaaS and the stakeholders. (Table 1: Functional requirement traceability).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

VAAAS-F-010	***STA-F-120***	*Devices' vulnerability assessment*
VAAAS-F-020	***STA-F-070***	*Protect against known cyber threats*
VAAAS-F-040	***STA-F-060***	*Link to external cyber threats repositories*
VAAAS-F-020	***STA-F-110***	*Vulnerability assessment*
VAAAS-F-030	***STA-F-110*** ***STA-F-120*** ***STA-F-130***	*Vulnerability assessment* *Devices' vulnerability assessment* *Vulnerability Assessment Checklist*
VAAAS-F-040	***STA-F-110*** ***STA-F-120***	*Vulnerability assessment* *Devices' vulnerability assessment*

Table 1: Functional requirement traceability

1.4.2 Human Factor

The Human factor for the current conception of the VAaaS, plays no significant role since human interaction/intervention is not required. VAaaS is an automated tool, which interacts with its environment and performs its functionality in a completely automated manner.

As far as the underlying environment is concerned, the vulnerability status of a devise or a service, is heavily dependent on the human factor, since human negligence or ignorance are some of the main reasons for miss-configured or un-configured devices that lead to exploitable vulnerabilities.

1.4.3 Technical Details

The VAaaS component is based on the opensource OpenVas tool and built with the Python3.8 programming language. The web application for the frontend administration was built with the Angular 9 web development framework, as a one-page application. The internal components of the VAaaS are presented in the figure below (Figure 1).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

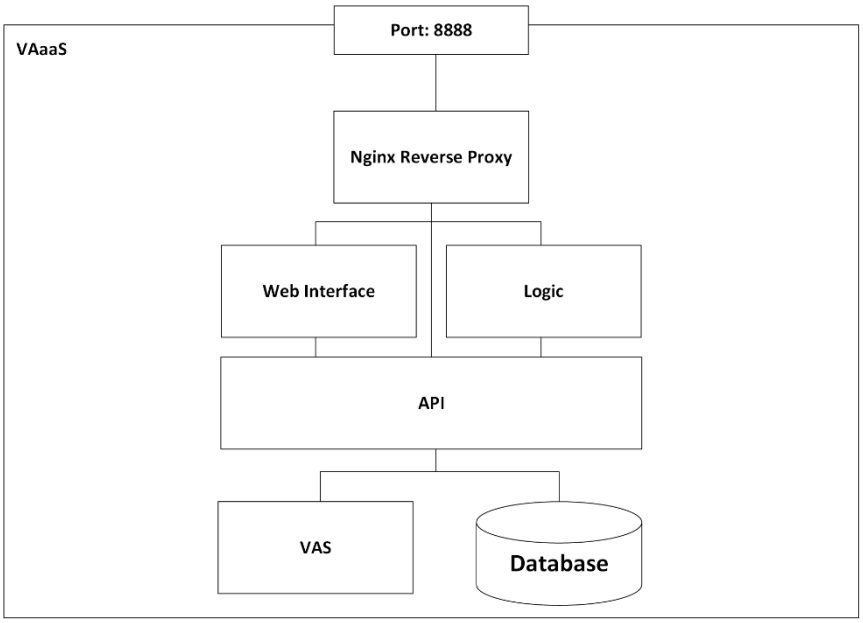

Figure 1 :VAaaS internal component diagram

1.4.3.1 Vulnerability Assessment Service (VAS)

The core of the VAaaS component is the Open Vas (Greenbone, n.d.) opensource vulnerability assessment tool. This is a well-known cybersecurity tool, which is a direct ancestor of the Nessus project. The tool is customized to accommodate the foreseen functionalities, and dockerised for the purposes of the SPHINX project and the requirements of the SPHINX deployment platform.

1.4.3.2 Application Programming Interface (API)

On top of the VAS component an API is developed to accommodate the overall management of the tool and all the foreseen functionalities. The API is developed in Python3.8, utilising an asynchronous, lightweight library (Sanic) (Huge Success, n.d.) to provide a fast and asynchronous RESTful API. The API endpoints developed so far, are displayed in the figures below (Figure 2). The API basically offers the appropriate endpoints to create and schedule a vulnerability assessment (Scan) and retrieve the produced report, in machine-readable form (JSON).

The API is structurally divided in six (6) thematic endpoints. These endpoints gather the whole current functionality of the tool. In more detail:

1.4.3.2.1 Targets

Targets, in our case are the network-enabled entities that are eligible for assessment. Each target is described by a set of meta-data, such as, IP address, description, MAC address, etc. Thus, the API allows the user to create and modify a target accordingly. The target can comprise one, or more hosts.

To create a target, users must POST a JSIN structured payload to the ***/targets*** endpoint. The basic payload is presented in the figure below (Figure 3).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

1.4.3.2.2 Scanners

Users can either utilize the preconfigured scanners, or they can create a new scanner with custom configurations. Currently, users can only use the existing scanners, by performing GET requests on ***/scanner*** or ***/scanner/{scanner_id}***.

1.4.3.2.3 Configs

Configs are the basic assessment configurations that instruct the scanner on how exhaustive the scan should be. Currently, users can only utilize the preconfigured configs, by performing GET requests in the ***/configs*** or ***/configs/{config_id}***.

1.4.3.2.4 Ports

The ***/ports*** endpoint is utilized by users to get or create a port range configuration, that will be utilized to create a target. Currently, users can only retrieve preconfigured port ranges.

1.4.3.2.5 Tasks

The ***/tasks*** endpoint is the most important part of the VAaaS API. It creates, updates, deletes, and retrieves tasks. To create a task, users must POST a JSON structured payload presented in the figure bellow (Figure 4). To create a target, users must utilize most of the above-mentioned endpoints, such as ***/config, /target, /scanner,*** etc.

1.4.3.2.6 Reports

Finally, after a task (assessment) has been completed, users can retrieve the assessment report by performing GET requests on the ***/reports*** endpoint. The report is structured in JSON format to allow for machine-readable results. This way, the consumption of the VAaaS results from all the internal SPHINX components/recipients, will be rendered seamless.

1.4.3.2.7 Response

The API's responses are uniform and structured in JSON format. The figure below presents the base structure of all responses produced by the VAaaS API (Figure 5).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

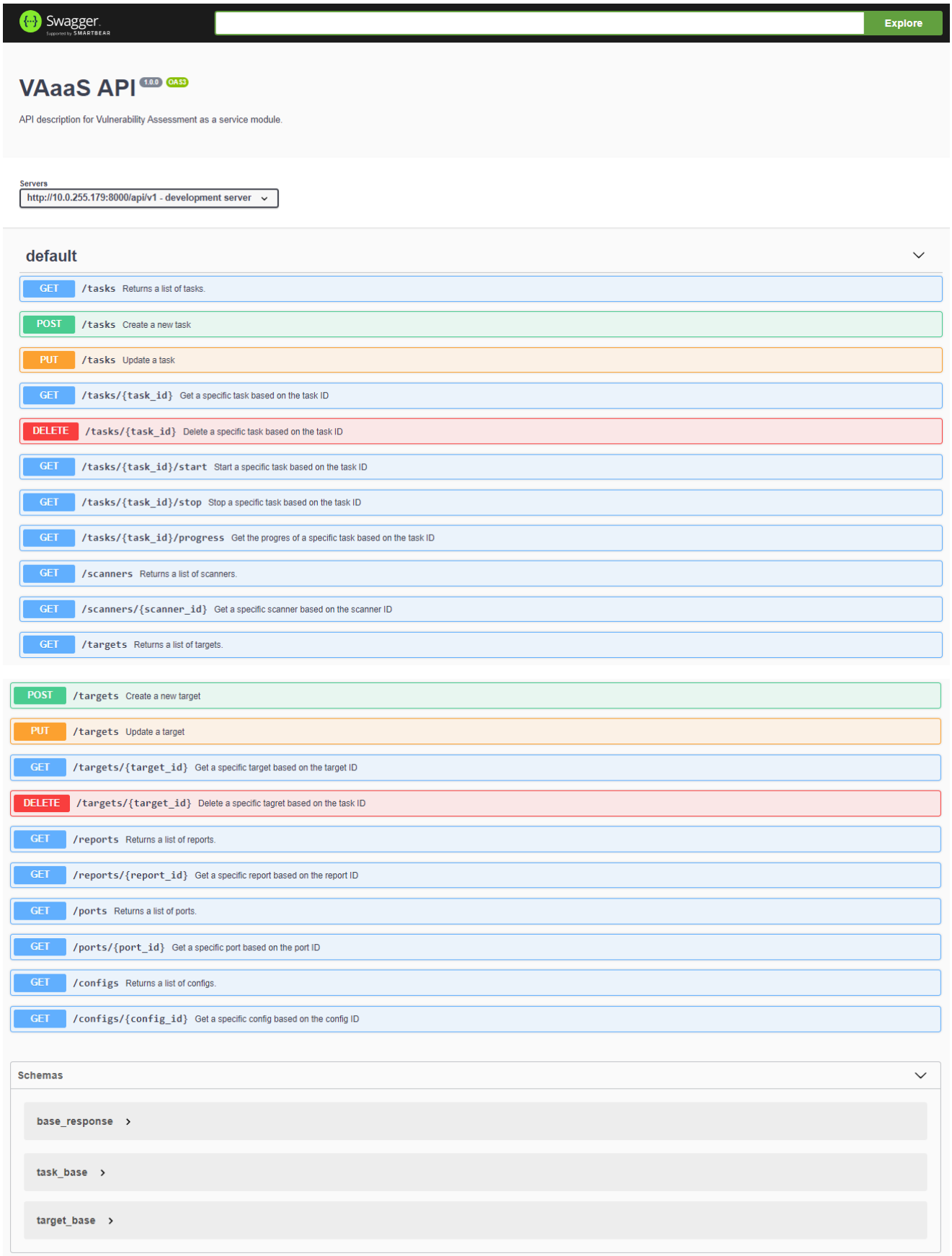

Figure 2: VAaaS API Swagger definition of RESTful endpoints

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

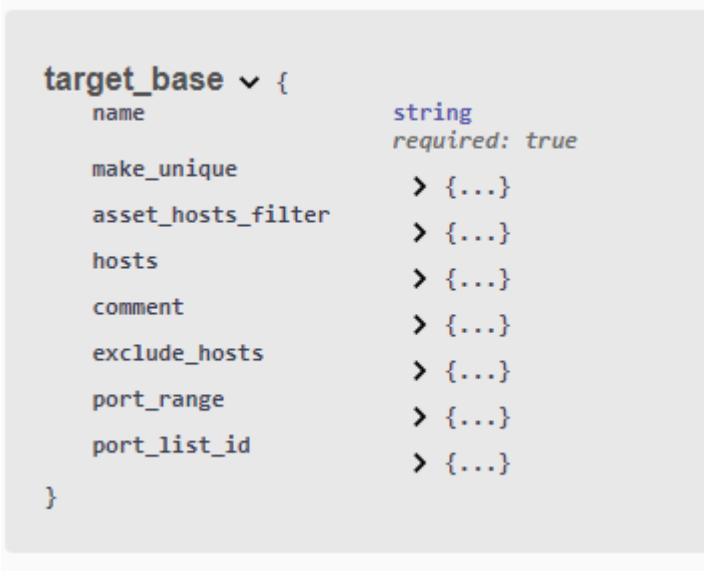

Figure 3: Create target base payload

Figure 4: Create task base payload

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

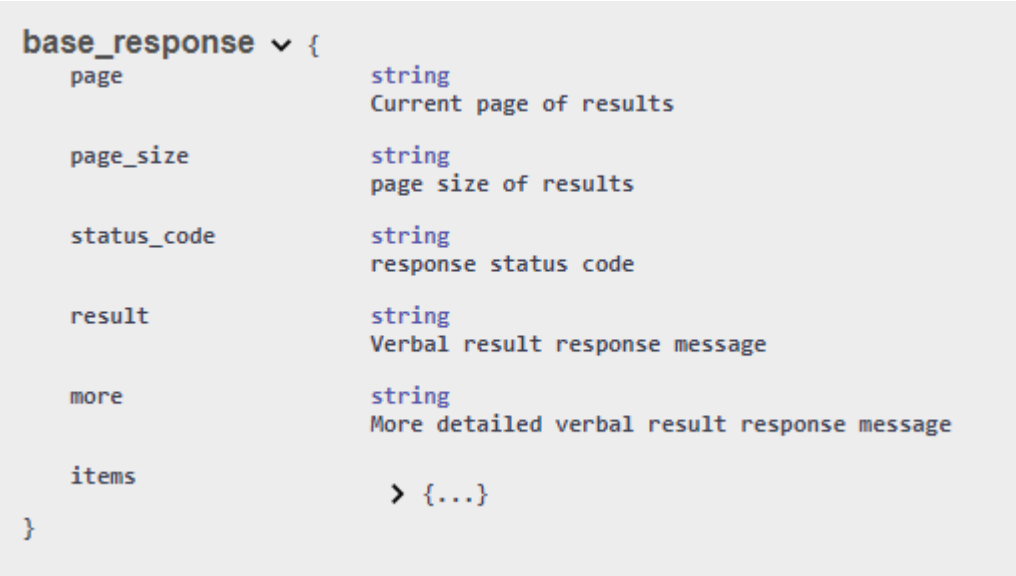

Figure 5: API response base format

1.4.3.3 Application Logic

This sub-component is the main automation logic of the VAaaS component. It is solely responsible for performing all the foreseen functionalities, such as retrieve the exiting network-enabled entities from the asset management component, maintain the local database by persisting the assessed/reassessed/non-assessed entities, perform the actual assessments (scans), and finally push the assessment result to all the corresponding SPHINX components (e.g. Situational Awareness). This component has not been developed yet, but it will be presented and detailed in the second version of this document.

1.4.3.4 Web Interface

The purpose of the web interface is to allow administrators to have a more thorough view/overview of the undergoing procedures within the VAaaS component, perform certain configurations, and perform ad-hoc actions, such as scan specific entities. The figures below illustrates the development status of the VAaaS web interface (Figure 6 - Figure 9).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

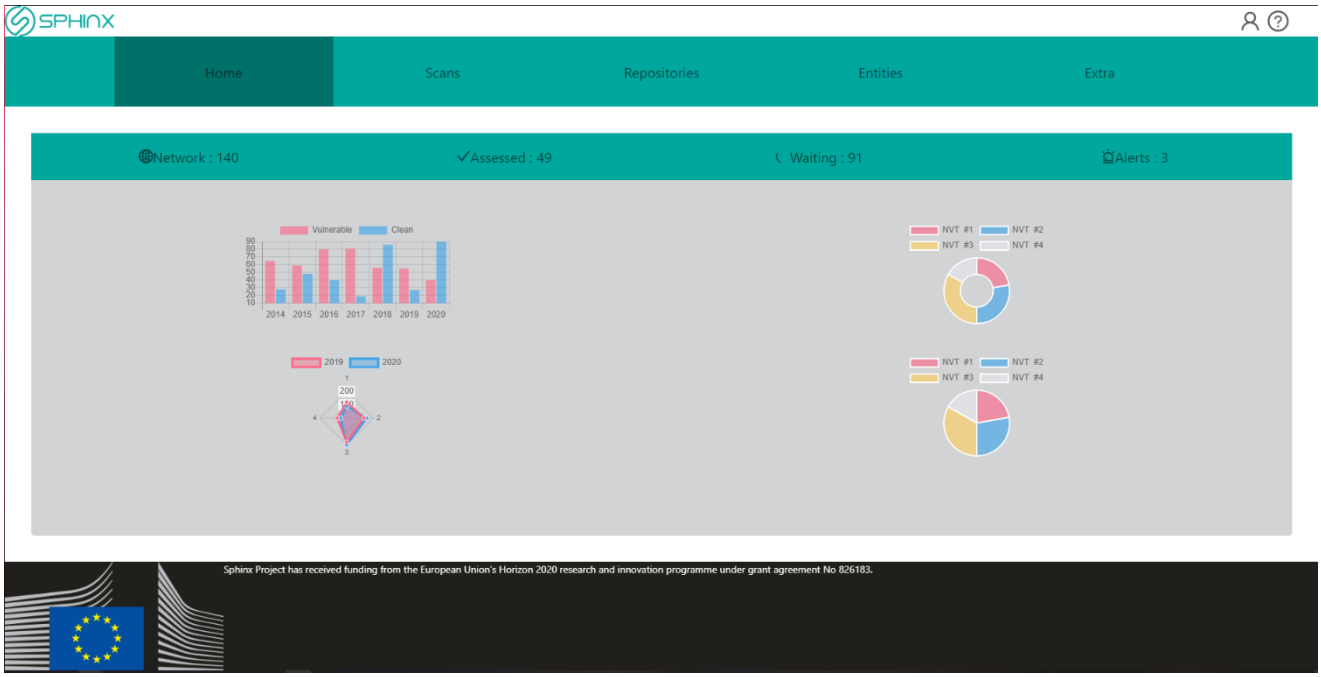

Figure 6: VAaaS web interface Home view

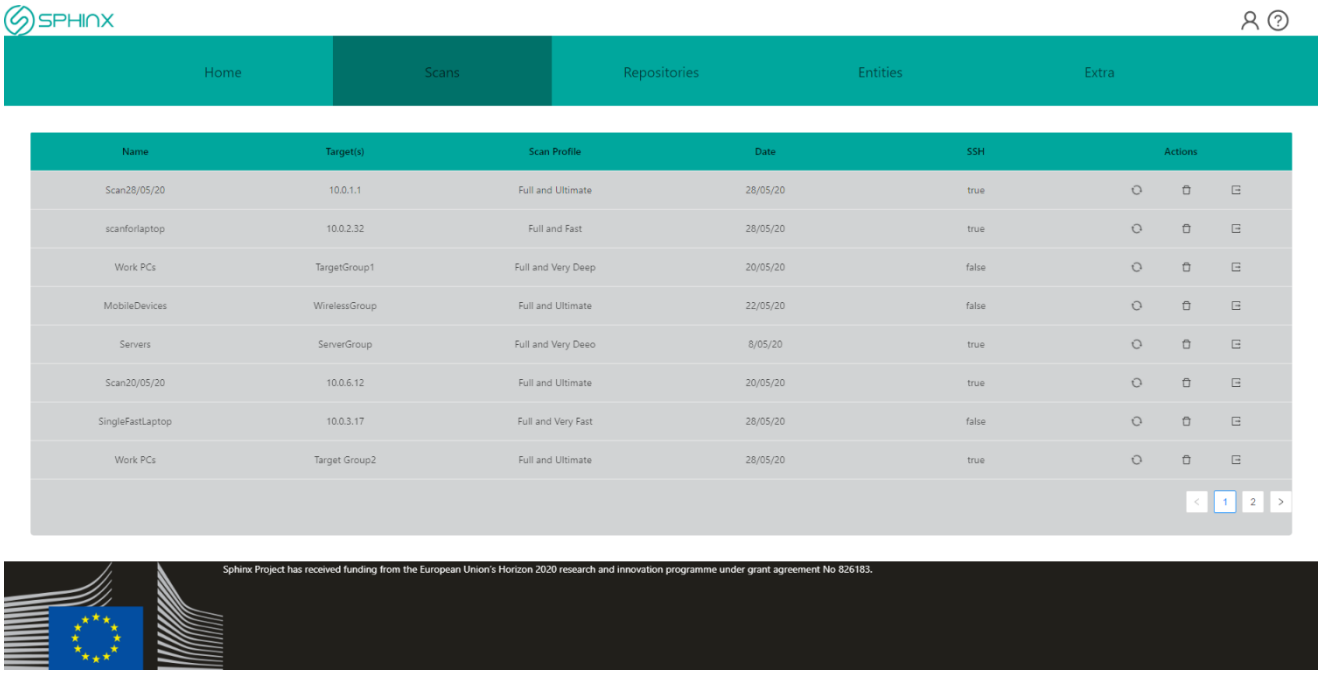

Figure 7: VAaaS web interface Scans view (list of past assessments)

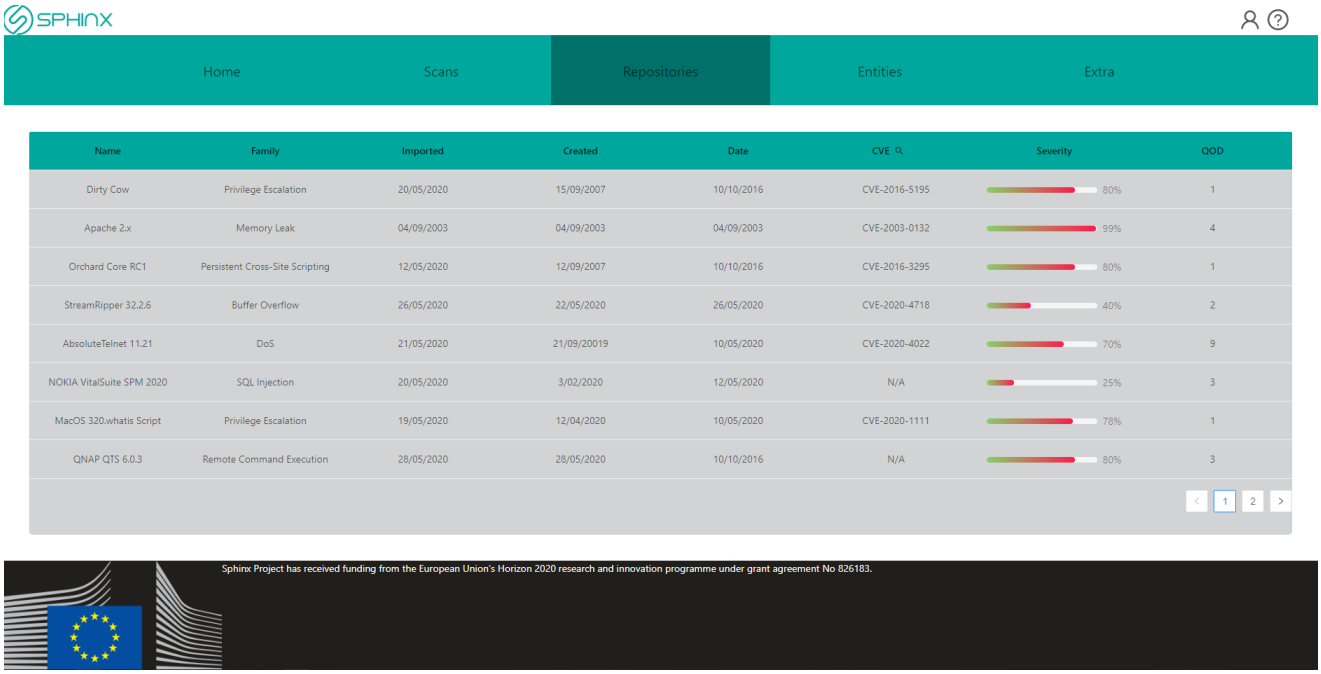

Figure 8: VAaaS web interface Repositories view (CVEs retrieved from external sources)

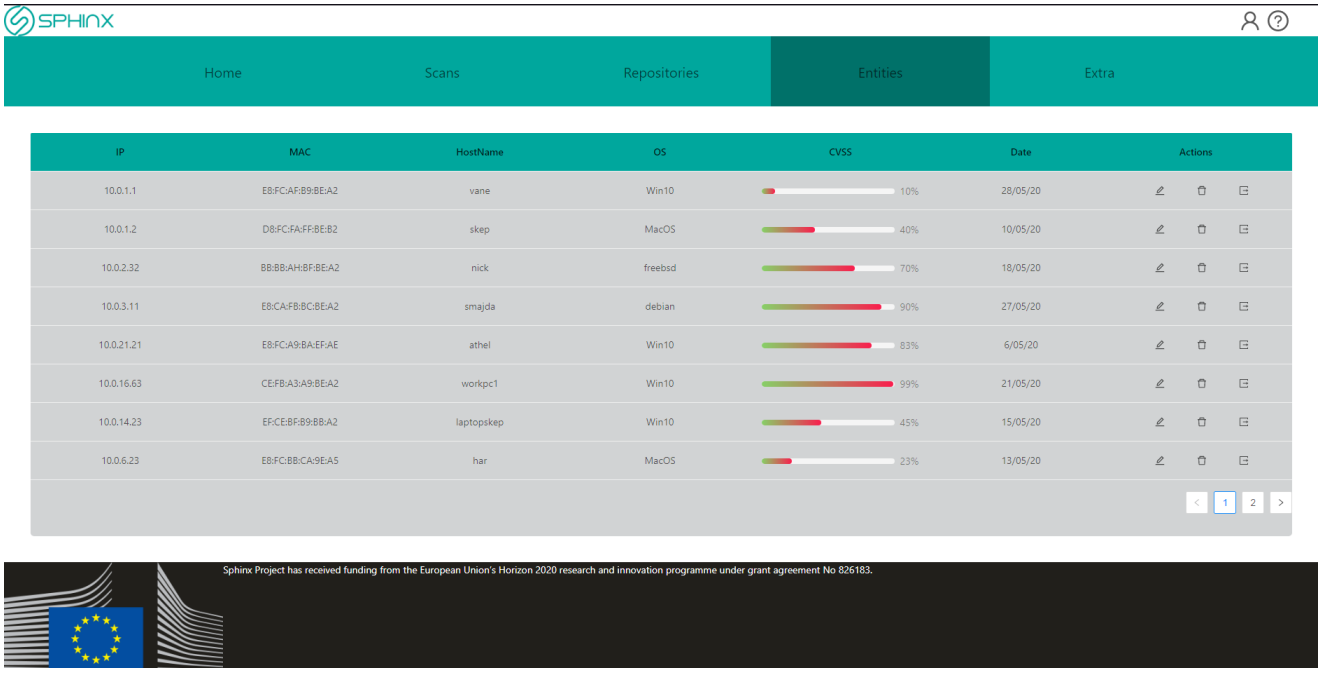

Figure 9: VAaaS web interface Entities view (network-enabled entities retrieved from asset management component)

1.4.3.5 Data Persistence Database

The VAaaS component requires to persist certain information, such as the entities list, accompanied by their assessment status, assessment date, assessment results, etc. For this purpose, the MongoDB document-based database was used, due do the framework's capabilities in terms of performance, scalability, integrity, and security.

1.4.3.6 Reverse Proxy

The VAaaS component comprises several sub-components, which expose several ports to allow internal connectivity and interaction. Following the microservice approach and the dockerisation concept, components must have a single point of entrance and perform a minimum number of tasks. To achieve this, the opensource NginX web proxy was utilised. We dockerised the proxy and used it as a reverse proxy of the API, only exposing TCP port 8888.

1.5 Background

In the cybersecurity domain, vulnerability assessment of entities in a network is one of the most important tasks in the overall complex endeavour of proactive threat detection. There are numerous open-source tools for vulnerability assessment; each one provides different functionalities and fulfil different requirements. Some of the most well-known are OpenScap (Open-Scap, n.d.), Nessus (Kenna Security, n.d.), now owned by Tenable, OWASP (OWASP, n.d.), Vuls (kotakanbe, n.d.). Most of the tools are very promising and provide state of the art functionalities. Nevertheless, few to none support standardised scoring of the assessment result. OpenVas fully supports the standardised Common Vulnerability Scoring System (CVSS) and moreover, it is written in C++, providing very efficient performance and scalability.

1.6 VAaaS in Sphinx

The VAaaS component pays a rather significant role in the SPHINX environment since it acts as an input for several internal components, such as the Decision Support System (DSS), the Real-time Cyber Risk Assessment (RCRA). A detailed description of these interdependencies is provided in deliverable D2.6 SPHINX Architecture v2, authored by EDGE. VAaaS performs two basic functionalities that are of great importance to the SPHINX environment, and its internal components. Namely,

i) it assesses all SPHINX devices and services against known vulnerabilities, in near-real-time
ii) it produces detailed reports of its findings and scores the results based on the CVSS (First, n.d.) v3 scoring system, and propagates those reports to the corresponding components that need them.

1.6.1 Vulnerability Assessment

The VAaaS component updates its internal database with extensively long lists of known vulnerabilities (CVEs) and scripts to perform individual assessments (NVTs) from external repositories, as well as from the internal SPHINX knowledgebase. Consequently, it performs vulnerability assessments to existing and newly introduced SPHINX devices and services (entities), to produce scores depicting their vulnerability status, and thus their risk level. Each entity's vulnerability status is a volatile state, since devices and services are most of the times operated by non-ICT related humans, whose awareness concerning cybersecurity is questionable. Thus, all entities are periodically re-assessed, to ensure they will not pose a threat to the SPHINX environment. Additionally, the tool, by exposing its API, it allows for ad-hoc assessments on specific targets, initiated by certain components such as the sandbox or the Cybersecurity Certification Service.

1.6.2 Vulnerability Reports to other components

The results of each vulnerability assessment are presented in extensively detailed reports, produced by the Vulnerability Assessment sub-component. These reports contain each discovered vulnerability and its

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

individual description, along with several extra information. Consequently, it is the API sub-component's task to transform these reports into a machine-readable format, so that other components can parse them and interpret them according to their needs, in an automated manner.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

2 Vulnerability Assessment

The VAaaS component is a mini ecosystem of microservices that combined provide unique functionality. Nevertheless, much like other SPHINX components, VAaaS relies on some other internal SPHINX components to perform its designated tasks. This section sheds some more light on the internal procedures that take place within the VAaaS component, as well as the interdependencies between the VAaaS and other SPHINX components.

2.1 Asset Management

This is the first and the most important interdependence that VAaaS must tackle to function properly. Without the list of existing entities, VAaaS cannot perform any scans. This functionality is provided by an internal SPHINX component provided by SIMAVI. This component provides a detailed list of the devices and services operating within the SPHINX network. The VAaaS utilises this list and performs its scheduled assessments accordingly.

2.2 Vulnerability Management

On its initial deployment, VAaaS updates its internal vulnerability database with daily updated CVEs, from several external repositories. Additionally, it retrieves any newly pushed vulnerabilities from the SPHINX knowledgebase. This also constitutes a rather crucial interdependency, since without an updated database, VAaaS will not be able to perform accordingly. The vulnerability management is a crucial part for VAaaS. This section will be further elaborated in the second version of this document, when the status of the tool, in terms of development will be mature enough.

2.3 Compliance to Standards

One of the most important advantages of the VAaaS component is its compliance with the standardised CVSS scoring system. Thus, the assessment results are scored based on CVSSs v3 scoring system standard. By following standardised frameworks and models, interoperability between different systems is allowed. Vulnerability Reports to Other Components (Figure 10).

CVSS is calculated based on three (3) basic metric groups: Base, Temporal, and Environmental. The base metric group refers to vulnerabilities whose characteristics remain constant over time and across environments. On the other side, the temporal metric group refers to vulnerabilities whose characteristics change over time but are not affected across different user environments. Finally, the environmental metric group refers to vulnerabilities whose characteristics are relevant and unique to particular user environments.

The scoring system, has a range between 0.0 and 10.0 and is calculated with a series of equations as presented in Figure 11. The calculation of the CVSS score is mainly a product of the equations from the base metrics group. The calculation of the temporal and environmental metrics is optional but performed in most of the cases. The base metric group equations derive from two sub-equations, the exploitability and the impact equations. The first derives from the base exploitability metrics and the later from the base impact metrics. A rather detailed description of the standard and its calculation algorithms are detailed in the standard's specification document. (First, n.d.) Finally, the score severity is mapped to a five-level scale(Table 2: CVSS Qualitative severity rating scale).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

Rating	CVSS Score
None	0.0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

Table 2: CVSS Qualitative severity rating scale

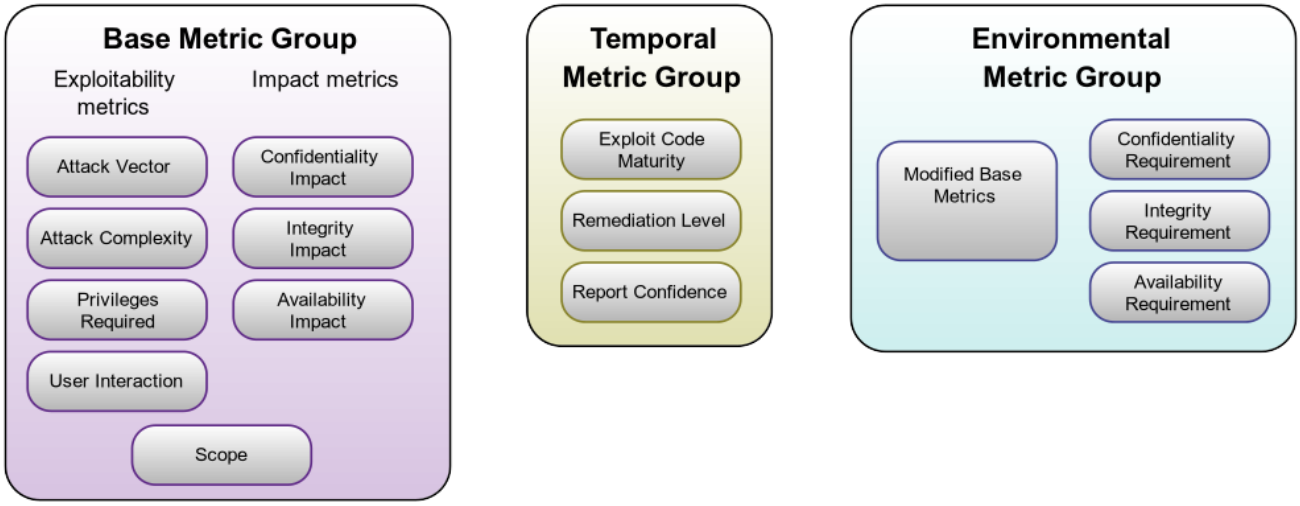

Figure 10: CVSS Metric groups

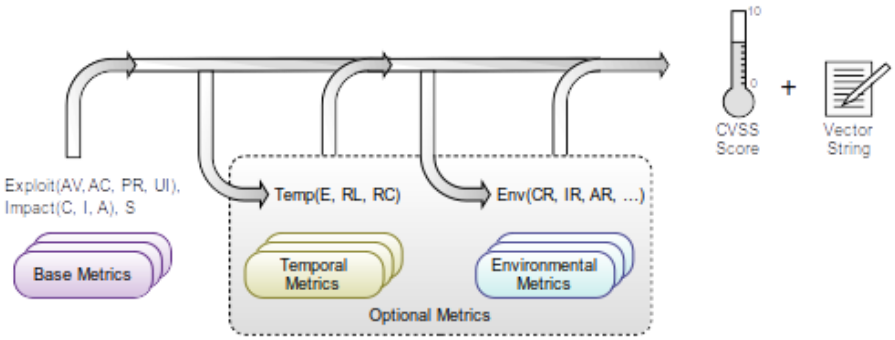

Figure 11: CVSS metrics and equations

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

3 Summary and Conclusions

This deliverable presented the overall development status of the VAaaS component on M18 (May 2020) of the project's lifetime and the end of the first interim of VAaaS two-staged development phases (M10-M18, M22-M30). This is a versioned document and describes the progress of the first development interim of the component. All the internal parts of VAaaS were presented and explained along with the component's role and scope within the SPHINX ecosystem. Several inter-component interdependencies were described and the significant role they play to all components' lifecycle. The second version of this document will present all the foreseen functionalities since it will report the second and final development interim of the components' development.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

4 References

First. (n.d.). Retrieved from First: https://www.first.org/cvss/

First. (n.d.). Retrieved from First: https://www.first.org/cvss/v3.1/specification-document

First. (n.d.). *First*. Retrieved from https://www.first.org/cvss/specification-document

Greenbone. (n.d.). *Github*. Retrieved from https://github.com/greenbone/openvas

Huge Success. (n.d.). Retrieved from Github: https://github.com/huge-success/sanic

Kenna Security. (n.d.). Retrieved from Github: https://github.com/KennaSecurity/chef-nessus

kotakanbe. (n.d.). Retrieved from Github: https://github.com/future-architect/vuls

Open-Scap. (n.d.). Retrieved from Github: https://github.com/OpenSCAP

OWASP. (n.d.). Retrieved from Github: https://owasp.org/www-community/Vulnerability_Scanning_Tools

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.