HORIZON 2020

Digital Security: Cybersecurity, Privacy and Trust
H2020-DS-2015-1

DS-04-2015 Information driven Cyber Security Management
Grant n° 700176

Secure Information Sharing Sensor Delivery event Network[†]

Deliverable D6.1: Trial definition and test plan

Abstract: The document contains the specification of the trial methodology and test plan of the SISSDEN platform. As a live document, it will be updated regularly during the project in order to provide full test coverage of the system, include additional tests and provide more details of the existing test scenarios as components reach maturity.

The document focuses on high-level tests verifying the functioning of the platform as a whole and on tests of public aspects of the system, such as public interfaces. For a full specification refer to deliverable D6.8 “Trial definition and test plan – non-public”.

Contractual Date of Delivery	30 April 2017
Actual Date of Delivery	4 August 2017
Deliverable Security Class	Public
Editor	Adam Kozakiewicz (NASK), Piotr Kijewski (SHAD)
Contributors	All *SISSDEN* partners
Internal Reviewers	Massimiliano Aschi (POSTE)
Quality Assurance	Arturo Campos (CYBE), Piotr Bazydło (NASK)

[†] The research leading to these results has received funding from the European Union Horizon 2020 Programme (H2020-DS-2015-1) under grant agreement n° 700176.

The *SISSDEN* consortium consists of:

Naukowa i Akademicka Sieć Komputerowa	Coordinator	Poland
Montimage EURL	Principal Contractor	France
CyberDefcon Limited	Principal Contractor	United Kingdom
Universitaet des Saarlandes	Principal Contractor	Germany
Deutsche Telekom AG	Principal Contractor	Germany
Eclexys SAGL	Principal Contractor	Switzerland
Poste Italiane – Società per Azioni	Principal Contractor	Italy
Stichting the Shadowserver Foundation Europe	Principal Contractor	Netherlands

Table of Contents

1 Introduction

1.1 Aim of the document

The goal of the document is to specify the trial methodology and test plan of the SISSDEN platform. The focus of this document is a technical trial of the platform developed in the project, understood as a verification of the proper functioning of the platform in its intended applications and confirmation of meeting the project goals. As a live document, it will be updated regularly during the project in order to provide full test coverage of the system, include additional tests and provide more details of the existing test scenarios as components reach maturity.

The document focuses on high-level tests verifying the functioning of the platform as a whole and on tests of public aspects of the system, such as public interfaces. For a full specification refer to deliverable D6.8 “Trial definition and test plan – non-public”.

1.2 Work done so far

Following the design work documented in the project’s earlier deliverables, most importantly D3.1 (Use cases and requirements) [2], D3.3 (Initial technical architecture) [3] and D4.1 (Data exchange interfaces specification) [4], active development of the SISSDEN platform has started. The pilot has been launched as planned, in April 2017, as shown in deliverable D6.3 (Activation of platform pilot) [5]. Further development is ongoing.

1.3 Structure of the document

Chapter 2 defines the approach taken to test coverage and defines some crucial terms. Chapter 3 of the document provides an overview of the project’s development phases and illustrates how the testing methodology is expected to evolve in correspondence to the maturity of the platform. Chapter 4 discusses the implications of the projects strict approach to ethics and privacy guidelines for the testing process. The core of the document is chapter 5, which specifies a list of tests. Chapter 6 revisits the issue of coverage, providing a mapping of the tests to defined public project artefacts. Finally, the content of the document is summarized in Chapter 7.

2 Terminology and test coverage

The focus of this document and its confidential counterpart D6.8 [6] is a technical trial of the platform developed in the project, understood as a verification of the proper functioning of the platform in its intended applications and confirmation of meeting the project goals. Therefore, the tests specified in this document are focused on the verification of project artefacts.

> ***Project artefact:*** *A defined, verifiable part or characteristic of the SISSDEN platform, such as a use case, requirement, public interface or component.*

Other types of tests (e.g. unit tests) can and will be used in development of the platform, but the amount of tests involved and their narrow scope makes it impractical to include them in an official report.

According to the above definition, four types of project artefacts have been identified within SISSDEN. The following sections describe each type and the approach taken to its verification. Note that while all project artefacts should by definition be verifiable, the appropriate verification method is not always through tests (e.g. code review, documentation review, etc.) – hence, project artefacts can be non-testable.

2.1 Use cases

The use cases have been defined in the deliverable D3.1 (Use cases and requirements) [2]. Since that document is confidential, the list of use cases is not generally available to the public and detailed description of the verification approach is only provided in deliverable D6.8 [6]. Here we only present the general approach.

Each use case defines some type of activity that can be performed using the platform. The kind and complexity of activity can and does differ significantly between use cases. A use case defines only what should be achievable, but the details of how this is achieved may not be clear (the use cases were created prior to the architecture and design work, as input for the designers). As a result, a use case is not testable a priori. Some simple use cases may map directly to some testable functionality of the platform and are themselves testable as a result, but that is by no means guaranteed. For example, a use case regarding ability to opt-out from a service describes a simple, testable function, while a use case regarding arbitrary analyses on collected data involves a complex interplay of different functions of the system, dependent on actual data and can only be demonstrated.

Verification of most of the use cases is therefore descriptive – we will show that a given activity can be performed using the SISSDEN platform. Tests are not expected to be specified explicitly for use cases (although they can be for testable ones). However, tests specified for other project artefacts can support the verification of a given use case by showing that functions needed for the activity are available and functioning properly.

The use case coverage is therefore defined as follows:

- Testable and non-testable use cases are identified,
- Tests will eventually be available for all testable use cases (unless the functionality defined in the use case is not delivered),

- Non-testable use cases have supporting tests identified where possible, which verify various testable functions of the system clearly needed to perform the activity.

2.2 Requirements

Requirements have been defined in the deliverable D3.1 (Use cases and requirements) [2]. Since that document is confidential, the list of requirements is not generally available to the public and detailed description of the verification approach is only provided in deliverable D6.8 [6]. Here we only present the general approach.

Both functional and non-functional requirements have been defined. While care has been taken to assure that every requirement is verifiable, many of them are not testable and need another method of verification. Some of them are organisational or structural by nature and must be verified through documentation (e.g. requirements to provide different types of documentation, regarding location of sensors, use of open source, etc). This involves showing that e.g. the design follows the requirement, that certain actions have been taken, targets (like number of sensors) have been met or that certain forbidden functions are consciously absent in the design. Full coverage is achieved by identifying the non-testable requirements and providing tests for all others. During verification, full coverage includes both execution of all relevant tests and providing appropriate proof for non-testable requirements.

Another important issue is the fact that the requirements defined for the project vary in priority. While all requirements with MUST priority must be implemented, the lower priorities (SHOULD and COULD) may be omitted. Since omitted requirements will obviously not be verified, the standard of full coverage is lowered. In general:

- Every testable MUST requirement must be covered by tests.
- All testable SHOULD requirements should be covered by tests unless the requirement is for any reason found very unlikely to be implemented.
- COULD requirements may not have any tests defined until there is a clear expectation that they will be implemented.
- Tests may not be specified if the design of a particular part of the system is not yet clear enough and there are multiple possible ways to implement the functionality, requiring different tests. This exception does not apply to requirements which were planned for early delivery, as those are already included in the design presented in deliverable D3.3.

2.3 Public interfaces

Public interfaces have been defined in the deliverable D4.1 (Data exchange interfaces specification) [4]. Since that document is public, the full descriptions of the interfaces are available. Therefore, we expect to be able to define tests or other verification methods in such a way that neither their description nor results reveal non-public aspects of the system, allowing them to be included in public deliverables. As a result, tests and coverage analysis for the public interfaces are included in this report.

A public interface is a technical function (or rather set of functions) of the system and as such can be verified through testing. It is also a user-facing part of the system, making hard to define, non-functional aspects of it important as well. The approach to verification is therefore two-fold:

1. Formal verification through testing. Proper functioning of each of the interfaces should be confirmed with a (set of) well-defined test(s).
2. Usability verification. No formal verification is required. The user feedback interface provided by the system (first for consortium usage in a simple form, then, after the maturing system is made available publicly, as a fully developed interface for all users) enables this to be a continuous process in the later stages of the project. This will be shown by summarizing the feedback in final documents of the project.

2.4 Components

Components of the SISSDEN platform are the basic building blocks identified in deliverable D3.3 (Initial technical architecture) [3]. Since that document is confidential, the full list of the components is not publicly available. and detailed description of the verification approach is only provided in deliverable D6.8 [6]. Here we only present the general approach.

Components are generally deployed software modules and/or hardware and as such are testable by default. Their complexity varies, so the number of tests needed to fully cover the functionality of a given component may vary as well. Tests must be provided for all components.

Two (non-software) exceptions have been found during analysis – components for which the general approach is not practical. These do not have any corresponding tests, but are discussed and have other methods of verification proposed in D6.8.

3 SISSDEN development and testing methodology

This chapter explains the different development phases of the SISSDEN platform and the expected approach to testing at each stage.

3.1 SISSDEN platform development stages

The SISSDEN platform is actively developed throughout the project lifetime, but four milestones mark important development phases. These affect both the required contents of this live document and the testing method applied on each stage.

3.1.1 Stage 1 – First System (FS)

This stage corresponds to Milestone MS3 of the project, as defined in the Description of Action. This point is reached when the platform is first launched with a basic subset of functionality, mostly as a development platform. The verification is limited to showing that the core functionality is already present and the architecture is viable.

3.1.2 Stage 2 – First Validation (FV)

This stage corresponds to Milestone MS4 of the project, as defined in the Description of Action. At this stage the core functionality of the system is ready for small-scale operational use. Use of manual, temporary operating procedures is minimized in favour of scalable automation. The system enters a growth phase, connecting new sensors. Validation is focused on the core functionality, but the testing is much more thorough than in the previous stage. As this is a first validation, negative test results are acceptable and treated as feedback for further development, as long as the platform as a whole delivers usable functionality.

3.1.3 Stage 3 – Maturing System (MS)

This stage corresponds to Milestone MS5 of the project, as defined in the Description of Action. At this stage the core functionality of the project is present and verified in real-life use conditions. The analytical components developed in work package WP5 are beginning to be deployed and used. The verification focuses on the newly added components. The non-functional requirements regarding performance and stability are beginning to be verified. The functionality available at the previous stage is expected to be already stabilized – if any negative test results are obtained, they must be thoroughly analysed and plans for fixing them should be proposed.

3.1.4 Stage 4 – Fully Operational (FO)

This is the final stage, coinciding with the end of the project and the Milestone MS7, as defined in the Description of Action. The whole system is expected to reach maturity at this point. The core functionality is both tested thoroughly and verified in everyday operational use at scale, at a level corresponding to TRL 9. The analytical part (WP5) is verified through thorough testing on real-life data and limited operational use, as expected at TRL 7. The testing at this phase must cover all testable requirements with MUST priority and all implemented requirements with SHOULD priority. All tests for the core functionality must pass for the project to be considered fully successful.

3.2 Testing method

In order to keep the test specification manageable, it is not required for the tests to be specified separately for each project artefact. A single test can be used to verify multiple requirements, components, etc. This approach is more scalable at the cost of higher difficulty of interpreting negative test results.

Since different stages require different levels of verification, the method applied also differs. Notably, at the final stages (MS and FO) the approach is more formalized – we assume that each test should be performed by a different Partner than the one who specified the test. Ideally the Partner responsible for the test should not be the author of any components involved in the test. This requirement may not be feasible to implement for some large tests involving the full system, in that case we will select the least involved Partner.

3.3 Lifecycle of this document

As stated above, this report is a live document and will continue to be expanded during the entire project. The expected contents depend on the current stage. The following sections explain this evolution.

3.3.1 Test specification levels

The following specification uses the following keywords referring to the level of detail to which a test is specified:

INFO – The test specification identifies the function or metric to be verified (the WHAT), enabling the reader to assess whether the envisaged test will sufficiently confirm that an artefact it covers is indeed implemented and working.

DESC – The test specification provides at least a rough description of the envisaged testing procedure (the HOW), enabling the reader to assess not only whether it can show that the artefact is implemented and working, but also whether the test itself is generally feasible and adequate for the task.

ROUGH – The test specification includes a concrete testing scenario with identified steps. The test's adequacy and feasibility can be fully assessed at this point and the methods to be used at each step are clear. The test can be used by developers, as each step can be performed with arbitrary data. It is also possible to identify any necessary testing tools that need to be developed to perform it.

FULL – The test specification is complete, including all necessary input and output data. The test can be performed by non-developers, since the specification provides sufficient information. In general such tests could usually be automated, although depending on the test it may not be practical or cost-effective.

3.3.2 Stage 1 – First System (FS)

The document should already provide a description of the testing approach adopted by the project and how it is expected to change during the project lifetime.

Regarding coverage, the document should show that full coverage is achievable, especially for the first few stages. For each testable project artefact (use case, requirement, public interface or component) the following assumptions are adopted, depending on the stage at which it is to be delivered:

For each artefact expected at stages FS or FV, there should exist at least one DESC-level test specification covering its core.

For each artefact expected at stage MS or FO, there should exist at least one INFO-level test specification covering its core. More detailed specification can be provided, but is not yet mandatory. Tests can be omitted if the design of a given part of the system is not yet clear enough or for optional functionality (COULD priority).

FULL-level specification is expected for at least one test at this stage: A test involving the full data flow from registering an event at a sensor in the system to successful delivery of a remediation report covering that event. This test is used to verify the availability of the basic functionality of the platform, proving a successful activation of the pilot – deliverable D6.3 [5].

If more tests are fully specified at this stage, they may or may not be performed at this stage. In general, any such tests are treated as early contribution to the First Validation stage.

3.3.3 Stage 2 – First Verification (FV)

The document should be extended with additional tests and the existing tests should be expanded. The expected coverage is as follows:

For each artefact delivered at stage FS or FV at least one FULL-level test covering its core functionality must be provided. If needed, additional INFO-level test specifications should be provided for these artefacts. The reader should be able to assess whether the planned range of tests sufficiently covers the artefact and whether the FULL-level test is a sufficient proof of the artefact being present.

For artefacts expected at stage MS, at least one DESC- or ROUGH-level test covering its core must be provided. If needed, additional INFO-level test specifications should be provided for these artefacts. The reader should be able to assess whether the planned range of tests sufficiently covers the artefact and whether the DESC-level test is a sufficient proof of the artefact being present. Tests can be limited to INFO level if the design of a given part of the system is not yet clear enough or for optional functionality (COULD priority) – in the latter case they can be omitted if the functionality is deemed unlikely to be delivered.

For artefacts expected at stage FO, at least one DESC- or ROUGH-level test covering its core. More tests can be provided at any specification level, but full coverage of the artefact is not expected at this stage.

3.3.4 Stage 3 – Maturing System (MS)

The document should be extended with additional tests and the existing tests should be expanded. The expected coverage is as follows:

For each artefact delivered at stages FS or FV the tests should sufficiently cover the artefact, showing all of its aspects. In general all tests should be specified at FULL level, unless some aspect of the artefact (e.g. a complex component) is still under development – in this case the corresponding tests may be specified to DESC level.

For each artefact delivered at stage MS at least one FULL-level test covering its core functionality must be provided. If needed, additional DESC-level test specifications should be provided for these artefacts. The reader should be able to assess whether the planned range

of tests sufficiently covers the artefact and whether the FULL-level test is a sufficient proof of the artefact being present.

For artefacts expected at stage FO, at least one DESC- or ROUGH-level test covering its core must be provided. If needed, additional INFO-level test specifications should be provided for these artefacts. The reader should be able to assess whether the planned range of tests sufficiently covers the artefact and whether the DESC-level test is a sufficient proof of the artefact being present.

3.3.5 Stage 4 - Fully Operational (FO)

The document should be extended with additional tests and the existing tests should be expanded. For each artefact delivered at any stage the tests should sufficiently cover the artefact, showing all of its aspects. All tests must be specified at FULL level.

Additional tests may be specified for artefacts which weren't delivered (e.g. low-priority requirements) but have been sufficiently researched and are planned as future work, post-project. Such tests can be at any specification level, but should not be mixed with the tests of delivered functionality in order to preserve the clarity of the document.

3.3.6 Lifecycle summary

The following table summarizes the above expectations, giving the minimal expected specification level at each stage for:

- Core – one core test, verifying the presence of a given testable artefact,
- Cover – full set of tests covering the functionality defined by the given testable artefact.

Note that the number and scope of tests may also grow as already delivered functionality is extended and enriched. The minimal levels in the table correspond to already delivered or at least fully designed functionality only.

Also note that these are minimal levels – if specifying a test before it is required or at a higher level than required at a given stage is possible, doing so is obviously encouraged.

Document at stage: →	**FS**		**FV**		**MS**		**FO**	
Artefact delivered at ↓	**Core**	**Cover**	**Core**	**Cover**	**Core**	**Cover**	**Core**	**Cover**
FS	DESC		FULL	INFO	FULL	FULL	FULL	FULL
FV	DESC		FULL	INFO	FULL	FULL	FULL	FULL
MS	INFO		DESC	INFO	FULL	DESC	FULL	FULL
FO	INFO		DESC		DESC	INFO	FULL	FULL
Low priority (optional) or not designed yet					INFO if possible		INFO if possible	
Main full data flow test	FULL from initial version of the document							

4 Privacy and ethics aspects

Like all other activities in the SISSDEN project, testing must also be considered from the privacy and ethics point of view.

4.1 Threats and liabilities

Special attention must be paid during the different testing phases of the project to monitor any privacy and ethical aspects. Identifying the most relevant privacy issues during the first stages of the project, namely First System (FS) and First Validation (FV) before going to public operation is key to finding appropriate solutions to privacy related challenges in future phases of the project. Having privacy in mind (Privacy-by-design) during the initial testing phases is a design principle of SISSDEN.

The “Privacy by Design” approach adopted by SISSDEN consciously tries to avoid future problems in this area. The tests shown in this document (and in D6.8 as well) do not include any actions that could be considered problematic from the Privacy and Ethics point of view. The currently defined tests either use data already collected by the project or artificially provided for testing purposes. In either case, the Ethics Advisor will assess the datasets and provide prompt advice to the Consortium if any privacy risks emerge.

Most of the tests and their results are not public and consortium Partners are bound by the rules set for personally identifiable data processing, as defined in the Description of Action [1] and privacy-related deliverables.

Nevertheless, each version of this document and its non-public counterpart D6.8 is reviewed by the Ethics Advisor to monitor that privacy guidelines are followed.

The same rule is applied to any deliverables including test results, especially public ones, as careful redaction of test results is needed to avoid accidental publication of PII (e.g. shown in neighbouring records, while the test-related ones are PII-free by design).

Whenever a test is likely to contain protected content in the results (even by accident), the additional fields of the test specification warn about this to simplify the preparation of deliverables with test results.

4.2 Opportunities

The tests provided in this document (and its non-public counterpart D6.8) provide ample opportunities to verify the privacy aspects of the SISSDEN platform. While the data used in many tests is artificial, their protection still follows the same principles as real data. Therefore, testers are encouraged to verify if proper privacy protection is applied wherever appropriate. Additional fields in test specification are used to clarify the details, both pointing out the likelihood of the test involving directly protected information and explaining which privacy protection issues can be verified during test.

These tests and their results will be reviewed by the Ethics Advisor and may result in modified, privacy-centric versions being created as supporting tests for privacy-related requirements.

5 Test specification

This chapter presents the specification of the various tests which will be performed during the validation of the platform. This public document provides only high-level tests of the overall system functionality and its public interfaces; some details of the tests may be omitted if they reveal non-public aspects of the system design. The complete specification of all tests is provided in the confidential document D6.8 [6].

The description of each test uses the following tabular format:

Test ID	*See below*	**Spec level:** *see below*	**Version:** *see below*
Test Name	The name should be short, but make it clear what is being tested. Uniqueness is encouraged, but (unlike the test identifier) a name of a cancelled test can be reused if appropriate.		
Coverage	List of project artefacts (use cases, requirements, components, interfaces) covered by the test.		
Description	This is a short description of the test's goals and general testing procedure.		
Prerequisites	Optional specification of the technical requirements for starting the test.		
Input	The detailed data to be used. Only used if needed, generally only on FULL specification level.		
Scenario	The steps to be performed, mandatory for ROUGH and FULL specification level.		
Exp. output	The expected results of the test needed to pass. If possible, exact results should be shown, otherwise a description of the required qualities of the observed output is provided. Expected for ROUGH and FULL specification level (more detailed in the latter case).		
Privacy: *see below*	Privacy considerations – optional comment outlining privacy-related aspects of the test and/or commenting on the privacy field to the left.		

The **Test ID** field contains a unique identifier of the test. Apart from uniqueness, no other restrictions are placed on it, although very long identifiers are discouraged. Note that identifiers cannot be reused if a test is cancelled.

The **Spec level** field gives a test specification level, as explained in the previous chapter. The valid values for this field are: FULL, ROUGH, DESC, INFO, CANCELLED. The first four values correspond to the specification levels described earlier. The special fifth value is used for tests which have been specified earlier but will not be performed, either due to a decision not to implement some low-priority functionality or due to design changes requiring a very different validation method. Cancelled tests are kept in the document in order to simplify control of the uniqueness of identifiers – only the first line from the table is preserved, optionally the Description field may be used to list tests superseding it.

The **Version** field is used to track changes in test specification and follows the format *x.y* where *x* is the major version number and *y* – minor version number. Editorial and minor technical changes should be reflected by increasing the minor version number. The major version number should be increased (resetting the minor version number to 0) whenever the test reaches a higher specification level or when the technical changes significantly modify the nature of the test. Note that if a test becomes completely different as a result of the latter type of change, the preferred approach is to cancel it and specify a new one, superseding it.

The initial version numbers for each tests have been assigned for the first release of this document depending on specification level. Tests at INFO or DESC specification level start with version 0.1, while those which have already reached ROUGH or FULL levels start with version 1.0.

The **Privacy** field is used to mark tests which are likely to involve PII processing either by design or as a potential side effect. The field takes one of three values: YES, NO or MAYBE. Tests marked as YES need to be performed in a controlled manner and analysis of the effectiveness privacy protection mechanisms involved (if any) is required. Tests marked as MAYBE have the potential to involve PII, but it cannot be clearly established at this point, either because the test specification is not yet detailed enough or because presence of PII is considered unlikely – the assessment should be repeated before executing the tests and if a clear answer is not obtained the testers are advised to err on the side of caution. Tests marked NO are not expected to raise any privacy issues.

The field to the right of the Privacy field is used to provide privacy-related comments. Note that use of this field is not strictly tied to the value of the privacy field – it is both possible to omit the comment for a test marked with YES in the Privacy field if the nature of the privacy issues with the test is sufficiently obvious, or provide a comment for a test marked with NO if it can provide useful observations of the privacy-preserving mechanisms, while not involving PII directly.

5.1 Full system tests

The following test is a verification of the full data flow of the core TRL9 functionality of the SISSDEN platform – remediation feeds. As such, it was used to verify the successful initial launch of the platform.

Test ID	FREM1	**Spec level:** FULL	**Version:** 1.0
Test Name	General test of the full remediation workflow		
Coverage	**Public interfaces:** 2: Free Remediation Report Sign Up; 4: Shadowserver Reporting System		
Description	Verifies the full operation of the system in the task of providing remediation feeds. The user signs up for a remediation feed, then a simulated attack is performed from his network. The attack is detected by the honeypots and processed by the system, resulting in a report being delivered to the user.		
Prerequisites	• A test network with at least one computer capable of connecting to a selected honeypot. • A working sensor supported by an SSH honeypot.		
Input	• Report recipient network data details (report subscription) • Attacker IP • Target Honeypot IP • Attacker payload: manual use of telnet client to simulate logins to honeypot		
Scenario	1. Connect to the user interface for registration to remediation feeds. 2. Provide requested data, including the network from which the simulated attack will be performed. 3. Wait for the request to be processed. 4. From a computer within the registered network connect to the SISSDEN sensor using an SSH client, attempt to log in several times simulating a brute force attack. 5. Disconnect. Monitor the registered e-mail account until remediation report arrives, up to 24 hours.		
Exp. output	A remediation report is delivered within 24 hours of the simulated attack to the email account used during registration. The report contains accurate data on the attack.		
Privacy: YES	The test involves PII directly unless fake user data are used. Remediation report will likely contain other data, including potentially PII, from the same time window, not directly related to the test.		

5.2 Interface tests

Test ID	CUSTPORT1	**Spec level:** DESC	**Version:** 0.1
Test Name	Verify whether a user can sign up for an account at the Customer Portal		
Coverage	**Public interfaces:** 1: Customer Portal Account Sign Up		
Description	As an anonymous user visit the public component of the Customer Portal and click on a link to sign up for a free SISSDEN account. Provide valid credentials (including a valid test e-mail address). Check if confirmation e-mail arrives. Confirm e-mail. Verify that the user can log in successfully to the Customer Portal.		
Prerequisites	• Running version of the Customer Portal.		
Privacy: YES	Verify that customer data is protected in the Customer Portal		

Test ID	CUSTFBS1	**Spec level:** DESC	**Version:** 0.1
Test Name	Verify if validated customer can leave feedback		
Coverage	**Public interfaces:** 3: Customer Feedback System		
Description	Create a test account in the Customer Feedback System as a validated customer. As a validated customer use system to leave feedback regarding a test issue. Confirm using a SISSDEN operator account that system brings the issue to the attention of an appropriate member of staff.		
Prerequisites	• Customer Feedback System. • Privileges in the system to create test accounts. • Mechanism for validating customers. • SISSDEN operator account access.		
Privacy: YES	Verify that protection of the communications between the customer and the system is in place and feedback is confidential.		

Test ID	INTF5	**Spec level:** INFO	**Version:** 0.1
Test Name	Verify if the anonymous users can view public information about SISSDEN		
Coverage	**Public interfaces:** 5: View Public Information about SISSDEN		
Description	Verify if an anonymous user can view and navigate the public SISSDEN website. He/she can choose to enable or disable cookies and tracking. Verify if the website is browsed via HTTPS/SSL and if the statistics to determine the number and type of users accessing the website are logged.		
Prerequisites	• Running version of the public website.		
Privacy: YES	Cookie and tracking behaviour are clearly privacy-related. Most importantly, tracking users with third parties should not be in use.		

Test ID	INTF6	**Spec level:** INFO	**Version:** 0.1
Test Name	Verify if the dedicated users can subscribe/unsubscribe to news about SISSDEN results		
Coverage	**Public interfaces:** 6: Subscribe/unsubscribe to news about SISSDEN		
Description	SISSDEN results include notification about new findings, datasets, analytics. These are made available to users with a SISSDEN account that subscribe to the service. Verify if a user can login to the Customer Portal that allows signing up to SISSDEN news service consisting of periodic or on the fly email notifications containing important information on the project's results. Verify if the user receives an email to confirm his/her subscription or cancel it. Verify also the function allowing the user to parameterize the service so that he/she can receive news on the fly or combined in one periodic email (daily, weekly or monthly).		
Prerequisites	• Running version of the Customer Portal. • A user with valid SISSDEN account.		
Privacy: YES	Verify if the user's PII (e-mail, etc.) is protected from open access.		

Test ID	CUSTPORT2	**Spec level:** INFO	**Version:** 0.1
Test Name	Verify if it is possible to update SISSDEN User account information		
Coverage	**Public interfaces:** 7: Manage SISSDEN User account information		
Description	Log in to test SISSDEN User account. Alter User preferences. Verify if changes persist by logging out and logging in again.		
Prerequisites	• Running version of the Customer Portal. • A test SISSDEN User account exists.		
Privacy: YES	Verify that customer data is protected in the Customer Portal		

Test ID	METRCDSH	**Spec level:** ROUGH	**Version:** 1.0
Test Name	Verify output of metrics dashboard		
Coverage	**Public interfaces:** 8: View high-level aggregated metrics; 9: View more-detailed high-level aggregated metrics		
Description	Ensures the correct data passes through the workflow of the metrics components. The configuration of the metrics to be generated is specified in component 60 Metrics Management, which are then generated in component 26 Metrics Generation and Aggregation. The metrics displayed on the website, by component 54 Metrics Dashboard, should be consistent with this configuration.		
Prerequisites	Access to the Metrics Dashboard, Metrics Management configuration files, and backend database.		
Input	TBD		
Scenario	1. View the configuration files of the Metrics Management component. 2. View the metrics dashboard. 3. For each page of the metrics dashboard, modify the element of each form element (such as drop-down menus, text fields, etc) in order to vary the current view. 4. For each output of each page, compare with the data from the backend. 5. Verify that that the configuration files have been interpreted and applied correctly to the backend data in order to produce the correct output on the metrics dashboard.		
Exp. output	Correct and expected data displayed on the metrics dashboard.		
Privacy: NO	The metric dashboard is not expected to handle any personal identifiable information.		

Test ID	CUSTPORT3	**Spec level:** INFO	**Version:** 0.1
Test Name	Verify if it is possible to update SISSDEN Data Privacy Settings		
Coverage	**Public interfaces:** 10: View and change data privacy settings		
Description	Log in to test SISSDEN Vetted User account. Alter User privacy settings. Verify if changes persist by logging out and logging in again.		
Prerequisites	• Running version of the Customer Portal. • A test SISSDEN Vetted User account exists.		
Privacy: YES	Test obviously focusing on privacy aspects.		

Test ID	CUSTPORT4	**Spec level:** INFO	**Version:** 0.1
Test Name	Verify if it is possible to update SISSDEN Opt-in/Opt-out status		
Coverage	**Public interfaces:** 11: View and change SISSDEN service opt-in/opt-out status		
Description	Log in to test SISSDEN Vetted User account. Alter User opt-in/opt-out settings for networks owned by the User. Verify if changes persist by logging out and logging in again.		
Prerequisites	• Running version of the Customer Portal. • A test SISSDEN Vetted User account exists.		
Privacy: YES	Test obviously focusing on privacy aspects.		

Test ID	INTF7	**Spec level:** ROUGH	**Version:** 1.0
Test Name	Sign up and request access to the SISSDEN curated reference data set		
Coverage	**Public interfaces:** 12: Sign up and request access to the SISSDEN curated reference data set		
Description	Ensures that signing up and requesting access to the SISSDEN curated reference data set works as intended.		
Prerequisites	Running version of the Customer Portal. A test SISSDEN User account exists.		
Scenario	1. Log in to test SISSDEN User account. 2. Navigate to the page that allows requests for access to curated reference data set. 3. Complete the request form. 4. Validate the request. 5. Check for email from SISSDEN confirming that access has been granted. 6. Attempt to access the curated reference data set.		
Exp. output	Access to the curated reference data set.		
Privacy: YES	Unless fake user data is used for the test.		

Test ID	CRDSM	**Spec level:** ROUGH	**Version:** 1.0
Test Name	Verify if the expected data is present in the curated reference data set		
Coverage	**Public interfaces:** 13: Successfully vetted researchers access the SISSDEN curated reference data set		
Description	Verifies that the correct data is present in the curated reference data set. Firstly, that no data is present in the data set that should not be (including ensuring that data containing Personal Identifiable Information is not present, according to requirements). Secondly, that no data is missing from the data set that should be present.		
Prerequisites	Access to the curated reference data set and to the backend database		
Input	TBD		
Scenario	1. Access and download the curated reference data set. 2. Ensure that all data appearing in the data set is expected. 3. Ensure that all data in the data set has the appropriate levels of PII removed. 4. Access the backend database and ensure that both sets of data are consistent (i.e. all data in the backend that is expected to be in the curated reference data set actually is).		
Exp. output	A correctly formed curated reference data set.		
Privacy: MAYBE	Ensure that all data in the data set has the appropriate levels of PII removed.		

6 Coverage map

As explained in chapter 2, the only type of project artefacts made public so far are the public interfaces. The mappings for other artefact types are provided in deliverable D6.8 [6].

6.1 Public interface coverage

The following table presents the full list of public interfaces, as specified in D4.1 [4] and lists the identifiers of tests which cover each of the interfaces. The Stage column refers to the SISSDEN development stage at which a given interface is expected to be operational. Note that the interfaces may be further developed, necessitating updates to the corresponding tests.

Interface No.	Title	Stage	Core tests
1	Customer Portal Account Sign Up	MS	CUSTPORT1
2	Free Remediation Report Sign Up	FS	FREM1
3	Customer Feedback System	FS	CUSTFBS1
4	Shadowserver Reporting System	FS	FREM1
5	View Public Information about SISSDEN	FS	INTF5
6	Subscribe/unsubscribe to news about SISSDEN	MS	INTF6
7	Manage SISSDEN User account information	MS	CUSTPORT2
8	View high-level aggregated metrics	MS	METRCDSH
9	View more-detailed high-level aggregated metrics	MS	METRCDSH
10	View and change data privacy settings	MS	CUSTPORT3
11	View and change SISSDEN service opt-in/opt-out status	MS	CUSTPORT4
12	Sign up and request access to the SISSDEN curated reference data set	MS	INTF7
13	Successfully vetted researchers access the SISSDEN curated reference data set	MS	CRDSM

As shown in the above table, all public interfaces are testable and have been covered by the proposed tests. Most of the interfaces will be delivered and tested much later in the project. Four interfaces have already been made available as part of the initial platform (First System stage) and will be formally tested in the First Verification phase. However, since the test FREM1 has been used as proof of successful activation of the core functionality of the platform, interfaces 2 and 4 can already be considered as verified – refer to deliverable D6.3 [5] for test results.

7 Summary

The document presents the testing approach taken in the SISSDEN project, explaining the evolution of the testing methodology during the project. The goal is to gradually formalize the testing process in order to:

- Provide a lightweight but verifiable trial approach in the early phases of development, allowing the tests to evolve in parallel with design changes and implementation progress,
- Ensure ability to avoid regressions through well-defined verification methods for mature functionality in later stages of development,
- Cover all delivered and testable project artefacts at the end of the project as means of final verification of the delivered product.
- Ensure that privacy and ethics aspects are identified during the early development and testing phases.

Finally, the document provides test specifications (to the level expected at First System stage) and their mapping to project artefacts. This information is restricted to the public information – high-level tests of overall functionality and coverage of public interfaces. For full specification refer to the confidential deliverable D6.8 [6], created and delivered in parallel with this document.

Bibliography

[1] SISSDEN consortium. Project SISSDEN - Description of Action. Call H2020-DS-2015-1 proposal n° 700176, Feb. 2016.
[2] SISSDEN consortium. Deliverable D3.1 “Use cases and requirements”, (confidential) Nov. 2016.
[3] SISSDEN consortium. Deliverable D3.3 “Initial technical architecture”, (confidential) Jan. 2017.
[4] SISSDEN consortium. Deliverable D4.1 “Data exchange interfaces specification”, Feb. 2017.
[5] SISSDEN consortium. Deliverable D6.3 “Activation of platform pilot”, (confidential), Apr. 2017.
[6] SISSDEN consortium. Deliverable D6.8 “Trial definition and test plan – non-public”, (confidential), Jun. 2017.