## Deliverable D5.2

### Data Sensitivity Analysis Tool

**Editor(s):** Sigal Asaf, Danielle Cohen, Micha Moffie, Muhammad Barham  
**Responsible Partner:** IBM  
**Status-Version:** V1.0  
**Date:** 21/12/2017  
**Distribution level (CO, PU):** CO
<table>
<thead>
<tr>
<th>Project Number:</th>
<th>GA 727301</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Title:</td>
<td>SHIELD</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title of Deliverable:</th>
<th>Data Sensitivity Analysis Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Due Date of Delivery to the EC:</td>
<td>December 31st 2018</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workpackage responsible for the Deliverable:</th>
<th>WP5</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Editor(s):</th>
<th>Sigal Asaf, Danielle Cohen, Micha Moffie, Muhammad Barham</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Contributor(s):</th>
<th>IBM, Tecnalia</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Reviewer(s):</th>
<th>Jason Xabier Mansell, Xabier Larrucea</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Approved by:</th>
<th>All partners</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Recommended/mandatory readers:</th>
<th>All WP’s</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Abstract:</th>
<th>This report lays the ground work for the development and integration of the data sensitivity analysis tool (DSA tool) within the SHIELD framework. The report reviews the most relevant state of the art, presents the main use case requirements for the DSA tool, and provides its high-level design.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Keyword List:</th>
<th>data privacy, sensitive data, sensitive attribute</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Disclaimer</th>
<th>This document reflects only the author’s views and neither Agency nor the Commission are responsible for any use that may be made of the information contained therein</th>
</tr>
</thead>
</table>
**Document Revision History**

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modifications Introduced</th>
<th>Modification Reason</th>
<th>Modified by</th>
</tr>
</thead>
<tbody>
<tr>
<td>v0.1</td>
<td>01/08/2017</td>
<td>First draft ToC version</td>
<td>Micha Moffie, Muhammad Barham (IBM)</td>
<td></td>
</tr>
<tr>
<td>V0.2</td>
<td>25/10/2017</td>
<td>Updated ToC, draft content</td>
<td>Micha Moffie (IBM)</td>
<td></td>
</tr>
<tr>
<td>V0.3</td>
<td>26/10/2017</td>
<td>Updated State of the Art</td>
<td>Danielle Cohen, Micha Moffie (IBM)</td>
<td></td>
</tr>
<tr>
<td>V0.4</td>
<td>01/11/2017</td>
<td>Updated Requirements</td>
<td>Micha Moffie (IBM)</td>
<td></td>
</tr>
<tr>
<td>V0.5</td>
<td>16/11/2017</td>
<td>Updated State of the Art, Updated design</td>
<td>Sigal Assaf, Danielle Cohen (IBM)</td>
<td></td>
</tr>
<tr>
<td>V0.7</td>
<td>29/11/2017</td>
<td>Updated State of the Art – Sensitivity analysis of unstructured documents, incorporate review remarks.</td>
<td>Sigal Assaf, Danielle Cohen (IBM)</td>
<td></td>
</tr>
<tr>
<td>V0.8</td>
<td>05/12/2017</td>
<td>Added introduction in Sensitive data definition and Existing products Sections</td>
<td>Sigal Assaf (IBM)</td>
<td></td>
</tr>
<tr>
<td>V0.9</td>
<td>13/12/2017</td>
<td>Updated Integration section, executive summary, section 2.1.4 sensitive data within source code</td>
<td>Jason Xabier Mansell, Xabier Larrucea (Tecnalia)</td>
<td></td>
</tr>
<tr>
<td>V1.0</td>
<td>21/12/2017</td>
<td>Finalized version. Incorporated Tecnalia review.</td>
<td>Sigal Asaf (IBM)</td>
<td></td>
</tr>
</tbody>
</table>
Table of Contents

Table of Contents ........................................................................................................... 4
List of Figures .................................................................................................................. 4
List of Tables ................................................................................................................... 5
Terms and abbreviations ................................................................................................... 6
Executive Summary .......................................................................................................... 7
  1 Introduction ................................................................................................................. 8
    1.1 Objective and Scope ............................................................................................... 8
    1.2 Dependencies and Document Structure .................................................................. 8
  2 State of the art ............................................................................................................... 8
    2.1 Academia ................................................................................................................. 8
      2.1.1 Sensitive Data Definition .................................................................................. 8
      2.1.2 Sensitivity Analysis of Relational Databases ...................................................... 9
      2.1.3 Sensitivity Analysis of Unstructured Documents .............................................. 11
      2.1.4 Sensitive Data within Source Code .................................................................... 12
    2.2 Existing Products .................................................................................................... 12
    2.3 Patent ..................................................................................................................... 13
  3 Use Case Requirements ................................................................................................. 13
  4 Data Sensitivity Analysis Tool Design ......................................................................... 16
    4.1 Data Store Package ............................................................................................... 16
      4.1.1 Interfaces .......................................................................................................... 17
      4.1.2 Implementation for Relational Database .......................................................... 17
    4.2 Sensitivity Classification Package ......................................................................... 17
    4.3 Heuristics Package .................................................................................................. 18
  5 Integration .................................................................................................................... 19
  6 Summary and next steps ............................................................................................... 20
  7 References .................................................................................................................... 21

List of Figures

FIGURE 1: FPSCR TABLE ............................................................................................. 14
FIGURE 2: DSA FLOW ................................................................................................. 16
FIGURE 3: CLASSIFICATION AND DATASTORE CLASS DIAGRAM ..................... 18
FIGURE 4: SENSITIVITY HEURISTIC INTERFACE ...................................................... 18
FIGURE 5: DATA SENSITIVITY TOOLS ARE USED DURING THE ANALYSIS, DESIGN AND DEPLOYMENT PHASES ........................................................................................................ 19
FIGURE 6: DATA SENSITIVITY WITHIN THE OVERALL PICTURE ............................ 20
List of Tables

DATA SENSITIVITY ANALYSIS TOOL - REQUIREMENTS .......................................................... 15
TABLE 2: DATA SENSITIVITY ANALYSIS TOOL - REQUIREMENTS .......................................... 15
TABLE 3: DATASTORE INTERFACES ......................................................................................... 17
TABLE 4: CLASSIFICATION PACKAGE CLASSES ............................................................... 17
TABLE 5: DATACLASSIFICATION CLASS FIELDS ....................................................................... 18
## Terms and abbreviations

<table>
<thead>
<tr>
<th>AB</th>
<th>Advisory Board</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>Consortium Agreement</td>
</tr>
<tr>
<td>DB</td>
<td>Database</td>
</tr>
<tr>
<td>DSA</td>
<td>Data Sensitivity Analysis</td>
</tr>
<tr>
<td>IC</td>
<td>Information Content</td>
</tr>
<tr>
<td>NP</td>
<td>Noun Phrases</td>
</tr>
<tr>
<td>WSE</td>
<td>Web Search Engine</td>
</tr>
</tbody>
</table>
Executive Summary

This document represents deliverable D5.2 - “data sensitivity analysis tools (requirements and design)“ which is related to data sensitivity tools and APIs. This first release is focused on a requirements analysis and technical design. The report lays the ground work for the development and integration of the data sensitivity analysis tool (DSA tool) within the SHIELD framework. The report reviews the most relevant state of the art, presents the main use case requirements for the DSA tool, and provides its high-level design.

This first release is enhanced by the following version of D5.2 “Data sensitivity analysis tools (prototype)” which provides a first prototype to be used as support to WP6. This second release will consider further requirements stemming from this initial evaluation of the architecture and functionalities.
1 Introduction

1.1 Objective and Scope

This deliverable is the first deliverable in Task 5.2: Data Sensitivity analysis tools. It describes the requirements analysis and technical design of the DSA tool. In addition, it provides the state of the art related to data sensitivity – definition, auto detection of sensitive data in relational databases, and auto detection of sensitive data in unstructured free text documents.

The intended audiences for this document are mainly designers and developers – those that will design and build the data sensitivity analysis tool. Service maintainers (users of the tool) may find it useful to understand how the tool is constructed and how it can be configured. In addition, the team responsible for WP4 tasks (Privacy by design models and tools) will find here a good reference for data sensitivity definition, and a generic description of the output expected from the data sensitivity tool.

1.2 Dependencies and Document Structure

The tool designed will provide indication on data sensitivity in relational databases, and is expected to be used in WP4: Privacy by design. To provide high quality results in the domain of medical information, concrete experiments on representative medical information databases need to be undertaken. As stated in the DOW this project will not deal with real personal data. So in order to be able to undertake this experiments, SHiELD partners (FCSR, OSAKIDETZA, Lancashire NHS) will provide filter databases in which the information provided will be fake, in the sense that the data provided will contain fake patients, even though the patient records information may be real. In this way we will be able to process real data of the patient record but linked to fake identities.

This deliverable is organized as follows. Section 2 describes the state of the art that relates to sensitivity analysis. Section 2.3 presents the data sensitivity analysis tool’s main requirements. Section 4 describes the core of this work – the tool design. Section 5 discuss integration issues, and lastly Section 6 presents the next steps toward the development of data sensitivity analysis tool.

2 State of the art

In this section we review the state of the art that relates to sensitivity analysis. We start by providing a short definition of data sensitivity as presented in related papers. Then, we focus on the analysis related to databases(DBs), and finally we provide some information on the methods used to search for sensitive data in free text documents.

2.1 Academia

2.1.1 Sensitive Data Definition

To better understand Sensitive Data definition, various definitions from related papers has been reviewed. Based on this investigation the tool output (i.e., what data should be considered as sensitive) which best meets the DSA Tool requirements will be refined.

There are many definitions of sensitive data which depends mainly on the type of data store (e.g., relational database, key-value database, or unstructured document store), as well as the techniques used to identify sensitive data.
Mouza et.al [1] defines the set of sensitive attributes ($S_s$) in relational databases using two other definitions. (1) Confidential attributes set (denoted as $S_c$), where an attribute is considered confidential if it contains a confidential instance(s), regardless to the number of the instance’s occurrences. (2) Identify attributes set (denoted as $S_i$), where an attribute is considered identify attribute if its instances occurs less then k times. Then the set of sensitive attributes is defined to the union set of these sets. ($S_s = S_c \cup S_i$).

Kamakshi et.al [2] uses weights to evaluate the sensitivity of attributes in PPDM (Privacy-Preserving Data Publishing, as described in [3]). The weights are assigned to attributes, based on two distinctions (1) Whether a single attribute can identify an individual. (2) Whether a group of attributes can indirectly reveal the identity of an individual.

Sanchez et.al [4] [5] presents method to detect sensitive information from textual documents. Though the method is not related to databases, it can be modified to be relevant for attributes’ names or instances. They consider sensitive information as information that can provide more information than common ones, or terms that increases the information that an attacker will gain if they poses the document.

Information Content (IC) of a term is defined to be the amount of information the term provides. IC of term is the inverse probability to appear in a corpus $p(t)$. Therefore, rare terms will have higher IC.

\[(1) \quad IC(t) = -\log_2 p(t)\]

This definition is extended to extract the appearance’s probability not from a single corpus but from the web. Where page_counts is the number provided by WSE (Web Search Engine) when querying $t$, and total_webs is the total amount of web sites indexed by the search engine.

\[(2) \quad IC_{\text{web}}(t) = -\log_2 p_{\text{web}}(t) = -\log_2 \frac{\text{page_counts}(t)}{\text{total_webs}}\]

The sensitivity detection is based on the IC of Noun Phrases (NPs) that do not contain delimiters r stop words.

\[(3) \quad Sensitive\_NPs = \{ NP_i \in d \mid IC_{\text{web}}(NP_i) \geq \beta \}\]

Where $d$ is the analyzed document, and $\beta$ is a threshold. A noun phrase $NP_i$ is considered sensitive if its $IC_{\text{web}}$ is bigger or equal than $\beta$.

Abril et.al [6] defines private entity in unstructured document as an entity that reveals information about the document’s correspondents. He divides the private entities into two categories: (1) identifier - entity that directly identifies correspondent, (2) quasi-identifier - entity that can be linked with contextual information and external knowledge to identify a correspondent.

### 2.1.2 Sensitivity Analysis of Relational Databases

An extensive search has been undertaken and little work directly related to sensitivity analysis in DBs has been identified. The most relevant work found was done by Mouza et.al [1] [7]. In their paper Towards an Automatic Detection of Sensitive Information in a Database, they describe a rule system approach to detect sensitive information. The initial rules are determined by human experts, and then based on these rules, other rules are constructed automatically. The rules are divided into intentional rules which searches for conditions on database schema.
(mainly the attribute’s name) and extensional rules which are based on the attribute’s values (attribute instances).

Finding attributes that match the rules is not sufficient, since there may be links between attributes in the database. Therefore, a propagation process is offered. Two types of links are defined (1) Integrity referential links, which are stored in the database scheme, (2) Semantical links (E.g. an attribute in a table may have the same semantics as another one in another table). The semantical links are found by NLP tools.

Overall, the attributes are sorted by their sensitivity score which is derived from the matching rules and the links. Then attributes with a score above the threshold that was set by the user, are considered sensitive.

Table 1 summarizes the different rules proposed on the paper. In the table, the rules are classified in two types: (1) Regular rules which are independent of other rules’ results; (2) Propagation rules which take as an input other rules’ results.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Type</th>
<th>Intuition</th>
<th>Input</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Look for terms from the expert dictionary</td>
<td>Regular</td>
<td>Experts know terms that may indicate a content is sensitive</td>
<td>- Expert terms - Database schema - Database data</td>
<td>- “street” in attribute name/instances may refer to employee address. - “euro” or € in attribute name/instances may refer to private data on salary.</td>
</tr>
<tr>
<td>Search for terms similar to the ones in the expert dictionary.</td>
<td>Regular</td>
<td>If a term A is similar to term B, and term A is considered sensitive by the expert, term B is also sensitive.</td>
<td>- List of terms - APPROX function - Database schema - Database data</td>
<td>Term A = “salary” Term B = “<em>SALARY</em>” If A is sensitive, so is B.</td>
</tr>
<tr>
<td>If an attribute is a primary/secondary key in the database.</td>
<td>Regular</td>
<td>The scheme may indicate the keys of the database that can be sensitive</td>
<td>- Database schema</td>
<td>The attribute “ID” is a key in the database therefore it is sensitive.</td>
</tr>
<tr>
<td>If the attribute’s instances contain instances of sensitive attribute.</td>
<td>Propagation</td>
<td>If attribute A is considered sensitive, and its instances appears in attribute B instances, Attribute B may be also sensitive</td>
<td>- Scores of the attribute so far - Database schema - Database data</td>
<td>Attribute A = Home address Attribute B = Shipment address There might be a shipment address that also appears as address of an individual in the database.</td>
</tr>
</tbody>
</table>
If a foreign key attribute references a primary/secondary key that are sensitive, the foreign key attribute is sensitive too.

<table>
<thead>
<tr>
<th>Propagation, Integrity referential links</th>
<th>A key may appear as instance of another attribute. This attribute should be considered sensitive as well.</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Scores of the attribute so far</td>
<td>- Database schema</td>
</tr>
<tr>
<td>Attribute “Name” is a primary key in table A.</td>
<td>Attribute “Manager Name” is a foreign key in table B, and it refers to attribute “Name” in table A.</td>
</tr>
</tbody>
</table>

Propagate sensitive score using semantical links. NLP tools can detect semantical resemblance between names/instances.

<table>
<thead>
<tr>
<th>Propagation, Semantical links</th>
<th>Attribute/instance may not be in the expert rules, but it has the same semantic of another attribute/term that is considered sensitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Scores of the attribute so far</td>
<td>- Database scheme</td>
</tr>
<tr>
<td>- Database data</td>
<td>Address-&gt;street, city. Salary-&gt;wage, income, bonus</td>
</tr>
</tbody>
</table>

| Table 1: Rules for automatic detection of sensitive information from [1] [7] |

2.1.3 Sensitivity Analysis of Unstructured Documents

Several papers have tried to tackle the problem of detecting sensitive information in unstructured documents.

Douglass et.al [8] presents an automated method for removing protected health information (PHI) from free-text nursing notes. The PHI list, defined by the Health Information Portability and Accountability Act (HIPAA), includes identifiers such as: names, all geographic subdivisions smaller than a state, all elements of dates (except year) for dates directly related to an individual, all ages over 89, telephone numbers, social security numbers, and medical record numbers. To detect these terms, the authors use two main methods: 1) de-identification algorithms based on pattern-matching. The algorithm detects terms that contain numeric tokens, and uses regular expression to classify those terms; 2) Search for names defined in existing lists such as 1990 U.S. Census, or list of all hospital locations. In addition, it applies approximate matching to catch incorrectly spelling or nicknames/shortcuts. The methods described by Douglass et.al are very efficient but are not scalable since not all sensitive entities in a document can be recognized by regular expression, and requires ad-hoc knowledge bases related to specific areas.

Cumby et.al [9] presents a semi-automatic sensitivity data detection and obstruction tool for Microsoft Word documents. It targets two types of sensitive information: 1) Client Identifying Information (CII) – information which includes words and phrases that reveal what client company the document pertains to; 2) Personally Identifying Information (PII) –Information which includes any person identifier like person names, location, social security numbers, etc. The authors present a system which recognizes the CII and PII terms and phrases given the text of a document. To detect the CII terms, it uses a trained classifier. For detecting PII terms, the system uses a statistical NLP tools such as Stanford Named Entity recognizer in addition to simple template based recognizers to identify the PII terms and phrases. The use of trained classifier
may hamper the generality of the method, in addition to requiring a manual labeled document store.

As reviewed in section 2.1.1, Sanchez et.al [4] [5] detects sensitive phrases using IC calculation. Phrase with IC values above $\beta$ threshold is considered sensitive. This method provides a generic solution since it uses the web as its corpora.

Abril et.al [6] proposed to use data extraction tools to identify private entities in a document. They proposed to detect the document name entities (NE), which refers to proper names, locations, or organization. Then any name entity is considered private entity ($PE$), since they claim that the set of name entities in a document ($NE$) and the set of private entities in a document ($PE$), are usually the same set ($NE \approx PE$). In order to extract the name entities, they use Named Entity Recognition and Classification systems (NERC) [10]. In order to validate this approach, they present an experiment where $PE$ set recognized by humans is compared to the $NE$ set generated with NERC system. Results show that in a document that contains 3092 words, only 9 words were classified as $PE$ and were not classified as $NE$ by the NERC system which is 0.29% from total.

2.1.4 Sensitive Data within Source Code

An abstract syntax tree (AST) is a tree representation of the abstract syntactic structure of source code written in a programming language [10]. These ASTs are basic structures used widely in order to analyze source code such as [11] where authors describe a novel Semantic Web enabled global source code analysis blending crowdsourcing and linked data. Literature reflects a huge amount of approaches for analyzing source code. Some of them are focused on automatically classifying source code [10] and some of them use heuristics [12] such as clustering algorithms [13] to identify patterns. One of the most relevant activity in this area is the design pattern recovery [14] which should be used to identify commonalities or specific structures. Patterns reduce the number of deficiencies which are critical for software quality including security vulnerabilities [15] This kind of approaches for analyzing source codes can be used not only for identifying specific structures but also to detect plagiarisms [16] or even improving secure coding [17]. In this sense we want to highlight areas of source code which are close to data manipulation.

2.2 Existing Products

We searched for existing products which provide solutions for sensitive information detection.

All the tools identified provide solutions for document classification. In document classification, the system assigns a document to one or more classes or categories. Labelling a document is done either by the user (User driven document classification), or in automatic manner using techniques such as pattern matching, dictionaries of sensitive words, and keyword matching.

While automatic document classification techniques such as pattern matching may be relevant for the DSA Tool, the focus in SHILED is in searching sensitive data in relational database. As part of the study, no tool which targets this specific domain has been identified.

The following is a list of the existing classification products identified.

1. Boldon James classifier [19]: it’s user-driven classifier, meaning, the user can assign labels (metadata) to the data they create/meet. Then, the enforcement depends on defined policies and the labels. They have several classifiers, each one is address different use case (email classifier, file classifier, etc.)
2. Symantec information centric tagging [20]: the solution is similar to [19] solution. Namely, the user has to identify the sensitive data and label it (user driven data classification).

3. Microsoft Data Classification Toolkit [21]: the purpose of the tool is to allow organization to identify, classify and protect the 'file servers' that run Microsoft servers. The toolkit allows the users to identify applicable IT GRC sensitive documents, and defining classification and protection policies.

4. Titus classification [22]: TITUS Classification identifies sensitive information by applying classification labels and visual markings to email and documents. 'user-driven classification'.

5. Digital guardian data classification [23]: No functional explanation is available. However, the following was written in the website "identifying sensitive data such as electronic health records (as required by HIPAA), cardholder data (as required by PCI), confidential or proprietary design documents (as required by ITAR), and other structured and unstructured data. With automated data discovery and classification". It has been assumed that digital guardian product uses heuristics to identify predefined formats.

6. Spirion [24]: Spirion’s data classification tool is a user-driven, as we mentioned before. This kind of tool allow the user to label (classify) the data. The tool will alert when the data shared or sent via mail or any other supported mechanism.

7. Varonis [25]: The Varonis’s data classification framework uses regular expression, dictionary searches, and unique metadata (created by Varonis’s metadata framework) to classify data. In addition, the framework supports integration with third party content classification products.

8. Checkpoint MultiSpect [26]: CheckPoint MultiSpect engine has 3 layers for classification, and it uses many options such as: pattern, dictionaries and keyword matching, similarity to commonly-used templates, file attribute-based matching and more. The product is not marketed as a stand-alone, but rather part of Checkpoint’s Blade.

2.3 Patent

Not many patents that aim at identifying sensitive information in databases have been identified. US 8856157 B2 - Automatic detection of columns to be obfuscated in database schemas patent has been identified as more relevant. This patent deals with Static detection of sensitive data by application type, static detection by column name, and data profiling by examining a sample of the data.

3 Use Case Requirements

This section collects the main requirements related to data analysis. Much of this work is based on Deliverable D6.1 - use case specification [27] as well as discussions with the use case providers.

One of the most pressing concerns expressed by the use case owners is related to the upcoming GDPR regulation. Specifically, GDPR regulation requires special care be taken of personal and sensitive information (e.g. consent, data hiding, security etc.). However, enforcing and adhering...
to such regulation requires first to accurately identify and locate all personal and sensitive information across organizations.

Discussion with IT professionals (e.g. FCSR) paint a complex picture where organizations manage a myriad of systems to record and store data. These systems include relational data bases (e.g. multiple versions of Oracle and MS SQL databases in FCSR), NoSQL databases, File systems, Document repositories and file hosting services such as drop box or one drive and much more. The data itself, can be structured, semi-structured or not structured at all. And, as expected the documents are of different formats – ranging from images (e.g. Dicom) to pdfs, text, json, html etc. Lastly, the amount of data stored by organization can easily reach Terabytes.

SHiELD goal in this work is ambitious, but in a first step, the target is to identify personal and sensitive information in SQL type databases. The reason for this is three-fold: (1) As described in section 2 - there is not much work that has been done in this area (2) SQL DB have a structure and semantics (e.g. schema) that can be used to improve the accuracy of the tool and (3) This would be a stepping-stone to other types of systems (including semi structured and not structured) and could provide meaningful insights – both in terms of methodology developed as well as practical information (e.g. data items may reside in multiple locations – identifying them first in DB may improve accuracy).

Figure 1 is an example of one of the tables provided by FCSR. This example illustrated the structure and semantics available in DB. During the analysis, the tool can make use of the repeated instances of data (multiple values of the same type) as well as the meta data associated with each column (e.g. column name)

![Figure 1: FPSCR Table](image)

Further discussion with the use case owners to clarify the specific requirements for identifying sensitive and personal information in DB is ongoing. The following table collects the high-level description of the critical requirements identified so as to the date this deliverable was finalised.
<table>
<thead>
<tr>
<th>#</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Identify sensitive data</td>
</tr>
<tr>
<td>2</td>
<td>Configuration Driven</td>
</tr>
<tr>
<td>3</td>
<td>Multiple DB types / versions</td>
</tr>
<tr>
<td>4</td>
<td>DB data types</td>
</tr>
<tr>
<td>5</td>
<td>Read only</td>
</tr>
<tr>
<td>6</td>
<td>Big Data</td>
</tr>
</tbody>
</table>

**Table 2: Data Sensitivity Analysis Tool - Requirements**
4 Data Sensitivity Analysis Tool Design

As described above, our first step in this task is to identify personal and sensitive information in SQL type databases. We will build on the work developed [1] [7] where different rules and heuristics have been proposed (see Table 1). Our plan in this task is to expand on the rules and propose new ones. These rules will be based and implemented on top of the generic framework that is defined here. Next, we provide the tool overall design, based on the requirements defined above.

The architecture contains three main components:

1. Data Store package - A package which represents the data store and is responsible for handling the tool input
3. Sensitive Heuristics package – Contains classification algorithms/heuristics – A set of heuristics which get as input a data store, analyses the input and outputs the data classification.

Figure 2 displays the tool’s data flow. From data stored in a database to the final sensitivity classification output. The figure displays a case of relational database, but can easily updated to any other data store. The Data Explorer library enables exploring the data and the schema of a relational database. The final output contains the sensitivity results in a machine-readable format (e.g., json format).

Figure 2: DSA flow

4.1 Data Store Package

The dataStore package provides the required functionality to handle the data. It includes the interfaces as well as the implementation required to support various type of data stores. As part of SHiELD, we plan to provide an implementation for relational database. However, the data
store package may include additional data stores such as key-value database, or a set of free text documents.

4.1.1 Interfaces

Table 3 describes the interfaces included in the dataStore package.

<table>
<thead>
<tr>
<th>Interface Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataIdentifier</td>
<td>A pointer to a specific data in the data store for which a classification information is provided. E.g. a column in the table.</td>
</tr>
<tr>
<td>DataStore</td>
<td>The data store. Provides an iterator over the data store’s DataIdentifier entities.</td>
</tr>
</tbody>
</table>

Table 3: DataStore Interfaces

4.1.2 Implementation for Relational Database

The classes RelationalDBDataIdentifier and RelationalDatabase implement relational database’s DataIdentifier and DataStore interfaces.

- A RelationalDBDataIdentifier instance represents a relational db column.
- A RelationalDatabase interface represents a relational db. Its constructor gets as input the information required to connect to the specific database. It enables iterating over the database’s columns. It also provides additional functionality related to relational database such as table iterator, or getting a column’s meta-data (e.g., the column type, an indicator if the column is a primary or foreign key, etc.).

The relational database implementation uses as its core the DataExplore tool which was developed by IBM’s Haifa Research Lab. The DataExplore library supports connecting to various type of relational databases, and enables explore the database data and schema. It provides API for exploring the database’s metadata as well as getting the data itself.

4.2 Sensitivity Classification Package

The classification package contains the classes which represents a data store classification. For each data identifier, a classification is attached. Table 4: Classification package classes describes the package’s classes, and Table 5 provides additional information on the DataClassification class. The classification package will enable output its data in a machine-readable format.

Figure 3 displays the classification package, the datastore package, and the relations between these two packages.

<table>
<thead>
<tr>
<th>Class Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataClassification</td>
<td>Describes the classification of a specific dataIdentifier. For detailed description, see Table 5.</td>
</tr>
<tr>
<td>DataStoreClassification</td>
<td>Describes the classification of an entire data store. Consists from a map from dataIdentifier to its classification.</td>
</tr>
</tbody>
</table>

Table 4: Classification package classes
### Table 5: DataClassification class fields

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sensitivityDegree</td>
<td>Defines the data’s sensitivity degree. How much information the specific piece of data reveals.</td>
</tr>
<tr>
<td>confidence</td>
<td>Defines how much the tool is confident in the sensitivity degree it provides.</td>
</tr>
<tr>
<td>categories</td>
<td>A list of categories the data belongs to (such as personal info, health, email). Each category is attached with its own confidence value.</td>
</tr>
</tbody>
</table>

**Figure 3: Classification and DataStore class diagram**

### 4.3 Heuristics Package

The heuristics package includes a set of heuristics which get as input a data store, and output the data store sensitivity. The sensitivity heuristics follow the interface defined by the SensitivityHeuristic class as illustrated in Figure 4.

```java
public interface SensitiveHeuristic {
    /**
     * Searches for sensitivity data
     * @param data - The data to explore
     * @param sensitivity - Previous sensitivity information gathers on the data.
     * @Return an updated data sensitivity
     */
    DataStoreClassification searchSensitivity(DataStore db, DataStoreClassification sensitivity)
        throws SensitivityAnalyzerException;
}
```

**Figure 4: SensitivieHeuristic interface**
The package includes generic heuristics which can be applied to any type of data store, and specific heuristics which are designated to a specific type of data store. For example, a heuristic which searches predefined terms (e.g., salary, SSN) has a generic nature versus a heuristic which recognizes primary key column in relational database.

In addition, the package will also include a generic composition heuristic. This heuristic is provided (through its constructor) with a list of other sensitive heuristics. It runs each of these heuristics, and combines their results to create its own classification output. For example, if two different heuristics A, and B recognize same dataIdentifier as sensitive, but heuristic A sets its sensitiveDegree to 80 while heuristic B sets it to 90, the composition heuristics should decide the proper sensitivityDegree value. For example, it can set the sensitivityDegree to the maximum value, or to the average.

5 Integration

Data sensitivity tools are basically used during the analysis, design and deployment phases (Figure 5) in order to identify which data requires a specific treatment. WP4 and WP5 have strong connections because we can identify vulnerabilities or sensitive information.

![Figure 5: Data sensitivity tools are used during the analysis, design and deployment phases](image)

For example, the data sensitivity analysis results would provide input to WP4 tasks – “Implementation of Security modelling tools”. Based on our initial discussions with WP4, the tool output will be used at design-time.
From the overall SHiELD picture (Figure 6) data sensitivity (highlighted in red) responsible on analysing each national database in order to identify weaknesses/vulnerabilities. The integration between these tools and the SHiELD architecture is going to be flexible. In case of interaction between them we will use a machine readable format (e.g., sensitive classification format will be JSON or XML formats.

![Figure 6: Data sensitivity within the overall picture](image)

6 Summary and next steps

This deliverable D5.2 “data sensitivity analysis tools (requirements and design)” outlines a state of the art related to the SHiELD requirements and needs stemming from the use cases described within WP6. Therefore, the state of the art, the use case requirements and the design are limited to use case descriptions and considerations. We can envisage further and more complex situations for data sensitivity such as data sensitivity during use cases’ executions or more complex databases.

This first release reports the ground work for the development and integration of the data sensitivity analysis tool (DSA tool) within the SHiELD framework. As result, this report reviews some works related to sensitivity analysis of relational databases and of unstructured documents. In addition, a high level design identifies the main functionalities supporting the use cases.
For the next step a prototype will be developed with several heuristics. This prototype will be evaluated on sample data and validate (1) the design and (2) the initial heuristics. Based on the results the required steps will be taken; (1) adapt/modify the design if needed and (2) update, improve or add heuristics.

7 References


Project Title: SHIELD Contract No. GA 727301

http://project-shield.eu/


