European Security in Health Data Exchange

Deliverable D4.2
Privacy by design models and tools: first full prototype plus documentation

<table>
<thead>
<tr>
<th>Editor(s):</th>
<th>Stefanie Wiegand, Mike Surridge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsible Partner:</td>
<td>IT Innovation</td>
</tr>
<tr>
<td>Status-Version:</td>
<td>Final</td>
</tr>
<tr>
<td>Date:</td>
<td>03/07/2018</td>
</tr>
<tr>
<td>Distribution level (CO, PU):</td>
<td>PU</td>
</tr>
</tbody>
</table>

Project Title: SHIELD          Contract No. GA 727301          http://project-shield.eu/
**Project Number:** GA 727301  
**Project Title:** SHiELD

<table>
<thead>
<tr>
<th><strong>Title of Deliverable:</strong></th>
<th>Privacy by design models and tools: proof of concept</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Due Date of Delivery to the EC:</strong></td>
<td>03/07/2018</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Work package responsible for the Deliverable:</strong></th>
<th>WP4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Editor(s):</strong></td>
<td>Stefanie Wiegand, Mike Surridge</td>
</tr>
<tr>
<td><strong>Contributor(s):</strong></td>
<td>Ken Meacham, Muhammad Barham, Chris Miles, Eleonora Ciceri, Borja Lopez Moreno, Eunate Arana Arri, Tony Schaffel, Xabier Larrucea Uriarte</td>
</tr>
<tr>
<td><strong>Reviewer(s):</strong></td>
<td>Ed Conley</td>
</tr>
<tr>
<td><strong>Approved by:</strong></td>
<td>All partners</td>
</tr>
<tr>
<td><strong>Recommended/mandatory readers:</strong></td>
<td>WP4 and WP6 partners, use case- and tool owners</td>
</tr>
</tbody>
</table>

**Abstract:**  
The software deliverable described in this report is the first coverage of regulatory compliance requirements, and includes (I) process-centric security threats and (II) countermeasures using data hiding methods during protracted operation of end-to-end data exchange scenarios. This deliverable is the result from Task 4.1 – Task 4.3.

**Keyword List:**  
Security Modelling Tools, Secure Knowledge Base, Secure Design Patterns, Privacy-by-design

**Disclaimer**  
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.
### Document Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modifications Introduced</th>
<th>Modified by</th>
</tr>
</thead>
<tbody>
<tr>
<td>v0.1</td>
<td>31/03/2018</td>
<td>First draft version</td>
<td>Stefanie Wiegand (IT Innovation)</td>
</tr>
<tr>
<td>v0.2</td>
<td>30/04/2018</td>
<td>Second draft version</td>
<td>Stefanie Wiegand (IT Innovation)</td>
</tr>
<tr>
<td>v0.3</td>
<td>14/05/2018</td>
<td>Third draft version</td>
<td>Stefanie Wiegand (IT Innovation)</td>
</tr>
<tr>
<td>v0.4</td>
<td>07/06/2018</td>
<td>Version for internal review</td>
<td>Stefanie Wiegand (IT Innovation)</td>
</tr>
<tr>
<td>v1.0</td>
<td>03/07/2018</td>
<td>Final version</td>
<td>Stefanie Wiegand (IT Innovation)</td>
</tr>
</tbody>
</table>
Table of Contents

Table of Contents ................................................................. 4
List of Figures ......................................................................... 6
List of Tables ........................................................................... 7
Terms and abbreviations ......................................................... 7
Executive Summary .................................................................. 8
1 Introduction ........................................................................... 9
2 Updated knowledge base ....................................................... 9
  2.1 Risk level evaluation ....................................................... 9
  2.2 Generic security threats ................................................ 11
  2.3 Compliance threats ....................................................... 12
    2.3.1 Italy ........................................................................ 12
      2.3.1.1 GD.x.HGenDHP.1 ................................................ 12
      2.3.1.2 GD.x.StHGenD.1 .............................................. 13
    2.3.2 Spain ...................................................................... 13
      2.3.2.1 PD.x.StHPersD.1 .............................................. 13
      2.3.2.2 PD.x.StHPersD.2 .............................................. 14
    2.3.3 GDPR ..................................................................... 14
  2.4 Threats addressed by WP5 tools ...................................... 15
    2.4.1 UC.IBM.1 – Data hiding tool .................................... 15
      2.4.1.1 Use case description .......................................... 15
      2.4.1.2 Identified threats ............................................. 16
        2.4.1.2.1 Lawfulness (PD.x.IBM.1.1) ............................ 16
    2.4.2 UC.Metrarc.1 .......................................................... 17
      2.4.2.1 D.C.Metrarc.1.1 ............................................... 17
3 Security Modelling Tools ..................................................... 18
  3.1 Introduction ..................................................................... 18
  3.2 Updated User Guide ....................................................... 19
    3.2.1 Getting started ....................................................... 20
      3.2.1.1 Main page ....................................................... 20
      3.2.1.2 Registration .................................................... 20
      3.2.1.3 User login ...................................................... 21
      3.2.1.4 Logout ........................................................... 22
    3.2.2 Model management ................................................ 22
      3.2.2.1 List models ...................................................... 22
      3.2.2.2 Create model ................................................ 23
      3.2.2.3 Export/Import ............................................... 24

Project Title: SHiELD  Contract No. GA 727301  http://project-shield.eu/
3.2.2.4 Delete model

3.2.3 Model construction

3.2.3.1 Select and add asserted asset

3.2.3.2 Add relationship between assets

3.2.3.3 Delete asset

3.2.3.4 Delete relation

3.2.3.5 Rename asset

3.2.4 Model validation

3.2.4.1 Validation issues

3.2.5 Threat management

3.2.5.1 Threats associated with a given asset

3.2.5.2 Selecting controls in the Control sets panel

3.2.6 Threat Editor

3.2.6.1 Using the Threat Editor to select controls

3.2.6.2 Accepting a threat

3.2.7 Regulatory Compliance

3.2.8 Misbehaviour Sets

4 Updated Secure Design patterns

5 Conclusions and next steps

6 References
List of Figures

FIGURE 1 - CONSOLE ACCESS TO A HOST LOCATED IN A PHYSICAL SPACE BY A STAKEHOLDER .................. 11
FIGURE 2 - ENCRYPTED STORAGE AND TRANSFER OF GENETIC DATA ........................................... 12
FIGURE 3 - ACCESS CONTROL AND LOGGING FOR GENETIC DATA .................................................. 13
FIGURE 4 - AUDITABLE LOGGING FOR PERSONAL DATA ACCESS ..................................................... 14
FIGURE 5 - AUTHORISED ACCESS TO PERSONAL DATA .................................................................... 14
FIGURE 6 - STORING PERSONAL DATA IN A PRIVACY-SHIELD COMPATIBLE LEGAL SPACE ................... 15
FIGURE 7 - CROSS-JURISDICTIONAL DATA TRANSFER .................................................................... 16
FIGURE 8 - SENSITIVE DATA LEAKING ............................................................................................... 17
FIGURE 9 - USING IC METRICS TO PREVENT DATA ACCESS ON A MODIFIED DEVICE ...................... 17
FIGURE 10 – SYSTEM SECURITY MODELLER MAIN PAGE ................................................................. 20
FIGURE 11 – USER LOGIN PAGE ........................................................................................................... 22
FIGURE 12 – LISTING THE MODELS .................................................................................................. 23
FIGURE 13 – MODEL DROP-DOWN MENU ........................................................................................... 23
FIGURE 14 – CREATING A NEW MODEL ............................................................................................... 23
FIGURE 15 – IMPORTING A MODEL FROM A FILE ................................................................................ 24
FIGURE 16 – MODEL EDITING ............................................................................................................... 25
FIGURE 17 – SELECTING ITEMS FROM NETWORK ASSETS ..................................................................... 26
FIGURE 18 – ADDING ASSETS TO THE MODEL CANVAS ...................................................................... 27
FIGURE 19 – CONNECTING ASSETS ..................................................................................................... 27
FIGURE 20 – TARGET ASSETS FOR MAKING CONNECTIONS ................................................................. 28
FIGURE 21 – CONNECTING ASSETS ..................................................................................................... 28
FIGURE 22 – DELETE RELATION .......................................................................................................... 29
FIGURE 23 - SUCCESSFUL VALIDATION MESSAGE ............................................................................ 29
FIGURE 24 – MODEL AFTER SUCCESSFUL VALIDATION ................................................................... 30
FIGURE 25 - EXAMPLE MODEL WITH MISSING RELATION .................................................................... 31
FIGURE 26 - FAILED VALIDATION MESSAGE ......................................................................................... 31
FIGURE 27 - MODEL ISSUES (NO ASSET SELECTED) ............................................................................ 32
FIGURE 28 - INCOMING RELATION ISSUES (BROWSER ASSET SELECTED) ......................................... 33
FIGURE 29 - CHANGE RELATION DIALOG .......................................................................................... 33
FIGURE 30 - MODEL SUMMARY (NO CURRENT ISSUES) ...................................................................... 34
FIGURE 31 - INCOMING RELATION ISSUES FIXED (BROWSER SELECTED) ........................................... 34
FIGURE 32 – LIST OF THREATS ASSOCIATED WITH HOST1 ................................................................ 35
FIGURE 33 – THREAT HIGHLIGHTING AND SELECTION ..................................................................... 36
FIGURE 34 – AVAILABLE CONTROL SETS FOR HOST1 ....................................................................... 37
FIGURE 35 – RESOLVING THREATS ..................................................................................................... 37
FIGURE 36 – CONTROL STRATEGIES IN THE THREAT EDITOR .............................................................. 39
FIGURE 37 – ACTIVE CONTROL STRATEGY ........................................................................................... 39
FIGURE 38 – ACCEPTING THREATS ....................................................................................................... 40
FIGURE 39 – ACCEPTING A THREAT ..................................................................................................... 40
FIGURE 40 – THE UI AFTER ACCEPTING A THREAT ........................................................................... 40
FIGURE 41 - INITIAL VALIDATED MODEL SUMMARIZING COMPLIANCE STATUS ............................. 41
FIGURE 42 - COMPLIANCE EXPLORER AND COMPLIANCE THREAT, INDICATING HOW TO BREAK PATTERN ....... 42
FIGURE 43 - DELETING THE HASACCESS TO RELATION ................................................................... 42
FIGURE 44 - REVALIDATING THE MODEL .............................................................................................. 43
FIGURE 45 - REVALIDATED MODEL SHOWING RESOLVED COMPLIANCE THREAT ............................ 43
FIGURE 46 – ASSET MISBEHAVIOURS PANEL .................................................................................... 44
FIGURE 47 – MISBEHAVIOUR EXPLORER ............................................................................................ 45
FIGURE 48 – SHIELD SOURCE CODE ANALYSER SCORECARD ............................................................ 45
FIGURE 49 – SONARQUBE OVERVIEW .................................................................................................. 46

Project Title: SHIELD
Contract No. GA 727301
http://project-shield.eu/
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Model status icons</td>
<td>23</td>
</tr>
<tr>
<td>Table 2</td>
<td>Model editing controls</td>
<td>25</td>
</tr>
<tr>
<td>Table 3</td>
<td>Results of the OpenNCP source code scanner</td>
<td>48</td>
</tr>
<tr>
<td>Table 4</td>
<td>82 CWE weaknesses found within OpenNCP</td>
<td>49</td>
</tr>
</tbody>
</table>

Terms and abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CWE</td>
<td>Common Weakness Enumeration</td>
</tr>
<tr>
<td>GDPR</td>
<td>General Data Protection Regulation</td>
</tr>
<tr>
<td>NCP</td>
<td>National Contact Point</td>
</tr>
</tbody>
</table>
Executive Summary

This is the accompanying report for WP4’s D4.2 software deliverable. It describes the work done in tasks T4.1, T4.2 and T4.3, and:

- describes the current proof-of-concept prototypes (i.e. describes D4.2 itself); and
- outlines the future development plan for the incremental updates to the SHiELD WP4 deliverables.

D4.1 described the existing tool including some generic improvements and initial versions of the extensions to support modelling of regulatory compliance. In D4.2 we have now progressed based on feedback and input obtained from project partners in WP5 and WP6. These include:

- An initial set of compliance threats as well as a user interface to explore them
- A mechanism to help prioritising threats based on the risk they pose to the system, where this depends on the threat likelihood (calculated by our tools) and the impact (as specified by the user)
- General usability improvements and optimisations

This version of the “System Security Modeller” tool enables the user to create design-time models of IT systems describing healthcare applications, using the first version of our SHiELD domain model. It provides some SHiELD-specific asset types and relationships that the user can assert and which can trigger the threats described in this report.

Furthermore, this document describes the work from T4.3, introducing the “SHiELD analyser” tool, which can be used for static code analysis on software such as OpenNCP, in order to detect vulnerabilities as defined by the Common Weakness Enumeration (CWE), a list of software security weaknesses maintained by the non-profit MITRE Corporation.

Summarising, the main achievements in the first half of the project are:

- extension of the System Security Modeller to capture regulatory compliance requirements;
- extension of the associated Knowledge Base to include models for threats to compliance and threats that originated from SHiELD WP5;
- incorporation of a risk level calculation algorithm from a sister H2020 project, and extension of the SHiELD knowledge base to support these features;
- development of a tool for finding software vulnerabilities in the OpenNCP code;
- support for an initial validation exercise in WP6.

The next steps towards D4.3 will be to extend the knowledge base by adding more threats, specifically relating to compliance, and link the results from T4.3 back to T4.1 and T4.2.
1 Introduction

This report accompanies D4.1, which is a software deliverable that will be deployed publicly by the end of M18. It provides mainly documentation for the outputs of tasks T4.1 and T4.2.

Section 2 describes changes to the knowledge base since D4.1. This includes the introduction of a risk level ranking mechanism to prioritise threats as well as an updated collection of generic network security threats, threats addressed by WP5 tools and some initial compliance threats as identified by the use-case partners from WP6.

Section 3 constitutes the updated user guide for the System Security Modeller tool from T4.1, covering features to support the updated knowledge base from T4.2.

Section 4 describes a separate dashboard based on static code analysis results, which can be used to discover vulnerabilities in the OpenNCP codebase.

Finally section 5 concludes this deliverable by summarising the changes and providing an outlook to D4.3 which will be due at the end of M24.

2 Updated knowledge base

2.1 Risk level evaluation

The main improvement to the knowledge base relates to updates to the schema, also known as “core model”. When validating a system model using the knowledge base as of D4.1, the user is given a list of all threats. Depending on the input system and the knowledge base against which this system is validated, the list of threats can be quite long, containing several hundred system-specific threat occurrences. The result if that the user is left without guidance on the priority in which these threats should be addressed.

To handle this, SHIELD is taking advantage of a procedure developed in the H2020 RestAssured project to evaluate the level of risks from each threat, which is now also integrated in the SHIELD privacy by design risk modelling tools. We have described the mathematical basis and calculation methods for these new concepts in RestAssured D7.1 [1]. To use this in SHIELD, it was necessary to enhance the knowledge base from D4.1 by adding features needed to use the RestAssured algorithms. The current update therefore includes the following new concepts:

- **Likelihood** of something (negative) happening,
- **Impact levels**, which describe the consequences of a threat on an asset,
- **Trustworthiness levels**, which specify how trustworthy different aspects of an asset are or how effective a control strategy is in protecting against a threat and
- **Risk levels** as a measure for the urgency in which threats have to be addressed.

For SHIELD and the rest of this section we can summarise the new concepts as follows:

- Compared to D4.1, threats no longer have a binary status (active/not active). Instead, they are described in terms of likelihood and impact level, which are used to calculate the risk level. Control strategies now reduce the likelihood that a threat can happen whereas impact levels on assets increase the threat risks. If for example (anonymised) log data is leaked, the impact is not as big as if it was sensitive patient data and threats leaking sensitive data should be addressed much more urgently.
- Threats now have what we call an “entry point”, i.e. a means of triggering the threat. This entry point refers to a weakness in an asset, expressed as a lowered
trustworthiness level. A local network for example has a “High” user trustworthiness whereas the internet has “Very Low” user trustworthiness.

- The new levels are used in the calculation of secondary effect chains and root cause analysis. If for instance a threat has two possible causes, one of which is “Very likely” and one of them has “Low” likelihood, the cause would be the likely threat which should be addressed first.

The threats we present in this section make use of these new concepts. We have not included an exhaustive list as this report serves for illustration purposes only. The full set of threats is deployed on the service.

The System Security Modeller software supports the ISO 27005 risk analysis methodology, and that does not define scales for likelihood, impact and risk. The scales are therefore defined in the knowledge base, and for the SHIELD knowledge base the following scales are used:

- threat Likelihood = Very High: the threat will occur, even if exposure to the risk is brief;
- threat Likelihood = High: the threat will occur if exposure to the risk is prolonged, and it is quite likely to occur even if exposure is brief;
- threat Likelihood = Medium: the threat is unlikely to occur if exposure to the risk is brief, but it is very likely to occur if exposure is prolonged;
- threat Likelihood = Low: the threat is unlikely to occur even if exposure to the risk is prolonged;
- threat Likelihood = Very Low: there is essentially no chance the threat will occur.

Asset trustworthiness actually uses the same scale, but with the labels inverted. Thus if an asset involve in a threat has High trustworthiness (with respect to the relevant attributes), it means the likelihood of the threat occurring will be at most Low, and vice versa.

The impact scale for asset misbehaviour used in the current SHIELD knowledge base was defined in a similar way:

- impact = Very High: means if the asset misbehaves in this way, it will be fatal to the stakeholders that depend on it;
- impact = High: means if the asset misbehaves in this way it will cause a serious impact to the stakeholders (e.g. adding significant costs), or it may become fatal if prolonged;
- impact = Medium: means if the asset misbehaves in this way the impact can be tolerated, but it will become serious if prolonged;
- impact = Low: means if the asset misbehaves in this way the impact can be tolerated indefinitely but it does still cause some harm to dependent stakeholders;
- impact = Very Low: means if the asset misbehaves in this way the impact will be small.

Combining these gives the overall risk level. Again, ISO 27005 doesn’t specify exactly how this is done, other than to say the risk level from a threat must depend on both the threat likelihood and its impact. At present, the SHIELD knowledge base also uses a five level scale for risk levels:

- risk = Very High: unacceptable risk, which will certainly cause serious harm;
- risk = High: unacceptable risk, likely to cause serious harm;
- risk = Medium: unacceptable risk, likely to cause significant harm but could be tolerated for a sufficiently short period;
- risk = Low: acceptable risk, but does cause minor problems so better avoided;
- risk = Very Low: acceptable risk causing only negligible problems.
Medium level risks may be acceptable while one is addressing a short term run-time requirement, e.g. between a data subject withdrawing consent and the relevant data being deleted. At present, our privacy by design tools don’t support modelling of these ‘dynamic’ risks so in practice, the design time analysis should aim to produce residual risk levels (after security measures have been introduced) that are Low or Very Low.

Note that because these scales of likelihood, trustworthiness, impact and risk are encoded as part of the SHIELD knowledge base, they can be modified in future to meet project needs.

2.2 Generic security threats

As in D4.1, our knowledge base covers basic security threats that would apply in any situation involving computer networks. They are not specific to SHiELD but still very relevant as they might compromise the system. This section shows a sample threat from that knowledge base. The full set of threats has been deployed as part of the tool, see section 3.

Threat names consist of multiple elements separated by dots:

- the affected asset (e.g. “H” for “Host”);
- the caused misbehaviour (e.g. “C” for “Confidentiality”) or “x” if the threat is a compliance threat (see below);
- the pattern it applies to (e.g. “CS” for the “Client-Service” pattern);
- a number if more than one threat exists for a combination of the above factors.

A threat diagram shows the following elements:

- the nodes in the threat pattern that are required for the pattern to be matched and their relationships to each other;
- the threat itself, pointing to the threatened asset;
- controls to mitigate the threat if a control strategy exists;
- the entry point for the threat (optional);
- caused misbehaviours (optional).

For example, Figure 1 shows a threat in which the attacker has physical access to a space containing a Host, and can use console access to gain administrator privileges that should only be accessible to the Stakeholder responsible for managing this Host. The entry point is the user trustworthiness of the space, i.e. how far users with access to the space may be trusted.

![Threat Diagram](image-url)
point. In this case the strategy is to control access to the space and using physical identification, e.g. a fingerprint or memorized passcode, for stakeholders who are allowed into the space.

2.3 Compliance threats

When analysing a system model, a security analyst wants to find all threats to the system and find ways to mitigate them. In SHiELD, however, another concern for systems is whether they are compatible with regulation in different jurisdictions. Like threats, such requirements can be queried in a system model in an automated fashion. To do this, the generic compliance “threats” need to be added to the knowledge base, modelling situations that are not allowed under a particular set of rules. In SHiELD, we need to check against multiple sets of these rules, each one representing a jurisdiction. There might also be more rules relating to organisational, local, regional or supra-regional compliance.

The data model we use for compliance threats is simple. A domain model defines zero or more compliance sets. These compliance sets contain a collection of compliance threats. A compliance threat is matched like a generic security threat using assets and relationships but it doesn’t have external causes, doesn’t cause any misbehaviours, and it cannot be accepted. A compliance threat in a system model can either be addressed by applying a control strategy if one exists or by changing the system model’s structure so that it doesn’t cause the threat to appear. This can be achieved by removing assets and/or relationships.

This section describes the initial sets of compliance threats included in the knowledge base. These are just a starting point showing how compliance can be checked using the System Security Modeller. More threats will be added during the remainder of the project to provide a more complete coverage, including privacy by design requirements from the legal analysis in WP3.

2.3.1 Italy

Italian regulations will – like all other European countries’ legislation – be adjusted to implement GDPR. However, in the current political situation, this adjustment has not yet taken place so parts of former Italian legislation [2] remain until they are phased out. As a first step to encoding Italian compliance requirements, we focused on rules for accessing genetic data.

2.3.1.1 GD.x.HGenDHP.1

This regulation stipulates, that any genetic data that gets transferred over a network needs to be encrypted.

![Diagram: Encrypted storage and transfer of genetic data]

Figure 2 - Encrypted storage and transfer of genetic data
The pattern of assets (the blue boxes) in Figure 2) defines the circumstances under which this threat is relevant, specifically when a process accesses data (held in a database) from a different Host than the database (and data) is stored in. That implies access must be via an exchange of messages over a network. In the situation the system does not comply unless the process that accesses the data in this way uses encrypted communications.

### 2.3.1.2 GD.x.StHGenD.1

This compliance threat describes the access to genetic data (where it applies more generally to both physical data, such as samples, and digital data although in SHiELD we focus on the digital data). The regulation states that access to the space where the data is stored must be restricted to authorised persons who must be identified using a biometric key. Furthermore, the host must log access so that it is obvious who accessed the data when.

![Diagram](image.png)

**Figure 3 - Access control and logging for genetic data**

### 2.3.2 Spain

Spanish regulations [3], which are identical with Basque regulations for the purposes of SHiELD.

#### 2.3.2.1 PD.x.StHPersD.1

These regulations require that whenever a stakeholder has access to personal data, the access needs to be logged. Furthermore, it stipulates that the audit trail is made available to the citizen, whose data is accessed.

A simplified version of this requirement has been modelled in Figure 4. Unless the data access is logged on the host, the pattern displayed is not compliant with Spanish regulations.
2.3.2.2 PD.x.StHPerD.2

Access to personal data triggers another regulation. Any stakeholder accessing personal data needs to identify using a key card and PIN to prevent unauthorised access. The threat shown in Figure 5 is closely tied to the one from Figure 4 and the authorization data will be contained in the log – yet both requirements have to be satisfied in order for the system to be compliant.

2.3.3 GDPR

The GDPR has come into effect throughout the EU on the 25th May 2018. The first example GDPR compliance requirement included in our knowledge base relates to the jurisdictions where personal data may be stored. Figure 6 shows that all personal data needs to be hosted in a legal space that is part of the Privacy Shield framework [4], i.e. an organisation within a set of selected countries that has signed up to the framework.
2.4 Threats addressed by WP5 tools

This section describes security threats that can be addressed by employing WP5 tools to block or mitigate the risk. These threats were developed in discussion with the partners developing these tools in WP5, so they also explain what threats those tools are intended to prevent. By including them in the knowledge base, the SHIELD privacy by design tools can help system designers detect when they should consider using those tools. In this way, the privacy by design tools from WP4 add value to the security tools from WP5, and vice versa.

2.4.1 UC.IBM.1 – Data hiding tool

"Data masking is the process by which sensitive data is replaced, possibly in a reversible manner, with data that is unintelligible to the receiver. The masked data is usually sensitive data, such as personally identifiable information (PII, i.e., any information that relates to an identified or identifiable living individual\(^1\)), health information, names, addresses, and so on. The main purpose of data masking is to preserve the data owner privacy enforce the data owners consent and comply with legal regulation (such as GDPR).

Encrypting all the data, or replacing the data with tokens or fake data would be a trivial solution for designing a masking data tool. However, when the requirement is that the data must remain meaningful and real, designing data masking tool become more challenging. For example, in case of Shield, a transmitted CT image must remain meaningful, however, it must not reveal sensitive data" [5].

2.4.1.1 Use case description

In SHIELD’s use cases, patient from a specific EU country (home country) travels to another EU country (host country), then she needs a health care consulting or treatment, which requires

---

querying the patient’s data from the home country. In such circumstances, the EU regulations (GDPR) specifies which data should be transmitted, which data must not be transmitted and which kind of transformation (e.g., pseudonymisation or anonymization based on Article 4 and Recital 26) data should undergo, in addition the home and host countries regulations and the user consent, which could further limit data transferring based on patient’s preferences.

Figure 7 – Cross-jurisdictional data transfer

Figure 7, shows personal health data relating to a patient from country ‘B’, which resides on a health provider’s servers in country ‘B’. It is subsequently moved to another health provider in country ‘A’, caused by one of SHiELD’s use cases (e.g., “Breaking glass circumstance”, in which an emergency situation in country ‘A’ has to be tackled). The data passes through the countries’ national contact points.

2.4.1.2 Identified threats

2.4.1.2.1 Lawfulness (PD.x.IBM.1.1)

According to Article 5 of GDPR (“Principles relating to processing of personal data”), “personal data shall be processed lawfully, fairly and in a transparent manner in relation to the data subject”. Moving health data between different countries may violate the regulations of the involved countries as well as the EU’s. This threat is a compliance threat that doesn’t necessarily originate from malicious behaviour but rather from a misconfiguration or erroneous design.

As with most compliance threats, no specific control strategy exists to address the entire threat. A specific control may need to be applied to address any specific compliance issue. For example,
modifying masking policies or redesigning the system data flow may be needed to prevent the compliance threat from occurring.

![Figure 8 - Sensitive Data leaking](image)

**Control strategy:**

The masking tool and the consent management tool can provide some mitigation to this threat by enforcing consent and addressing several privacy issues.

### 2.4.2 UC.Metrarc.1

Metrarc’s technology as developed in WP5 addresses the issue of impersonation by means of theft of (mobile) client devices. This could give the thief access to passwords saved by client apps stored on the phone, allowing them to use these apps to access services and data.

#### 2.4.2.1 D.C.Metrarc.1.1

This threat addresses the case, in which a stolen mobile device might be used by an attacker to access data on a service, using stored credentials on the device. This can be the case because it is for instance hard to spoof the credentials.

![Figure 9 - Using IMetrics to prevent data access on a modified device](image)
Figure 9 shows that this can be prevented by using ICMetrics, where usage data such as the angle in which the device is held or the strength of the interaction with the touchpad is used to generate a key. Along with the regular credentials, this is then used to authenticate the user to the service. In effect, ICMetrics provides a type of two-factor authentication, in which the second factor is transparent to the user.

3 Security Modelling Tools

3.1 Introduction

Since the last release (D4.1), the “System Security Modeller” tool and the underlying knowledge base have been extended to address more SHiELD-specific targets, such as determining the compliance status of a design-time system. This section describes the updated user guide with a focus on new and improved functionality.

Firstly it should be noted that we rebranded the tool from “System Modeller” as is appears in D4.1 to “System Security Modeller” to better reflect the nature of the tasks it supports.

Overall, the System Security Modeller tool allows a system designer to create a high level model of their system, detect potential security threats, analyse the level of risk from each threat, and find out what security measures could be used to reduce the level of risk in the system by addressing the highest risk threats.

The tool is based on work from previous EU and UK projects which has been extended in SHIELD to handle cross-border regulatory compliance by using compliance sets to model compliance requirements in each jurisdiction. The tool uses a knowledge base which has also been extended in SHIELD (as described in Section 2 above) to include some sample compliance threats for each jurisdiction, and security threats that SHIELD security measures (from WP5) are intended to address. The SHIELD knowledge base also includes a range of basic asset types plus mundane threat types and countermeasures (mostly drawn from previous work) that are relevant to SHIELD scenarios.

The System Security Modeller is also being developed in other projects including one other H2020 project (RestAssured) which focuses on evaluating risks in cloud based applications. In RestAssured, it was necessary to add algorithms to calculate risk levels using the approach specified in ISO 27005. This risk level assessment functionality is useful in SHIELD, so this feature has also been included in the current SHIELD release. The scales used to represent risk factors form part of the SHIELD knowledge base and can be customised to meet SHIELD needs. The current scales are described in Section 2.1 above, and with these scales a risk is considered acceptable if the associated risk level is Low or Very Low.

Compliance threats are handled slightly differently, because a system that does not comply with relevant regulations is simply not allowed. Compliance threats therefore don’t have risk levels, but capture design patterns that should not be used within the system, or can only be used if certain mandatory security measures are in place. Later in the project, we intend to improve the coverage of compliance requirements by the knowledge base to cover the trial partner jurisdictions and the GDPR. This will allow the knowledge base to be used to find cross-border compatibility issues and highlight them as areas where further convergence may be needed. In the meantime the SHIELD enhancements to the software allow users to check for compliance threats under each jurisdiction supported by the knowledge base.
Here, we chose not to provide a detailed description of the tool implementation, part of which would duplicate material on risk evaluation algorithms from RestAssured Deliverable D7.1. Instead, this section provides a user guide to assist SHIELD partners in using System Security Modeller to assess privacy risks during the design of a system (or connection between systems) involving data exchange between organisations or across borders. The overall procedure consists of the following steps:

1. Create a high level model of the system structure by dragging assets of types defined in the SHIELD knowledge base into the System Security Modeller canvas, and creating the appropriate relationships between them.
2. Validate the model, which fills in some details that can be inferred from the high level structure, checks that the model doesn’t break any rules encoded in the knowledge base, and (if it is OK), generates a threat catalogue for the system.
3. Specify assumptions about the trustworthiness of each asset, and also the impact of different types of misbehaviour at each asset.
4. Specify security controls (from the knowledge base) that should be implemented in the system, including (but not limited to) those provided by tools from SHIELD WP5, and any that are mandated to comply with regulations.
5. If after this there are compliance threats that cannot be addressed by using security controls, it means the system design itself is flawed. In that case return to step (1) and modify the system design by adding or removing assets or inter-asset relationships.
6. Evaluate the likelihood of each threat and hence misbehaviours caused by threats, and combine this information with the impact of each misbehaviour to determine the level of risk from each system threat.
7. If the risk level is too high, return to step (4) and add more security controls, focusing on reducing the likelihood of the threats that have the highest risk levels.
8. If the risk level is acceptable, use the security measures defined for the system as part of the design specification for the system.

The user guide focuses mostly on the features of the software needed for steps (4) and (5), to find and address high risk threats, and to find compliance issues.

This procedure represents a first draft of the SHIELD approach to privacy by design. The key innovation is the use of automated machine reasoning to speed up the identification and analysis of security risks and detection of privacy and data protection regulatory compliance requirements. This makes it possible to use the approach with an agile software development which is today the norm for the development of apps used to collect or access data, or when one needs to establish a (possibly cross-border) connection between health data systems.

### 3.2 Updated User Guide

The following sections provide details of the System Security Modeller functionality. We have included material produced in previous projects in this section so it provides a complete description of the user interface for project partners wishing to use the proof of concept release. This covers:

- Getting started
- Model management
- Model editing
  - Stage 1: defining assets and relationships which provide the initial model of a network
  - Stage 2: validation and auto-generation of threats
Stage 3: defining threat management strategy (selecting controls for assets or control strategies for threats)

- Model outputs (e.g. export)

The modelling process has three stages that may be repeated several times. First, the user constructs a model by placing assets onto the modelling canvas and creates links (or relations) between them (these user-defined entities assets are called “asserted assets” and “asserted relations”). The validation process in Stage 2 automatically generates inferred assets/relations, threats and security controls to counteract these threats. The validation process also determines whether the information provided about the assets and relationships is consistent and complete. If the validation fails (i.e. the model gets marked as ‘invalid’) then the user should go back to Stage 1 and update the model, so that it contains sufficient/correct information for a successful validation. In Stage 3, the user addresses threats by selecting or modifying the set of security controls that protect the assets in the system. The aim is to eliminate or at least mitigate the threats.

3.2.1 Getting started

The software can be accessed at https://shield.it-innovation.soton.ac.uk/system-modeller/. Upon first use, the user will be directed to the welcome page, as described in the next section.

3.2.1.1 Main page

On the main (welcome) page of System Security Modeller (Figure 10) there are several links (also available on other pages): System Modeller, Home and a drop-down menu under Account. The System Modeller and Home links return the user to the main page of System Security Modeller. The options under the Account dropdown are Sign In, Register and Forgot Password.

![System Security Modeller main page](image)

Figure 10 – System Security Modeller main page

3.2.1.2 Registration

If the user is new to the system, they must first register a new user account on System Security Modeller. If they have previously received an invitation email from the System Security Modeller Admin, they simply click the link in that email, which opens the registration page. Otherwise, the user manually clicks the Register button on the welcome page, or the Register menu option.
The registration form indicates the mandatory fields (red asterisk). The user enters their preferred user name (which should be unique on the system), their normal email address, first name (optional), last name (optional), and password (entered twice). Any errors are indicated to the user on clicking the Register button, otherwise the details are stored. Once the form has validated, the user will receive a message “Registration pending confirmation. Please await an email confirming activation”. The must then wait for the Administrator to approve the new account, at which point they will receive a confirmation email.

3.2.1.3 **User login**

The login page of System Security Modeller is activated either by clicking on *Sign In* link in the dropdown menu or by clicking on the *Login* button (see Figure 11). The user must enter their username and password. These are case-sensitive.
3.2.1.4 Logout

Logout is activated by clicking on the Sign Out link in the dropdown menu on the main page under the currently logged in user.

3.2.2 Model management

The term “Model Management” incorporates several functions such as:

a) Listing models  
b) Creating models  
c) Importing/exporting models  
d) Deleting models

3.2.2.1 List models

After a successful login, the user is presented with a list of their models (see Figure 12).
In the model details, the **Domain** used by the model is labelled. There is also a description of the model, which can be edited via the *Edit Details* function (described in Figure 13). At the bottom of the model panel, there are several icons that reflect the current status of the model (see Table 1).

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄</td>
<td>last modification of the model</td>
</tr>
<tr>
<td>🕒</td>
<td>when the model was created</td>
</tr>
</tbody>
</table>

**Table 1 – Model status icons**

The drop-down menu in the top right corner of the model window offers several functions, these are: *Delete*, *Export* and *Edit Details* (see Figure 13).

**3.2.2.2 Create model**

By clicking on the “Create New Model” button, the user can create a new model (see Figure 14). The “Domain Model” drop-down allows the user to choose which domain model to use (N.B. for the SHiELD project, please select the specific SHiELD domain model). The new model is added to the models list, as described in the previous section.

**Figure 12 – Listing the models**

**Figure 13 – Model drop-down menu**

**Figure 14 – Creating a new model**

The “Import Existing Model” option (available on the “Create New Model” drop-down list) allows the user to import a model from a file (see next section).
3.2.2.3 Export/Import

Once we have constructed the model, it may be exported into a file. Using the Export option in the model drop-down menu (see Figure 13), the model is exported into the user’s “Downloads” directory.

Similarly, the Import operation allows the user to upload a previously saved file into a new model. In order to do this, the user clicks on the “Create New Model” control and selects the “Import Existing Model” option. A dialog appears (see Figure 15). The user may choose to Import asserted facts only (e.g. asserted assets, relations). If restoring a previous version of a model, the user can check Overwrite existing model. Attempting to import the same model without this being checked will result in an error. A user can re-import an existing model with a different name by checking New Name.

![Figure 15 – Importing a model from a file](image)

3.2.2.4 Delete model

The delete action removes the model along with any associated data items (e.g. assets, relationships).

3.2.3 Model construction

Clicking the Edit button (on a model in the models list) opens the model editing page that consists of three main parts (see Figure 16).

On the left side is the “Asset Palette”, in the middle the Model Construction Canvas and, on the right, the Model Summary. These are described below.

Asset Palette: contains icons representing the asset types (e.g. “Host”) that may be added to the model.

Model Construction Canvas: the main area for designing/displaying the model.

Model Summary: summarizes details about the model and shows a lists of assets, threats, compliance sets, etc. (these are empty initially).
The top right corner contains five buttons, these are shown in Table 2 below:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄</td>
<td>Process (validate) the model</td>
</tr>
<tr>
<td>🔄</td>
<td>Calculate risk levels</td>
</tr>
<tr>
<td>🔄</td>
<td>Configure the model</td>
</tr>
<tr>
<td>🔄</td>
<td>Zoom controls</td>
</tr>
</tbody>
</table>

Table 2 – Model editing controls

3.2.3.1 Select and add asserted asset
The Asset Palette contains various assets; these fall into four main categories:

- a) HostedAsset
- b) NetworkAsset (for illustration see Figure 17)
- c) Space
- d) Stakeholder
Figure 17 – Selecting items from Network Assets

An asset can be added to the model by selecting an icon in the Asset Palette and dragging it onto the model canvas (see Figure 18, showing five added assets). N.B. these have already been renamed, as described in section 3.2.3.5).

By clicking on the asset, the right-hand panel updates to show the following asset details:

a) Name, type and description of the asset
b) Cardinality (i.e. how many instances there could be during run-time)
c) Data Properties (additional information and annotations, such as port numbers or mappings to other systems)
d) Trustworthiness (i.e. how much certain aspects of this asset can be trusted)
e) Incoming relations
f) Outgoing relations
g) Inferred relations (once the model has been validated)
h) Control sets (empty until the model has been validated)
i) Threats (empty until the model has been validated)
j) Misbehaviours (empty until the model has been validated)

For illustration purposes, we analyse a typical use case representing the threats associated with accessing a web page hosted on a remote server. Constructing a security model involves placing assets onto the canvas and establishing connections between them. The model itself consists of two hosts connected to the Internet. The user of Host1 uses a Browser to access a WebService deployed on Host2. This is a simple model that represents the scenario of downloading a web page from the remote web server. The reasons for selecting this simple model are as follows:

a) Frequently occurring case, typical for all web applications
b) Simplicity
c) All types of inference can be well demonstrated, these are:
   o Inferred assets
   o Inferred relationships
d) The threats inherent to the model are well understood
e) The effect of controls and threats can be easily interpreted
3.2.3.2 Add relationship between assets

Once the assets have been put onto the modelling canvas, the user can connect pairs of assets by establishing links between them, by clicking on the green cross that appears in the top left corner (Figure 19).

After clicking on the green cross (on Host1 in our example) a blue tick sign will appear on several other assets, indicating that a link can be made to these assets (see Figure 20).
By clicking on one of the blue tick icons, we can establish a connection between the two assets. Here, we continue making connections (or “asserted relations”) between assets until the model appears as in Figure 21.

3.2.3.3 **Delete asset**

Assets can be deleted by clicking on the red trash icon of the asset in the top right corner (see Figure 19). The delete operation also removes all links between the selected asset and other assets.
3.2.3.4 Delete relation

By right clicking on the connection between two assets, a dialog with a delete button pops up (see Figure 22). The delete operation applies to the relationship itself along with any associated inferred assets (i.e. those generated after model validation), however the originally connected user-asserted assets remain on the canvas.

![Figure 22 – Delete relation](image)

3.2.3.5 Rename asset

The user can rename an existing asset by editing the asset’s name under the corresponding icon. N.B. by changing the name, the asset’s connections will stay unaffected. All asset names must be unique.

3.2.4 Model validation

Once the model has been constructed, it then needs to be validated. This operation is initiated by clicking on the red “play” button (see Table 2), which indicates that the model is currently invalid (for a valid model, this button would be green). The validation operation runs semantic reasoning that generates inferred assets and relations that are added to the model automatically, and produces a list of threats and misbehaviour sets that can be associated with the given model. This operation guarantees that the inferred assets are consistent with the asserted assets and relationships.

The overall validation operation can take some time, depending on the complexity of the model. On successful completion, a message pops up as follows:

![Figure 23 - Successful validation message](image)

The user clicks “Continue” and the updated model is presented.

Once one has a valid model including potential Threats and asset Misbehaviours, it is possible to calculate risk levels for all threats and misbehaviours. These depend on the Trustworthiness Attributes for assets (which determine the likelihood of threats), and the Impact level settings for Misbehaviours. The validated model will have default values for these which are determined from the types of Asset and the type of Trustworthiness Attribute or Misbehaviour at each Asset. In most cases the defaults will be acceptable, but in most system models a few of the default settings will need to be overridden by user input, indicating which assets are likely entry points.
for attackers, and which asset misbehaviours would cause serious problems for operation of the system.

Once the relevant inputs have been provided, the Risk Calculation is launched via the “Calculate risks” button (see Table 2). This calculation can be re-launched if any of the trustworthiness levels or misbehaviour impact levels are changed, or if security measures are selected to reduce the likelihood of threats. The full validation process only has to be run again if changes are made to the composition of the system (i.e. adding or removing assets or their relationships).

Assuming that no asset is currently selected, the Model Summary panel will also be displayed (see Figure 24).

![Image of Model Summary and Threat List](image)

**Figure 24 – Model after successful validation**

Here, you will see that certain assets are now highlighted in light yellow (with a small leaf icon), which indicates that there are now associated (invisible) inferred assets and relations connected to this asserted asset (these may be viewed via the asset details panel on the right). Similarly, a relation marked in yellow indicates that there are newly added (inferred) assets and relations in the model, connected to this relation. By selecting this relation, the user can view information related to the inferred asset, via the asset details panel.

With no asset currently selected, the Model Summary is displayed. The total numbers of assets and relations is updated (now including inferred assets/relations). Now, we also see a number of threats in the system. This total is also displayed at the top right of the screen. All model threats are displayed in the Threat List panel (once expanded). Finally, the Compliance panel summarizes any Compliance Sets that have been defined, according to the domain model. Each Compliance Set indicates any associated Compliance Threats that must be addressed, in order for compliance to be achieved for that jurisdiction. In this example, the model is fully compliant, indicated by the green ticks for all Compliance Sets. We will discuss compliance further in section 3.2.7, along with examples of non-compliance.
3.2.4.1 Validation issues

In the previous section, we described the successful validation of a model. However, on occasion, the validator will return with a failure, indicating that the system model is not valid (i.e. one or more assets or relations are missing). In this section we describe an example of an invalid model and how to fix it.

One simple example of an invalid pattern of assets/relations is a Process asset which is not hosted by any Host (common sense would also suggest that a Process could never exist without a Host to run on). We can simulate this situation by taking our previous example “Test Model” and deleting the “hosts” relation between Host1 and the Browser process (see section 3.2.3.4 “Delete relation”). We therefore start with a system model as follows (Figure 25):

![Figure 25 - Example model with missing relation](image)

The user clicks the validation button (as described in the previous section) and waits for the operation to complete. In this case, the validation finishes more quickly than usual, as a problem has been found. A message pops up, reporting that the “Model has issues to fix” (see Figure 26).

![Figure 26 - Failed validation message](image)

The user clicks “Continue” and (assuming no assets are currently selected) the screen appears as in Figure 27. You will notice that the Browser asset is now highlighted in pink, indicating that it has one or more issues. In the Model Summary, you will also see that there is now a red “Model Issues” panel, showing “Host? Hosts Browser”. This indicates that the validator expected there to be an incoming relation to the Browser asset from a Host asset (i.e. any asset of type “Host”).
The same information may be viewed in the System Security Modeller as follows. By selecting the Browser asset (i.e. the asset highlighted pink), the asset details panel is displayed (see Figure 28). Expanding the red “Incoming Relations” panel, it now reports “incomplete relations defined” and lists the missing “Host? Hosts Browser” relation, as described above.

In either of the above-described views, the user may fix the issue by clicking on the “Host?” button. This pops up the “Change Relation” dialog, as shown in Figure 29. The user is presented with a drop-down list of assets of type “Host” that are already present in the system model².

---

² N.B. if there were currently no assets in the model of type “Host” then System Modeller would simply report that no suitable assets are currently available. In this case, the use must first create a new Host asset on the canvas.
The user selects a Host from the list (in this case, Host1), then clicks “Save changes”. The new relation is created and the System Security Modeller re-checks the model for issues. In this case, there are no remaining issues in the model, so the screen appears as follows (Figure 30, assuming no assets are currently selected):
The Browser asset is now displayed normally, and the “Model Issues” panel has disappeared from the Model Summary. If we now check the Browser asset in more detail (by selecting this asset), the asset details panel appears as in Figure 31. Here, the “Incoming relations” panel has reverted back to blue and the “Host1 hosts Browser” relation is listed, as expected.

Once all model issues have been fixed (e.g. by adding missing relations, etc.), the user must then re-validate the model (as we still have not determined any system threats, etc.). The user follows the process described in section 3.2.4, after which they should have a successfully validated model, containing all the newly inferred assets, relations, threats, etc.
3.2.5 Threat management

3.2.5.1 Threats associated with a given asset

In order to view the threats associated with a given asset, the user must first select this asset on the canvas then click to expand the “Threats” panel within the asset details panel (see Figure 32).

Figure 32 – List of threats associated with Host1

In our example, the user has selected Host1. Here, none of the threats have yet been addressed, so each status icon is red. Alongside each threat name, the System Security Modeller also displays the associated Likelihood and Risk levels, as calculated during the Risk Calculation phase. The threats list is ordered, by default, from highest to lowest risk, as generally those threats with higher risk are of more concern to the user. The user may instead choose to order by Likelihood, or simply by the threat name.

As the number of threats in the list may be quite large, several filtering options are available here. For example, the user can display only those containing a particular string. They can also choose to display primary threats (i.e. root causes of misbehaviour within the system) or secondary threats (which model the propagation of effects within the system), or those which are still untreated, in any combination.

The user may hover over a threat in the list, which highlights the threat in green, along with its associated pattern of assets and relations on the canvas. For example, Figure 33 shows the
highlighted threat to Host1, “H.A.RoH.1_RoH_LogicalPath_(0)_Internet_Host1”. The code stands for a threat applying to a host (“H”), causing a loss of availability (“A”) in a remote attack on a host (“RoH”) involving the Host1 and Internet assets and the network path (“LogicalPath”) between them. Clicking the threat name pops up the Threat Editor that allows the user to view threat details, and mitigate the threat in various ways (this will be described further in Section 3.2.6). Note that the path from the Internet (where the attacker entered the system) and the target Host is an inferred asset, created during the validation procedure, so it is not shown in the modelling canvas. Only the user defined assets involved in the threat are highlighted (the Host and the Internet).

Figure 33 – Threat highlighting and selection

3.2.5.2 Selecting controls in the Control sets panel

Threats may be resolved by selecting one or more controls within the Control sets panel (N.B. the exact control(s) required depend on each threat). Each control set represents a control on this asset. For example, the controls that are available for Host1 are shown in Figure 34; to expand the Control sets panel, simply click on it.
For the purpose of demonstration, we select the Software Patching control set and see which threats will be “resolved”. These threats are then indicated in green (see Figure 35).

Figure 34 – Available control sets for Host1

Figure 35 – Resolving threats
Figure 35 shows that five threats have been resolved as a result of selecting Software Patching from the Control sets. Resolving the threats is an iterative process; the user needs to go through the assets one by one, selecting options from the Control sets and checking which threats have been eliminated (resolved). Threats may also be resolved via the Threat Editor, as we shall see in the next section.

This User Guide describes the threat resolution steps for one asset (Host1) but the same steps are applicable to all assets. By following these steps, the user should be able to resolve most (if not all) threats associated with the given model.

3.2.6 Threat Editor

The Threat Editor is opened by clicking on a threat in a threat list (see previous section). This displays details about the threat, including a description along with its Likelihood and Risk (as also displayed in the threats list itself). Other expandable panels are summarised below:

- **Applies To Pattern**: shows the pattern of nodes (assets) and relations in the model that have been identified as causing this specific threat
- **Cause**: the direct cause (misbehaviour set) of this threat (unless it is a primary threat)
- **Effects**: misbehaviour sets directly caused by this threat
- **Secondary Effects**: other misbehaviour sets caused by this threat, further down the threat/misbehaviour chain
- **Control Strategies**: available strategies for resolving threats (see next section)

Figure 36 shows an example view of the “H.A.RoH.1_RoH_LogicalPath_(0)_Internet_Host1” threat, as described previously.

3.2.6.1 Using the Threat Editor to select controls

In section 3.2.5.2, we described how to address threats by applying controls to assets directly. This is particularly useful for experienced users who know about the effects that the controls have on the assets. Inexperienced users, however, generally need some guidance on how a threat may be addressed. This is done using Control Strategies, which are essentially collections of Control Sets. An active Control Strategy manages a threat. Whilst a Control Set describes a Control located at a particular asset, a Control Strategy contains one or more Control Sets. Semantically, this means that for the threat to be managed (blocked or mitigated), all Control Sets within the Control Strategy must be applied. The validated system model contains mappings between Threats and Control Strategies that are made visible in the Threat Editor, see Figure 36.
Figure 36 – Control Strategies in the Threat Editor

If any of the Control Strategies for a threat are active (checked), this means that the threat is managed. Managed threats appear in green. Hovering over the threat icon inside the threats panel shows the management type (e.g. “blocked”), see Figure 37.

Figure 37 – Active Control Strategy
3.2.6.2 Accepting a threat

Threats are normally accepted when the existing security measures have reduced the residual risk level to a sufficiently low level. There is no need to explicitly enter the fact that these threats can be accepted – one simply accepts the whole system model including those low risk threats.

In some cases where no control strategy exists for a given threat (e.g. see Figure 38), or the control strategy would be difficult or expensive to implement the user can also explicitly accept the threat.

Figure 38 – Accepting threats

This has no effect on the risk level – it means the threat is acceptable despite the risk level. In such cases one must also type in a reason for accepting the threat (see Figure 39).

Figure 39 – Accepting a threat

Typically this is done when there is an appropriate response that isn’t supported in the system security modeller’s knowledge base, e.g. because it involves a run-time management action rather than a security measure that can be included in the system design (as in the example shown above).

Upon saving, the icon for the accepted threat will change to indicate the new status (see Figure 40). “Accepted” threats are differentiated from other resolved threats, as they are highlighted in yellow.

Figure 40 – The UI after accepting a threat
3.2.7 Regulatory Compliance

As introduced in section 3.2.4, and shown in Figure 24, the Compliance panel summarizes any Compliance Sets that have been defined, according to the domain model. Each Compliance Set indicates any associated Compliance Threats that must be addressed, in order for regulatory compliance to be achieved for that jurisdiction.

The following section provides an example of fixing a model that is non-compliant.

First, we create a simple model, consisting of a DB Server storing Genetic Data, which is located in a Country Legal Space. Here, a Doctor stakeholder has access to the Genetic Data. The model is validated successfully, however the Compliance panel indicates that three Compliance Sets have issues (see Figure 41).

![Initial validated model summarizing compliance status](Figure 41)

In order to examine the situation in more detail, we click the “Details” button in the Compliance panel (and we can hide the Model Summary panel for the time being). The Compliance Explorer window pops up, which shows an accordion of all Compliance Sets, each of which may be expanded. Here, we expand “Compliance with Italian Regulations” (see Figure 42). This shows a single Compliance Threat which is currently present. Clicking on this opens the Compliance Threat explorer (equivalent to the Threat Editor), so we can view the details. This particular threat is described as “A Doctor accessing Genetic Data on a DB Server located in a Country Legal Space is at risk of non-compliance”.

Project Title: SHiELD  Contract No. GA 727301  http://project-shield.eu/
Some compliance threats encode the need to use security measures mandated by regulations, and these threats will have a control strategy containing the measures that should be used. In other cases (like the threat in Figure 42) the compliance threat encodes a regulatory constraint on the system architecture or structure. For these, no control strategies will be available, and the only option is to break the pattern (as indicated by the tool-tip next to the “Relations” section). Hovering over the “Doctor has access to Genetic Data” relation highlights this on the canvas (see Figure 43). Next, we delete the relation (via the red cross icon).

The relation then appears crossed out in the Relations list and disappears from the canvas (see Figure 44). We re-validate the model, as indicated. Once validated successfully, the Compliance Explorer indicates that this Compliance Threat has been resolved, by turning green and showing no further issues (see Figure 45).
3.2.8 Misbehaviour Sets

As described earlier in this document, Misbehaviour Sets are caused by threats; a Misbehaviour Set is the combination of a Misbehaviour on a specific Asset, e.g. “Overloaded” on “Host1”. During the validation process, Misbehaviour Sets are determined, along with their associated threats. These can be viewed and explored in System Security Modeller in a variety of ways.

Firstly, Misbehaviour Sets may be viewed in the asset details panel. After selecting an asset, the user must expand the Misbehaviours panel (see example in Figure 46):
Here, each asset misbehaviour is listed, along with its Impact level, which can be modified by the user before calculating risk levels, as discussed in Section 3.2.4. Also shown are the calculated Likelihood and Risk from this misbehaviour; these are coloured according to their severity. A misbehaviour may be examined in more detail by clicking on its name (label). This brings up the Misbehaviour Explorer, as shown in Figure 47.

The Misbehaviour Explorer shows the following details:

- **Name**: e.g. “Loss of Availability at WebService”
- **Impact, Likelihood and Risk**: as described above
- **Direct Causes**: threats which directly cause this Misbehaviour Set
- **Root Causes**: threats at the start of the threat/misbehaviour chain

The Misbehaviour Explorer helps the user to determine threat cause/effects. For example, they can click on a listed Direct Cause (threat), opening up the Threat Editor for this, then use this, in turn, to determine that threat’s cause (e.g. a misbehaviour on an asset). In this way, the threat/misbehaviour chain may be navigated until reaching the root cause. Alternatively, the user may examine Root Causes directly, via the bottom panel in the Misbehaviour Explorer. In either case, the idea is to find a threat which can be addressed by using security measures, thus breaking the path between the root cause and the troublesome misbehaviour. Such threats will have control strategies describing the control measures that can be used, and allowing the relevant controls (or control strategies) to be added to the design time model.

In the second half of the project we intend to add more of these ‘navigational’ features to help users decide which threats should be addressed to manage the worst case risk level in a system.
4 Updated Secure Design patterns

Deliverable D4.1 defined the design of the solution for this section (Figure 48). This section related to secure design patterns are focused on source code analysis. SHIELD project uses OpenNCP for connecting the different countries (National Contact Points), and this platform is implemented by using JAVA as the main programming language. Therefore, we need to analyse JAVA source code in order to identify threats defined by MITRE, SANS and OWASP. There are several existing tools performing static analysis of source code. Sonarqube (Figure 49) is an available source code tool (LGPLv3) which can be used for analysing and identifying security threats in source code.
In fact, the Sonarqube scanner performs a pattern matching based on non-compliant code examples. For example, if we want to check whether credentials for accessing a database are hard coded or not within the source code we need to check the following risky code chunk:

```
"aConn=DriverManager.getConnection("jdbc:mysql://localhost/test?"+"user=steve&password=blue");"
```

A potential solution would be to encrypt username and password. Similar problems are described in MITRE, CWE-798 (Use of Hard-coded Credentials - http://cwe.mitre.org/data/definitions/798), and MITRE, CWE-259 http://cwe.mitre.org/data/definitions/259. One potential solution will look as:

```java
String username = getEncryptedUser();
String password = getEncryptedPass();

"aConn=DriverManager.getConnection("jdbc:mysql://localhost/test?"+"user=" + username + "password="password");"
```

In order to provide a low hanging fruit prototype, we have implemented a dashboard (Figure 50) which summarises the current projects being developed in our development environment (Eclipse) as described in Figure 48. This dashboard helps us to identify the number of metrics used, the number of projects being analysed, the number of vulnerabilities and the number of issues to be solved.
Figure 50 – SHIELD source code analyser dashboard

Figure 51 provides a view for analysing source code, and a Pattern visualizer containing a set of security patterns. This first prototype helps you to navigate through source code of different projects, but we cannot select the secure patterns.

Figure 51 – Analysing source code

The current OpenNCP source code is composed by 23 Maven projects, and this code has some vulnerabilities which help us to identify which patterns are the most relevant in terms of quantity. Its potential impact is something we need to research on, but we have done a preliminary study. Basically, this study is based on the analysis of the OpenNCP source code, and we identified some patterns in the current source code that must be tackled. This study reveals some interesting security flaws described in Table 3. OpenNCP source code involves more than 146k lines of code, 588 bugs and 93 vulnerabilities.
### Table 3. Results of the OpenNCP source code scanner

<table>
<thead>
<tr>
<th>Module</th>
<th>Lines of Code</th>
<th>Bugs</th>
<th>Vulnerabilities</th>
<th>Code Smells</th>
<th>Duplications</th>
</tr>
</thead>
<tbody>
<tr>
<td>eHealth - OpenNCP Code</td>
<td>146k</td>
<td>588</td>
<td>93</td>
<td>5k</td>
<td>32.2%</td>
</tr>
<tr>
<td>assertion-validator</td>
<td>1.1k</td>
<td>0</td>
<td>0</td>
<td>123</td>
<td>7.2%</td>
</tr>
<tr>
<td>audit-manager</td>
<td>3.3k</td>
<td>47</td>
<td>48</td>
<td>646</td>
<td>2.6%</td>
</tr>
<tr>
<td>da-display-tool</td>
<td>4.1k</td>
<td>24</td>
<td>0</td>
<td>128</td>
<td>15.7%</td>
</tr>
<tr>
<td>configuration-manager</td>
<td>374</td>
<td>0</td>
<td>0</td>
<td>33</td>
<td>0.0%</td>
</tr>
<tr>
<td>consent-manager</td>
<td>2.5k</td>
<td>0</td>
<td>0</td>
<td>88</td>
<td>5.0%</td>
</tr>
<tr>
<td>data-model</td>
<td>82k</td>
<td>13</td>
<td>0</td>
<td>1.8k</td>
<td>41.4%</td>
</tr>
<tr>
<td>default-policy-manager</td>
<td>338</td>
<td>10</td>
<td>0</td>
<td>47</td>
<td>9.3%</td>
</tr>
<tr>
<td>e-sens-non-repudiation</td>
<td>7.2k</td>
<td>6</td>
<td>6</td>
<td>235</td>
<td>8.4%</td>
</tr>
<tr>
<td>eadc</td>
<td>1.2k</td>
<td>37</td>
<td>0</td>
<td>27</td>
<td>0.0%</td>
</tr>
<tr>
<td>openncp-gateway</td>
<td>4.6k</td>
<td>30</td>
<td>9</td>
<td>245</td>
<td>14.1%</td>
</tr>
<tr>
<td>openncp-portal</td>
<td>13k</td>
<td>205</td>
<td>5</td>
<td>647</td>
<td>12.7%</td>
</tr>
<tr>
<td>openncp-web-portal</td>
<td>11k</td>
<td>94</td>
<td>8</td>
<td>443</td>
<td>3.2%</td>
</tr>
<tr>
<td>security-manager</td>
<td>2.5k</td>
<td>41</td>
<td>14</td>
<td>152</td>
<td>21.1%</td>
</tr>
<tr>
<td>transformation-manager</td>
<td>2.9k</td>
<td>20</td>
<td>0</td>
<td>110</td>
<td>0.5%</td>
</tr>
<tr>
<td>trc-sts</td>
<td>712</td>
<td>0</td>
<td>0</td>
<td>18</td>
<td>25.3%</td>
</tr>
<tr>
<td>trc-sts-client</td>
<td>439</td>
<td>13</td>
<td>0</td>
<td>25</td>
<td>13.1%</td>
</tr>
<tr>
<td>tsam</td>
<td>1.7k</td>
<td>15</td>
<td>0</td>
<td>65</td>
<td>7.1%</td>
</tr>
<tr>
<td>tsam-sync</td>
<td>1.1k</td>
<td>0</td>
<td>0</td>
<td>16</td>
<td>12.4%</td>
</tr>
<tr>
<td>util</td>
<td>4.7k</td>
<td>33</td>
<td>3</td>
<td>192</td>
<td>3.5%</td>
</tr>
</tbody>
</table>

Figure 52 reflects on a histogram diagram which OpenNCP Maven projects have vulnerabilities, and their quantities. The projects related to security are those with the highest number of vulnerabilities.
In future we believe it should be possible to use this type of analysis to estimate how trustworthy processes defined in the high level system model will be, and feed this into the System Security Modeller analysis. This will be investigated in the second half of the project.

Finally, Table 4 summarises the analysis of CWE weaknesses identified within OpenNCP source code. These are the main patterns that are going to be included in our prototype.

### Table 4. 82 CWE weaknesses found within OpenNCP

<table>
<thead>
<tr>
<th>CWE</th>
<th>Occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td>MITRE, CWE-259 - Use of Hard-coded Password</td>
<td>2</td>
</tr>
<tr>
<td>MITRE, CWE-391 - Unchecked Error Condition</td>
<td>12</td>
</tr>
<tr>
<td>MITRE, CWE-397 - Declaration of Throws for Generic Exception</td>
<td>4</td>
</tr>
<tr>
<td>MITRE, CWE-459 - Incomplete Clean-up</td>
<td>1</td>
</tr>
<tr>
<td>MITRE, CWE-493 - Critical Public Variable Without Final Modifier</td>
<td>31</td>
</tr>
<tr>
<td>MITRE, CWE-500 - Public Static Field Not Marked Final</td>
<td>1</td>
</tr>
<tr>
<td>MITRE, CWE-582 - Array Declared Public, Final, and Static</td>
<td>2</td>
</tr>
<tr>
<td>MITRE, CWE-607 - Public Static Final Field References Mutable Object</td>
<td>2</td>
</tr>
<tr>
<td>MITRE, CWE-754 - Improper Check for Unusual Exceptional Conditions</td>
<td>10</td>
</tr>
<tr>
<td>MITRE, CWE-798 - Use of Hard-coded Credentials</td>
<td>17</td>
</tr>
</tbody>
</table>
5 Conclusions and next steps

We have continued the development of the WP4 tools as planned and added support for the modelling and matching of compliance threats. The threats themselves have been identified with the help of the use-case owners in WP6 and are based on current legislation. Furthermore, we have taken feedback from project partners to improve productivity of users using the “System Security Modeller” tool. The detection of security vulnerabilities in software code has also been implemented in a separate tool.

The main achievements in the first half of the project (including work reported in D4.1) are:

- extension of the System Security Modeller tool to capture regulatory compliance requirements by treating non-compliance as an additional type of security threat;
- extension of the associated Knowledge Base to include models for threats to compliance and threats that may require advanced security measures from SHIELD WP5;
- incorporation of a risk level calculation algorithm from a sister H2020 project, and extension of the SHIELD knowledge base to include scales for expressing the trustworthiness of assets, the likelihood of threats or misbehaviours, and the impact of misbehaviours;
- development of a tool for finding software vulnerabilities in the code used, and application of this tool to check for security issues in the OpenNCP software;
- support for an initial validation exercise in WP6.

For the next release, we’re planning to add more compliance threats, to provide a better coverage. We also plan to link the results of T4.3 back to T4.1 and T4.2 by enabling the modelling of vulnerabilities in the processes of a design-time system so that the risks incurred through the usage of vulnerable software becomes more visible to the user.

6 References


