TOOL SUPPORT FOR ACHIEVING QUALITATIVE SECURITY ASSESSMENTS OF CRITICAL INFRASTRUCTURES
The ESSAF Framework for Structured Qualitative Analysis

Nguyen HanhQuyen, Köster Friedrich, Klaas Michael, Brenner Walter
Institute of Informationmanagement, University of St. Gallen, Müller-Friedbergstr. 8, 9000 St. Gallen, Switzerland

Obermeier Sebastian, Brändle Markus
ABB Corporate Research, Segelhofstr. 1K, Post Box, 5405 Baden, Switzerland

Keywords: Security assessment, Critical infrastructures, System design, Security implementation, Collaboration support.

Abstract: Devices that are designed for the use in critical infrastructures demand a high level of security. Therefore, a consideration of cyber threats and security mechanisms should be done in an early state, at best at the product's design phase. In this paper, we present a security assessment method in addition to a support tool that allows the involved participants to conduct security assessments in a reproducible and standardized way. Special for our method is the focus on the collaboration of different domain experts at various abstraction levels, which is typical for critical infrastructure device assessments.

1 INTRODUCTION

As devices used in critical infrastructures rely more and more on embedded software and networking technology, cyber security becomes increasingly important. Thus, an early consideration of cyber security, at best in its design phase, helps to lower the security associated costs. Therefore, security assessments that are carried out during the product's design phase can, when carried out methodically and systematically, help system developers to untangle the complexity of a product's system in order to reveal security and implementation weaknesses at an early stage. Furthermore, structured security assessments provide the following benefits:

- Supporting and documenting architectural system designs including security implementations
- Enhancing analysis efforts
- Advancing communications between system developers, security experts and other stakeholders (e.g. managers, customers, etc.)

However, motivating a security assessment among product developers is a hard task as it is known to be tedious and error-prone. Security assessments require extensive and costly brainstorming sessions among involved parties who often do not have time to participate. Consequently, the need for a supporting tool arises, which offers structured guidance and collaboration to facilitate the assessment process. Furthermore, a support tool can provide to all involved parties a means to identify their roles and responsibilities within a security assessment.

2 CONTRIBUTIONS

In this paper we introduce the method ESSAF (Embedded System's Security Assessment Framework), which is based on a previous contribution (Koester et al., 2008). Here, we would like to differentiate between this paper and another research work of this conference which introduces a methodical approach towards collaborative security assessments of embedded systems. Beyond this contribution as well as other researches in the field

Hanh Quyen N., Friedrich K., Michael K., Walter B., Sebastian O. and Markus B. (2009).
TOOL SUPPORT FOR ACHIEVING QUALITATIVE SECURITY ASSESSMENTS OF CRITICAL INFRASTRUCTURES - The ESSAF Framework for Structured Qualitative Analysis.
In *Proceedings of the International Conference on Security and Cryptography*, pages 297-304
DOI: 10.5220/0002188202970304
Copyright © SciTePress

of security for embedded systems, we delineate the requirements for a supporting tool and introduce the ESSAF TOOL which is designed to assist system developers in achieving qualitative security assessments of critical infrastructures during design phase. This research work describes how the ESSAF TOOL alleviates complexities of system and security modelling processes and enables collaboration among stakeholders during the assessment process.

3 REQUIREMENTS FOR A SECURITY ASSESSMENT TOOL

We have collaborated with various system experts to identify crucial requirements that a security assessment tool for devices used in critical infrastructures should meet:

Collaboration Support (CS). In practice, knowledge about a system is oftentimes distributed among different stakeholders and the best way to achieve qualitative system analysis is to consolidate that knowledge. The supporting tool, therefore, needs to incorporate features to support a collaborative security assessment process that allows systematic documentation and evaluation of system needs and security implementations. Furthermore, the tool should recursively aggregate and consolidate design decisions that have been made by different people at different points of time. Thus, it is necessary that the tool establishes a role model and provides capabilities to document and to attribute inputs to their origin, in the way that it enables user and change tracking.

Abstraction Level (AL). The tool must be able to describe the system and security model of any critical infrastructure product. The best way to achieve this requirement is to examine system artifacts at high-level of granularity, i.e. on the implementation level where system assets are broken down into functions, storages, data and data flows. To ensure modeling flexibility, the tool needs to be able to aggregate and refine system elements recursively. Also, it must be able to associate meaningful semantics with elements at any level of abstraction. Meeting this requirement is essential to supporting iterative refinement and analysis of the system and security models. For example, it must be possible to coalesce several functions into one single function as well as to refine one vulnerability into multiple weaknesses.

The tool should also use graphical notations (like data flow diagrams) to illustrate the system model for better understandings and communications among the involved parties.

Assessment Mode (AM). The supporting tool needs to enable asynchronous assessment mode that allows different process steps to be executed, analyzed and communicated separately. For example, the tool must allow creating threats and vulnerabilities before creating the related assets and vice versa. This requirement is necessary to avoid deadlocks during the collaboration process among stakeholders from different domains.

Applicability in Design Phase (ADP). According to McGraw, it is more than twice as efficient to spend resources for securing software products in the design phase vs. the testing phase, and about half of security issues are caused by design flaws (McGraw, 2006). Thus, the tool should be applicable at the system design phase.

Validability/Plausibility (V/P). The tool support should incorporate abilities to enforce stakeholders to document reasoning and rationales about design decisions that they have made on the system during the assessment process. Meeting this requirement will add transparency to the decision making processes of users and is crucial to validating and ensuring the plausibility of a system's design and security implementation. It further enables participants to trace the "evolution" of system elements, to enhance comparing design alternatives or asking about the feasibility of security measures.

Data Requirements (DR). A system manufacturer often produces devices that can be used in very diverse settings. This makes it impossible to quantify security risks in terms of their probability and (monetary) impact. The supporting tool must be applicable without requiring information on concrete use cases or environment settings of the device. It should rather use quantifiable measures more closely related to the causes of risk in order to prioritize risk.

4 RELATED WORK

We have evaluated five representative tools of major security risk assessment methods, specifically Trike 1.1.2a (Saitta, 2005), Microsoft SDL Threat

Modelling Tool v3 (Microsoft, 2008), practical threat analysis tool v4.5 (PTA Technologies, 2007), CORAS Language Editor v2.0.b5 (Braber, 2007) and EBIOS v2 (EBIOS, 2005). The tools have been evaluated on four security assessments of different embedded devices to measure their compatibility degree towards the mentioned requirements. The targeted outputs have been reports on system assets, potential threats, identified vulnerabilities and suggestions on possible mitigations.

The feature-based qualitative case study has yielded that none of the evaluated tools sufficiently satisfies our needs:

CORAS Language Editor v2.0.b5. The CORAS tool is developed by the Norwegian research group SINTEF. CORAS Language Editor v2.0.b5 is a graph editor that is limited to graphical modelling of corporate assets using UML entities. It does not provide collaboration support and fails requirement AL and ADP.

EBIOS v2. The open source EBIOS tool has been issued by the Secrétariat Général de la Défense Nationale of France to improve the protection of national networks. The inflexible functionalities provided by EBIOS are only applicable for assessments of well known domains in fairly static contexts. They do not (sufficiently) answer to requirement CS, AL, AM, ADP and V/P.

Microsoft SDL Threat Modelling Tool v3. Recently Microsoft has introduced the SDL threat modeling tool (TMT) v3.0 that focuses on identifying security weaknesses in the implementation of a software product early in the design phase. TMT does not consider causes of threats (i.e. vulnerabilities), nor imparts capabilities to evolve security measures to mitigate them. Also, it fails requirement CS, AM and V/P.

Practical Threat Analysis Tool v4.5. The practical threat analysis (PTA) tool of the Eldan Software Systems Ltd. focuses on analyzing probabilistic risk potentials of threats. A key conceptual flaw of PTA is that it does not consider interconnections among assets and therefore falls short of identifying security weaknesses that can emerged from these connections. The PTA also does not answer to requirement ADP and DR since it requires non-trivial security knowledge or at last very good heuristic experiences to quantify risk potentials of a threat in general and its damage potentials on an asset in particular, which is hard to determine for new assets. The PTA also does not provide means to meet requirement CS and V/P.

Trike 1.1.2a. The Trike tool is developed by and under the copyright of Paul Saitta, Brenda Larcom and Michael Eddington. Trike's key functionality is limited to the generation of threats without taking into account threat causes or considerations for security improvements. Trike also fails to meet requirement CS, AL, AM, ADP and V/P.

The most important evaluation results are charted in Table 1.

5 ESSAF METHOD

The main goal of the ESSAF method is to enhance the security assessment process that is collaboratively carried out by the stakeholders, who are involved in the production of an embedded system.

Based on previous research (Koester et al., 2008), we have derived three main phases of a cyclic security assessment process (Figure 1) which aims at continuously refining design architectures and revising security implementations of embedded devices. These phases are specified below:

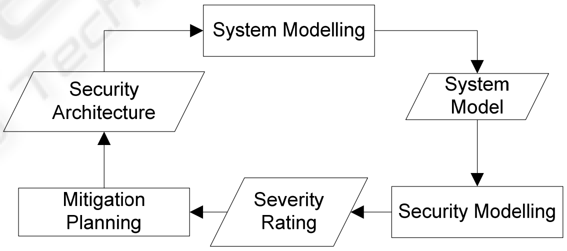

Figure 1: The cyclic security assessment process defined by the ESSAF method aims at refining design architectures and revising security implementations.

System Modelling Phase. The system modeling phase aims at collecting system assets (functions, storage and data) and modeling security characteristics of the system. To do so, it particularly focuses on identifying security measures and security objectives of system assets. Security measures constitute the security of the assets themselves and are directly related to security objectives, as they reflect whether or not and how significant security objectives are secured. The main use of the system model is to monitor the evolution of assets and their interconnections (data flows) during design phase. Therefore, documenting rationales and assumptions on design decisions are

Table 1: The evaluation results have yielded that none of the evaluated tools sufficiently meet the requirements for a supporting tool of the ESSAF method.

	Collaboration support	*Abstraction level*	*Assessment mode*	*Applicability in the design phase*	*Validability /plausibility*	*Data requirements*
CORAS v2.0.3	○ Does not provide means for collaboration support.	○ Is designed for assessments on corporate level only.	● Allows asynchronous assessment mode.	○ Only applicable when system is well known.	◖ Provides semantic validation only.	◖ No software support
EBIOS v2	◖ Provides user management for synchronous team working.	○ Is designed for assessments on corporate level only.	○ Enforces strict dependencies among different assessment steps.	○ Only applicable when the assessment domain is well known.	◖ Provides validation for security objectives only.	●No probabilistic data required for risk quantifications.
Microsoft SDL Threat Modeling Tool v3.0	◖ Allows documentation of design decisions, but does not impart user roles.	◖ Is designed for software evaluation. But does not consider threat causes or mitigation strategies.	○ Enforces strict dependencies among system modeling and threat identification.	● Is applicable during design phase.	◖ Provides data flow diagram validation only.	● No probabilistic data required for risk quantifications.
Trike 1.1.2a	○ Does not provide means for collaboration support.	◖ The definitions of assets are too restricted for software security assessments.	○ Enforces strict dependencies among asset identification and threat identification	◖ Can be applied during design phase, but still requires too many unknown asset information.	○ No validation	● No probabilistic data required for risk quantifications.
Practical Threat Analysis Tool 4.5	○ Does not provide means for collaboration support.	◖Allows free definitions of assets but does not consider asset interactions.	●Allows to freely switching between different assessment steps.	○ Only applicable when system and system behavior are well known.	○ No validation	○ Requires probabilistic valuation of risks and damage potentials.

● = Requirements fulfilled ◖ = Requirements partly fulfilled ○ = Requirements not fulfilled

essential for enabling verification and plausibility of system development and security implementations.

Security Modelling Phase. This phase focuses on identifying threats of security objectives and detecting vulnerabilities of system assets. The main output of the security modelling phase is the severity rating of identified threats and vulnerabilities. The rating is determined by stakeholders based on the complementary interdependencies among threats and vulnerabilities. For example, a vulnerability that may cause the realization of a threat can be considered more severe than a vulnerability that does not. Additionally, the rating lays the groundwork for selecting mitigations in the mitigation planning phase.

Mitigation Planning Phase. The mitigation planning phase is the process of identifying appropriate security architectures that are compositions of suggestions on changes to the system's design and improvements for the security implementation. It can entail additional security measures and assets in order to mitigate identified vulnerabilities. The creation of new security architectures induces further iterations of the cyclic assessment process. The diversity of security architectures enhances comparison of design alternatives and supports determining conformance between security needs and system scopes.

A supporting tool of the ESSAF method must support the processes of these three phases in order to aid in embedded system design as a creative activity and embedded system design as an analysis activity. To achieve this it must support collaborative team-working environments where all stakeholders can use it in a joint effort.

6 ESSAF TOOL

In this section the authors will give a comprehensive description of the ESSAF TOOL that has been designed to implement the security assessment process specified by the ESSAF method and is intended to be used in daily works of system developers of critical infrastructures. The ESSAF TOOL aims at aiding stakeholders in communicating and exchanging experiences on system designs. It is designed to convey different points of views and to trigger discussions. It enables asynchronous collaborative team-working, motivates knowledge sharing and enhances communications between system developers, security experts and other stakeholders.

The ESSAF TOOL's architecture incorporates three main features: the *assessment engine* for collecting and modeling system assets and security elements, *validation and plausibility* capabilities for verifying design decisions, and *collaboration support* for enabling asynchronous team-working among different stakeholders during the production process.

6.1 Assessment Engine

The assessment engine encompasses functionalities to model and to evaluate system assets and security implementations. Most often, stakeholders are uncertain about how and where to efficiently start an assessment. They may wonder what information is needed and how to semantically consolidate them so that other stakeholders can collaboratively join in the assessment process. To answer to these needs, the ESSAF TOOL offers well-structured GUIs that offer at a glance all information that are related to and required for an assessment element. Among other element attributes, the GUI masks comprehensively depict all changes that have been performed on and rationales that have been made for an assessment element.

To answer to the issue of how to start with the assessment process, the ESSAF TOOL provides capabilities of guiding users through the entire assessment process without restricting them: Users can choose to go through the assessment process on their own (recommended for advanced users) or to follow instructions given by a wizard (recommended for untrained users). Using the wizard, stakeholders are guided to process the assessment in the order that has been defined by the method (section 5). When not in wizard-mode, ESSAF TOOL allows users to entering any information they can provide at any point of time. To consolidate entered information for further analysis, the ESSAF TOOL assists users in determining dependencies among entered assets, threats and vulnerabilities.

6.1.1 System Modelling

In accordance with the underlying ESSAF method, the ESSAF TOOL automatically guides users through three steps of modeling the following system entities: system assets, security implementation of these assets and interactive connectivity among these assets.

Modelling System Assets. The asset modelling process integrated in the ESSAF TOOL encompasses the steps of modeling atomic assets and interdependencies among them (e.g. the data "User credential" is stored in the storage "Flash Card"). To enhance modelling flexibilities, the ESSAF TOOL allows to delay the modelling of asset interdependencies (e.g. it is not required to specify that the data "User credential" is stored in the storage "Flash Card" at the creation time of this data and vice versa). The ESSAF TOOL also incorporates a graph editor that aids in modelling and visualizing system models. The editor represents system models as multi graphs of assets and data flows. It can represent single assets, clusters of assets or entire sub-systems in this notation. Clusters are maintained as a set of arbitrary graphs which can be treated as one atomic asset-node.

Determining Security Implementations of Assets. To identify the security implementation of the target system, the ESSAF TOOL focuses on modeling security objectives and security measures of system assets. In terms of basic features, the ESSAF TOOL automatically assigns a set of predefined security objectives to each asset type (functions, storage and data). For each individual asset it requires users to specify whether or not the given security objectives are relevant and to document textual reasons for these decisions. On the other hand, Security measures are functionalities of assets and can ensure certain security objectives (e.g. the security measure "Encryption" can ensure the security objective "Integrity"). By encouraging users to determine interdependencies among security measures and security objectives and by enforcing documentations of rationales on this matter, ESSAF TOOL increases the plausibility of security needs and actual implementations of the system.

Modelling Data Flows. From the discussions with system experts we have concluded that beside assets, data flows are also essential parts of the system model, as they convey underlying design intent and interactive principles of the system. Therefore, the ESSAF TOOL puts great efforts in guiding users to comprehensively model data flows. In the ESSAF TOOL a data flow is modeled as a connection that transmits a data from one function to another function or storage (e.g. "User credentials" are transmitted from "FTP" to "File Storage"). To ensure that users consider security-relevant issues of this connection, ESSAF TOOL incorporates capabilities to suggest security objectives that need to be met by the involved assets (e.g. security objective "Confidentiality" is relevant for the data "User credentials" and needs to be ensured). Furthermore, ESSAF TOOL puts special attention on observing data flows that leave the system boundary to detect and to warn about possible communication breakdowns and security leaks that may occur during these interactions with (unknown) external entities. In this case, ESSAF TOOL automatically points out the assets that are made vulnerable through these connections.

In the ESSAF TOOL collections of assets can be coalesced into a single cluster. Clusters of assets are considered usage scenarios. The ESSAF TOOL ensures that the connectivity constraints of the individual assets are met by the coalesced cluster. It also ensures that the semantics of the grouped assets are accurately represented by the cluster. Any asset within any cluster can be connected per data flows to assets in one or more of the other clusters. Because of the semantic foundations attributed to each cluster, any cluster of assets can be treated exactly as though it were an atomic asset. This sort of flexibility is of prime importance to architectural mining and understanding: being able to flexibly coalesce and to refine system components based upon user-defined level of granularity.

6.1.2 Security Modelling

In the ESSAF TOOL the most important steps of modeling security issues are:

Modelling Threats and Vulnerabilities. The ESSAF TOOL supports the process of identifying threats and vulnerabilities by providing numerous questions that make users concern about security issues which may arise for certain types of assets and their security objectives (e.g."How could the security objective 'Confidentiality' of your asset be threatened?", "Which weaknesses of this asset are already known?", etc.). The severity rating of vulnerabilities and threats are dependent on their reciprocal connectivity.

Although the ESSAF TOOL is not designed to automatically establish these connections, it does facilitate the rating process by highlighting the related threats and vulnerabilities. Additionally, the severity of vulnerabilities can be influenced by the existence of mitigations that aim at solving them. Consequently, the ESSAF TOOL allows to discerning and to ranking types of (mitigated) vulnerabilities for severity rating purposes.

Establishing Correlations to Assets. As defined by the ESSAF method, threats characterize possible ways to endanger certain security objectives, while vulnerabilities constitute implementation weaknesses of assets. Because ESSAF TOOL allows to creating threats and vulnerabilities independently from related assets, the step of establishing correlations is necessary. This process is carried out by referring threats to endangered security objectives and vulnerabilities to related assets (and vice versa). The ESSAF TOOL strongly recommends users to document rationales for these actions.

6.1.3 Mitigation Planning Phase

In the ESSAF TOOL, realizing mitigations implies a chain of changes to the system's design and security implementation. To enhance verification and plausibility of these changes, the ESSAF TOOL allows to tracing security measures and system adjustments that have been induced in accordance with these mitigation suggestions. For the same reason, it strongly recommends users to comprehensively document rationales they have made for each mitigation suggestions, as well as to justify why they have chosen certain mitigations opposed to others.

As mitigations are merely suggestions on possible security measures (e.g. integrating new security measures for existing assets, creating new assets, etc.), the ESSAF TOOL is not able to decide on which mitigations should be realized. Instead, it supports users in making their own decisions on the matter. It does so by requesting them to consider and to verify targeted mitigation measures in regard of correctness, redundancy and feasibility. Thus, the ESSAF TOOL can support increasing rationality and consistency of system design and feasibility of security implementation.

6.2 Validation and Plausibility

To maintain transparency and hence to ensure plausibility of design decisions, the engine integrates a logging function that elaborately manifests user activities and circumstantial changes of all assessment elements. ESSAF TOOL also requires that users document rationales and justify actions to be performed on an element. These rationales are attributed to the edited element and accessible to other stakeholders for consultation. In that way, a user is always able to trace changes and to verify whether the changes are correct or correctly implemented in accordance with the given rationales. For example, when deciding that a Flash Card needs not provide constant availability, the user has to specify his rationale for this decision e.g. that this storage may be substituted by another resource that is constantly available. The ESSAF TOOL has proved that enforcing stakeholders to communicate and document design and modeling decisions, supports them achieving qualitative security assessments.

Further benefits of the exchange of transparent design decisions that have been distilled since the ESSAF TOOL's inception are: enhanced accuracy of system models, verified feasible security models, and increased design alternatives.

6.3 Collaboration Support

The ESSAF TOOL is constructed to support collaboration among multiple stakeholders. To achieve this goal it incorporates the following features.

Change Notification. The notification mechanism of the ESSAF TOOL provides capabilities to highlight and to trace all changes of assessments, particularly changes that have been performed by other stakeholders since the last login of the current user. Also, it is able to determine and to highlight elements that need attentions (e.g. completion, adjustment, revision, etc.) or are subject to verification and validation.

User Management System. The ESSAF TOOL incorporates an extensive user management system to support change tracking and to determine which users are authorized to work on which assessments. Using the ESSAF TOOL, stakeholders can (or have to) be invited to join an assessment process by the assessment owner and are registered for one assessment only. In order to participate in other assessments they have to repeat the registration/invitation procedure. Assessments are only accessible and distributed among authorized users to prevent disclosure of confidential corporate information that may be incorporated within a device.

Assessment Management. In the ESSAF TOOL, assessments are maintained as an independent closed system that cannot be influenced by other assessments. Therefore, the ESSAF TOOL does not permit elements of one assessment to be transferred to another assessment. In that way, the ESSAF TOOL avoids unintended dependencies and deadlocks among assessments of different systems. For sharing purposes, assessments are centrally stored as XML-files on a server. Remote access is only granted to authorised ESSAF TOOL-users. Any access requests by unauthorized users or other software-tools are denied to ensure the integrity and compatibility of these assessments. To avoid concurrent modification conflicts a versioning system is connected upstream to coordinate assessment assignments: Whenever an assessment is assigned to a user, it is changed to in-editing-mode and unavailable to other users until the current editor has finished or the reserved locking time has passed.

Providing these features, the ESSAF TOOL ensures confidentiality of assessment contents in a flexible collaborative working environment without versioning and editing conflicts.

6.4 Evaluation of the ESSAF TOOL

To prove its practicability, ESSAF TOOL has been evaluated against the significant criteria specified in section 3. The evaluation results (charted in Table 2) show that by its incorporated functionalities ESSAF TOOL is capable of supporting asynchronous collaborative security assessment processes at flexible levels of granularity while enabling

Table 2: The evaluation results have proved that the ESSAF TOOL succeeded in implementing the security assessment process proposed by the ESSAF method while answering to the significant requirements for a supporting tool.

	Collaboration support	*Abstraction level*	*Assessment mode*	*Applicability in the design phase*	*Validability /plausibility*	*Data requirements*
ESSAF TOOL	● Provides user management with integrated user roles and change notification features for collaboration support.	● Is able to flexibly break down software implementations of critical infrastructures into atomic assets as well as clusters of assets.	● Allows to asynchronously carrying out different phases of the assessment process at different points of time.	● Is constructed to be applicable in the design phase and supports comparing competing design alternatives.	●Provides semantic validation and verification of design decisions.	● Does not require data for probabilistic risk quantification, but uses interdependencies among threats, vulnerabilities and mitigations to rate severity.

● = Requirements fulfilled ◖ = Requirements partly fulfilled ○ = Requirements not fulfilled

validability and plausibility of gathered information and abstaining from requiring probabilistic data for risk quantification.

7 CONCLUSIONS

This research work has derived significant requirements for a supporting tool of the ESSAF method that defines the steps for a security assessment process of embedded systems. This paper has demonstrated that existing tools do not sufficiently meet the significant requirements to achieve qualitative security assessments of critical infrastructures. Consequently, the ESSAF TOOL has been introduced to support the ESSAF method while answering to these requirements.

The ESSAF TOOL has proved to be applicable in the design phase and supportive in collaborative team-workings among different stakeholders from different domains. To summarize, we consider the ESSAF TOOL a significant step towards structured and guided security assessments for critical infrastructure devices.

Further development of the ESSAF TOOL aims at integrating a knowledge base infrastructure for enabling reuse and enhancement of system design elements and security solutions. The focus is on providing a central platform for consultation by stakeholders to facilitate information exchange during collaborative security assessments and design processes.

ACKNOWLEDGEMENTS

This work was supported by the Swiss Confederation's innovation promotion agency CTI.

REFERENCES

Braber, F., Lund, M., Seehusen, F., Stolen, K., and Vraalsen, F., 2007. CORAS Language Editor v2.0.b5, http://coras.sourceforge.net

Secretariat General de la Defense Nationale, 2005. EBIOS: Expression of Needs and Identification of Security Objectives, http://www.ssi.gouv.fr/en/confidence/ebiospresentation.html

Microsoft, 2008. SDL Threat Modeling Tool v3.0, http://msdn.microsoft.com/en-us/security/dd206731.aspx

Koester, F., Nguyen, H. Q., Klaas, M., Braendler, M., Naedele, M., and Brenner, W., 2008. ESSAM: A Method for Security Assessments by Embedded Systems Manufacturers, In: *3rd International Workshop on Critical Information Infrastructures Security*, Frascati (Rome), Italy.

PTA Technologies, 2007. PTA Risk Assessment Tool, http://www.ptatechnologies.com/.

Saitta, E., Larcom, B., and Eddington, M., 2003-2005. Trike v1.1.2a, http://www.octotrike.org

SCITEPRESS Science and Technology Publications