# Aligning the Concepts of Risk, Security and Privacy towards the design of Secure Intelligent Transport Systems

Vasiliki Diamantopoulou[1], Christos Kalloniatis[2], Christos Lyvas[1], Konstantinos Maliatsos[1], Matthieu Gay[3], Athanasios Kanatas[1], and Costas Lambrinoudakis[1]

[1] Department of Digital Systems, University of Piraeus, 150 Androutsou St., 18532 Piraeus, Greece {vdiamant,clyvas,kmaliat,kanatas,clam}@unipi.gr
[2] Department of Cultural Technology and Communication, University of the Aegean, University Hill, 81100 Lesvos, Greece chkallon@aegean.gr
[3] Airbus CyberSecurity, Metapole, 1 boulevard Jean Moulin, Elancourt, 78996, France
matthieu.gay@airbus.com

**Abstract.** Intelligent Transport Systems (ITS) play a key role in our daily activities. ITS development over the last decades has been based on the rapid evolution of information technologies, which include processing capabilities, availability of hardware and communication technologies. Moreover, ITS use Information and Communication Technologies (ICT) to improve sustainability, efficiency, innovation and safety of transportation networks helping towards better management of transportation networks with the use of advanced technologies, which facilitate monitoring, and management of information. However, as the development of ITS services increases so does the users' awareness regarding the degree of trust that they show on adopting this kind of services. The later has brought to light several security and privacy concerns that ITS analysts should consider when implementing various IT related services. This paper moves into this direction by identifying how risk analysis can interact with security and privacy requirements engineering world, in order to provide a holistic approach for reasoning about security and privacy in such complex environments like ITS systems. The key contribution of the paper is the conceptual alignment of three well-known methods (EBIOS, Secure Tropos and PriS) as the first step towards the design of a complete assurance framework that will assist analysts in designing safe and trustworthy ITS services.

**Keywords:** Intelligent Transport Systems · Risk Analysis Methodologies · Security Requirements Engineering Methodologies · Privacy Requirements Engineering Methodologies.

## 1 Introduction

The way humans, smart things and engineered systems interact and exchange information has dramatically changed due to the recent advances in communications, computation, networking, software, and hardware technologies. The

paradigm of *Connected Vehicles* constitutes a major technology and paradigm shift in the automotive industry, where enabling technologies and concepts of networked ICT, Internet-of-Things (IoT) and Cyber-Physical Systems (CPS) introduce new services and applications that will dramatically change driver-vehicle interaction. Based on a report that EC published [3], in the near future, the self-driving vehicles' market is expected to grow exponentially, developing profits of up to €620 billion by 2025 for the EU automotive industry. The benefits from these technological achievements are quite many [30], such as the transformation of roads to safer ones, the protection of the environment, the improvement of accessibility for disable people, the creation of new job positions and, consequently, the economic growth, to name a few.

However, autonomous driving rises a number of challenges that the scientific community, in cooperation with industry, has to overcome. Road safety, liability issues, data processing, and the necessary infrastructure are some that have been already identified in the early stages of the progression to the full automation of connected vehicles. Staying in the direction of the identification of challenges, a recent report of the European Commission [10] highlights the importance of finding the right balance in sharing only the appropriate amount of public and private data. As the market of the driverless vehicles increases [23], security research in this field will play a key role. Connected vehicles offer enormous opportunities for innovative features and services that in turn increase vehicles' cyber attack surface. Research in this area [31, 39, 21] has revealed that connected vehicles are prone to attacks due to the increased trend of high connected ICT, IoT and cloud services introduced.

Towards the direction of filling the aforementioned gaps, the ultimate goal of our work is to build a security assurance framework able to support connected vehicular technology, by addressing the safety, security and privacy of the handled data. This framework will be based on three well-established methodologies, each one focusing on addressing specific requirements, namely EBIOS [1], Secure Tropos [27], and PriS [18]. For this reason, in this study we present the first step towards the development of this framework, which is about the identification of the concepts shared in these three methodologies. In order to provide a more efficient design of the unified framework, an alignment of the EBIOS concepts with the concepts of Secure Tropos and PriS is important in order to identify any conceptual conflicts or any similarities in the terms used. Since Secure Tropos and PriS have their origins in the Software Engineering world [15], there was no need to align their concepts as well.

The rest of this paper is organised as follows: Section 2 presents related work regarding the three research areas that we examine. Section 3 presents the baseline of our work, by analysing all three methodologies that will allow us to align the concepts of the examined methodologies. Section 4 describes the outcome of this analysis, focusing on the common concepts of the analysed methodologies. Finally, Section 5 concludes the paper by raising issues for further research.

## 2 Literature Review

One of the novel aspects of the security assurance framework that we aim to develop, is that it integrates three different research areas, i.e. risk analysis, security requirements engineering, and the area of privacy requirements engineering. For this reason, since, to the best of our knowledge, there is no other integrated method that combines these three areas, in this section, we focus our literature review on these three areas separately.

**Risk Analysis** In the area of risk analysis, OCTAVE methodology [2] focuses on activities, threats, and vulnerabilities. Its main concept is self-direction, which means that people within the organisation must practice information security risk assessment [22]. The OCTAVE approach has three stages, each of which is divided into processes. Each process has certain activities that must be completed, and within each of these activities, the different phases must be taken to achieve the desired results.

CORAS [35] was developed using information society technologies. One of its main objectives is to develop a structure that uses the methods of risk analysis, semi-formal methods for object-oriented modelling, and computer tools for an accurate and unambiguous assessment of risk, and efficient critical safety systems [13]. The methodology is based on Unified Modelling Language (UML), a language that uses diagrams to illustrate relationships and dependencies between users and the environment in which they work.

The CCTA Risk Analysis and Management Method (CRAMM) [37] is a qualitative risk analysis and management tool. It calculates/estimates risk for each group of assets versus the threats to which it is vulnerable on a scale of 1 to 7, utilizing a risk matrix with the default values, by comparing it with the activity level of threat and vulnerability.

Compared to the review conducted in risk analysis area, EBIOS is an adequate and industrially validated tool to start the study since it assists analysts by guiding them in the early steps of the system design, especially for defining system's security objectives [29].

**Security Requirements Engineering** In the area of security requirements engineering, the authors of [33] propose Model Oriented Security Requirements Engineering (MOSRE) framework for Web Applications which considers security requirements at the early stages of the development process. It covers all phases of requirements engineering and suggests the specification of the security requirements in addition to the specification of systems requirements. The objectives, stakeholders, and assets of the Web application are identified during the inception phase. The final security requirements are elicited after a sequence of actions that include the identification - categorisation - prioritisation of threats and system vulnerabilities the risk assessment process, the analysis and modelling, and finally the categorisation - prioritisation - validation of the final security requirements.

SQUARE (Security Quality Requirements Engineering) methodology [24] is a risk-driven method that supports the elicitation, categorisation, prioritisation and inspection of the security requirements through a number of specific steps.

It also supports the performance of risk assessment to assess the tolerance of a system against possible threats. The method outputs all the necessary security requirements that are essential for the satisfaction of the security goals of a system. The methodology introduces the concepts of security goal, threat, and risk, but does not consider the assets and the vulnerabilities of a system. All the required security requirements should be identified by the requirements engineering team and the relevant stakeholders.

Another approach is the Security Requirements Engineering Framework (SREF) [14] which enables the elicitation and analysis of security requirements. This framework includes four stages. Firstly, it identifies functional requirements and afterwards, the security goals. Continuing, it identifies the security requirements of the functional requirements. Each security requirement satisfies one or more security goals. After these steps, the framework decides if the system satisfies the security requirements. The authors introduced an asset-based approach for the elicitation of security goals from business process models which are then translated into security requirements.

In [11, 12] the authors propose the Problem-based Security Requirements Elicitation (PresSuRE) Methodology that facilitates the identification of security needs during requirements analysis of software systems. More specifically, it provides a computer security threat recognition and then the development of security requirements. This methodology uses problem diagrams to support the modelling of functional requirements. Firstly, based on its contents, this methodology identifies system's assets and the rights of authorised entities. Then, it determines possible attackers and their abilities. Based on these steps, PresSuRE generates graphs which depict threats on system's assets. Every functional requirement of each asset is related with possible threats and security requirements.

Compared to the methodologies presented in this sub-section, Secure Tropos offers a more advanced tool for modelling, while the programming language used for the development of the tool is easily extended. Moreover, the methodological approach can be easily aligned with a risk-based approach. Finally, it combines actor and goal-based modelling, which is very important for the alignment of the common concepts of the three examined areas.

**Privacy Requirements Engineering** In the area of privacy requirements engineering, in [7] the authors present LINDDUN, a privacy threat analysis framework which, in its first release, aimed at the elicitation and fulfilment of privacy requirements in software-based systems. The process that LINDDUN follows is that a data flow diagram (DFD) of the system is designed and then the identified privacy threats are related to DFD elements. Privacy threat trees and misuse cases are used for the collection of threat scenarios that might affect the system. Moreover, this methodology supports the elicitation of the final privacy requirements and the selection of appropriate privacy enhancing technologies. The final stage of this methodology is the prioritisation and validation of privacy threat through risk assessment.

Next, in [34] the authors adopt the concepts of privacy-by-policy and privacy-by-architecture, and propose a three-sphere model of user privacy concerns, re-

lating it to system operations (i.e. data transfer, storage and processing). Additionally, the Modelling and Analysis of Privacy-aware Systems (MAPaS) framework [6] is a framework for modelling requirements for privacy-aware systems. The ABC4Trust project [32] protects privacy in identity management systems.

Compared to the methodologies presented in this sub-section, the PriS method is one of the oldest and mostly evaluated privacy-by-design methodologies, while it is successfully used for the elicitation and modelling of privacy requirements in traditional and cloud-based systems.

Finally, on a conceptual level the Secure Tropos and PriS methods are already successfully tested under a unified framework [28].

## 3 Background Analysis

This section presents the methodologies that we will rely upon, in order to develop an enhanced security assurance framework, able to support connected vehicular technology, by addressing safety, security and privacy of the handled data. More specifically, the methodology for the risk analysis is EBIOS, for the identification of security requirements, we present Secure Tropos methodology and finally, for the identification of privacy requirements, we present PriS methodology.

### 3.1 Risk Analysis

EBIOS (English: Expression of needs and identification of security objectives) is the risk analysis methodology created by the french Agence Nationale de la Sécurité des Systèmes d' Information (ANSSI) (English: National Cybersecurity Agency of France). A risk analysis method identifies the critical part of the system and their corresponding threats in order to evaluate the risk for these assets and then the proper security objectives regarding the evaluated risks. EBIOS is composed of five steps and offers many advantages, particularly the flexibility, quickness besides the fact that it is a proven methodology that has been used in several risk assessments and that it is compatible with the ISO 27005 risk analysis phase.

During the first step, *Circumstantial study*, the analyst can define the perimeter (boundaries) of the study. A global vision of the components and communications between components will be clarified. At this step, the following data will be collected and formalised (non-exhaustive list):

- Essentials assets in a connected vehicle system
- Functional description of components and relations between components
- Security issues that need to be addressed by the study
- Assumptions made if appropriate
- Existing security rules (law and regulation, existing rules in other studies)
- Potential constraints (internal or external) that might be imposed from the specific under examination system

At the end of this step, a clear vision of the components and the links between them will be formalised.

The second step, namely *Expression of security needs*, contributes to risk estimation and definition of risk criteria. The expression of security needs will be performed based on scale of needs. Security criteria and hypothetic impacts will be stated. Security needs will be associated with each essential component by taking into account the security criteria. A security needs report will be the output of this step. Next, the *Threat study and modelling* step follows, where the threats affecting the connected vehicle systems are studied. The threats are specific to the connected vehicles. There will be no dependencies between these threats and the security needs collected in the previous step. The list of the pertinent threats and the type of attacks will be the main outputs of this step.

Step 4 follows, entitled *Identification of security objectives*. The purpose of this step is to evaluate the risks affecting the connected vehicle environment. The security objective is highlighted by comparing the threats with security needs. The security objectives will contain the security requirements fulfilled in the development of secure connected vehicle system (or component).

The final step, Step 5 *Determination of security requirements*, brings an answer to the question how the security objectives will be achieved.

### 3.2 Security Requirements Engineering Analysis

Secure Tropos [27] is a security requirements engineering methodology that supports elicitation and analysis of security requirements. It is based on the principle that security should be analysed and considered from the early stages of the software system development process, and not added as an afterthought. To support that approach, the methodology provides a modelling language, a security-aware process, and a set of automated processes to support the analysis and consideration of security from the early stages of the development process. The Secure Tropos language consists of a set of concepts from the requirements engineering domain, and in particular Goal-Oriented Requirements Engineering [36, 4], such as actor, goal, plan, and dependency, which are enriched with concepts from security engineering, such as security constraint, secure plan, and attacks. This methodology closely follows the software development life-cycle, i.e. capturing of early requirements, late requirements, architectural design, detailed design, and finally, implementation. Thus, it allows the developer to create and refine models, starting from the system-as-it-is, in order to finally develop the system-to-be, during the analysis and design stage [9].

**Concepts Description**

Secure Tropos combines concepts from requirements engineering for representing general concepts and security engineering for representing security-oriented concepts [25].

A (hard) *Goal* [38] represents a condition in the world that an actor would like to achieve. In other words, goals represent actors' strategic interests. In Tropos, the concept of a hard-goal (simply goal hereafter) is differentiated from the concept of soft-goal.

A *Soft-Goal* is used to capture non-functional requirements of the system, and unlike a (hard) goal, it does not have clear criteria for deciding whether it is satisfied or not and therefore it is subject to interpretation [38]. For instance, an example of a soft-goal is the "system should be scalable". According to Chung et al. [5], the difference between a goal and a soft-goal is underlined by saying that goals are satisfied whereas soft-goals are satisfied under specific circumstances.

An *Actor* represents an entity that has intentionality and strategic goals within the multi-agent system or within its organisational setting. An actor can be human, a system, or an organisation.

A *Plan* [4] represents, at an abstract level, a way of doing something. The fulfilment of a task can be a mean for satisfying a goal, or for contributing towards the satisfying of a soft-goal. In Tropos different (alternative) tasks, that actors might employ to achieve their goals, are modelled. Therefore, developers can reason about the different ways that actors can achieve their goals and choose the best one.

A *Resource* [4] presents a physical or informational entity that one of the actors requires. The main concern when dealing with resources is whether the resource is available and who is responsible for its delivery.

A *Dependency* [38] between two actors represents that one actor depends on the other to attain some goal, execute a task, or deliver a resource. The depending actor is called the depender and the actor who is depended upon is called the dependee. The type of the dependency describes the nature of an agreement (called dependum) between dependee and depender. Goal dependencies represent delegation of responsibility for fulfilling a goal. Soft-goal dependencies are similar to goal dependencies, but their fulfilment cannot be defined precisely whereas task dependencies are used in situations where the dependee is required to perform a given activity. By depending on the dependee for the dependum, the depender is able to achieve goals that it is otherwise unable to achieve on their own, or not as easily or not as well [38]. On the other hand, the depender becomes vulnerable, since if the dependee fails to deliver the dependum, the depender is affected in their aim to achieve their goals.

A *Secure Dependency* [28] introduces one or more Security Constraint(s) that must be fulfilled for the dependency to be valid. In the Secure Tropos methodology we distinguish among three types of secure dependencies: dependee secure dependency, depender secure dependency, and double secure dependency. In terms of the modelling language, different Secure Dependency types are defined using depender and dependee attributes of Security Constraints.

A *Security Constraint* is used to represent security requirements. A Security Constraint is a specialisation of the concept of constraint. In the context of software engineering, a constraint is usually defined as a restriction that can influence the analysis and design of a software system under development by restricting some alternative design solutions, by conflicting with some of the requirements of the system, or by refining some of the systems objectives. In other words, constraints can represent a set of restrictions that do not permit specific actions to be taken or prevent certain objectives from being achieved. Constraints are

often integrated in the specification of existing textual descriptions. However, this approach can often lead to misunderstandings and an unclear definition of a constraint and its role in the development process. Consequently, this results in errors in the very early development stages that propagate to the later stages of the development process, causing many problems when discovered; if they are discovered. Therefore, in the Secure Tropos modelling language, security constraints are handled as a separate concept. To this end, the concept of security constraint has been defined within the context of Secure Tropos as: A security condition imposed to an actor that restricts achievement of an actor's goals, execution of plans or availability of resources. Security constraints are outside the control of an actor. This means that, differently than goals, security constraints are not conditions that an actor wishes to introduce but it is forced to introduce.

A *Vulnerability* [28] is defined as a weakness, in terms of security and privacy, that exists in a resource, an actor and/or a goal. Vulnerabilities are exploited by threats, as an attack or incident within a specific context.

A *Threat* [28] represents circumstances that have the potential to cause loss; or a problem that can put in danger the security features of the system.

Threats can be operationalised by different attack methods, each exploiting a number of system vulnerabilities. An *Attack Method* [26] in Secure Tropos is an action aiming to cause a potential violation of security in the system.

*Security Mechanisms* [26] represent security methods for helping towards the satisfaction of the security objectives. Some of these methods are able to prevent security attacks, whereas others are able only to detect security breaches. It must be noted that further analysis of some security mechanisms is required to allow developers to identify possible security sub-mechanisms. A security sub-mechanism represents a specific way of achieving a security mechanism. For instance, authentication denotes a security mechanism for the fulfilment of a protection objective such as authorisation. However, authentication can be achieved by sub-mechanisms such as passwords, digital signatures and biometrics.

### 3.3 Privacy Requirements Engineering Analysis

PriS (Privacy Safeguard) is a privacy requirements engineering methodology, which provides a set of concepts for modelling privacy requirements in the organisation domain and a systematic way-of-working for translating these requirements into system models.

PriS, initially introduced in [19, 20, 18], is a privacy requirements engineering methodology, developed for assisting designers on eliciting, modelling, designing privacy requirements of the system to be and also providing guidance to the developers on selecting the appropriate implementation techniques that best fit the organisation's privacy requirements. In a recent work [8], privacy process patterns have been integrated to PriS, in order to facilitate system developers to bridge the gap between design and implementation. PriS provides a set of concepts for modelling privacy requirements in the organisation domain and a systematic way-of-working for translating these requirements into system models. This methodology identifies privacy as a multifaceted concept and defines it

in the context of eight technical privacy requirements (such as anonymity and unlinkability) and adopts the use of process patterns as a way to:

- describe the effect of privacy requirements on business processes; and
- facilitate the identification of the system architecture that best supports the privacy-related business processes.

PriS was designed for supporting the realisation of privacy-aware information systems on traditional environments and not for the cloud. Cloud environments introduced a number of new privacy related concepts that along with the ones already stated form a new set of concepts that need to be considered when designing privacy-aware services over the cloud. Thus, extended versions of PriS were introduced [16, 17] for assisting designers to reason about privacy concerns in cloud environments as well.

**PriS Conceptual Model**

The conceptual model of PriS uses the concept of *goal* as the central and most important concept. Goals are desired state of affairs that need to be attained. Goals concern stakeholders, i.e. anyone that has as interest in the system design and usage. Also, goals are generated because of *issues*. An issue is a statement of a *strength*, *weakness*, *opportunity* or *threat* that leads to the formation of the goal. *Cloud Service Providers* (CSPs) constraint the functionality of the developed system or service due to the technologies they use, the policies they follow, the contractual requirements with third parties, etc. Thus, the CSP may provide requirements that designers need to take under consideration during the realisation of the system. Protection of users' privacy is stated in many European and national *legislations* through the form of laws, policies, directives, best practices, etc. All these sources need to be taken under consideration during the identification of functional and non-functional requirements for traditional and cloud-based systems. Thus, goal identification needs to take under consideration all these elements before further analysis is conducted.

PriS distinguishes two types of goals, namely *organisational goals* and *privacy goals*. Organisational goals express the main organisation objectives that need to be satisfied by the system into consideration. Organisational goals will lead to the realisation of system's functional requirements. In parallel, privacy goals are introduced because of specific cloud based privacy related concepts namely *anonymity*, *pseudonymity*, *unlinkability*, *undetectability* and *data protection*. *Unobservability* is realised if the system sufficiently realises undetectability among the respective assets and anonymity of the user accessing them. Thus it is not accomplished directly but indirectly through the realisation of the respective two concepts. Finally, the concepts of *isolation*, *provenanceability*, *traceability*, *interveanability* and *accountability* are related to data protection of user's or systems data over the cloud, as it was explained previously. Thus, all these concepts are grouped under the data protection class. Privacy goals may have an impact on organisational goals. In general, a privacy goal may cause the improvement/adaptation of organisational goals or the introduction of new ones. In this way, privacy issues are incorporated into the system's design.

Goals are realised by *processes*. The transition process from goals to processes includes the causal transformation of general goals into one or more subgoals that form the means for achieving desired ends. During this process, in every step, new goals are introduced and linked to the original one through causal relations, thus forming a hierarchy of goals. Every subgoal may contribute to the achievement to more than one goals.

As it was mentioned previously, goals are realised by processes. PriS uses a set of *privacy process patterns* [8] as a more robust way of bringing the gap between the design and the implementation phase. Privacy process patterns are usually generalised process models, which include activities and flows connecting them, presenting how a business should be run in a specific domain. Privacy process patterns are applied on privacy related processes in order to specify the way that the respective privacy issues will be realised through a specific number of steps. This assists also the developer who can understand in a better and specific way, how to implement the aforementioned privacy concepts. Privacy process patterns are also used for identifying a number of *Privacy Enhancing Technologies* (PETs) already available for implementing the system's privacy requirements. In this way, the developer can choose the most appropriate technology based on the privacy process patterns applied on every privacy-related process.

## 4 Concept Alignment

For proposing a generic approach that combines risk analysis with security and privacy requirements elicitation and modelling approaches, it is important to examine if a correlation between the aforementioned methodologies can occur from a conceptual point of view. The goal is to design a methodology that facilitates analysts and software engineers to get from the system description and threats knowledge a detailed, clearly justified, and well-structured set of security and privacy requirements, covering these threats. EBIOS is an adequate and industrially validated tool to start the study since it assists analysts by guiding them in the early steps of the system design, especially for defining system's security objectives. Secure Tropos, a well-known security requirements engineering methodology can use the EBIOS output as input for deriving "formally" the adequate security requirements for the various elements of the system. Finally, PriS provides an extra focus on privacy, which is a very important topic in the field of ITS security, aiming to increase users' trust, by providing privacy-aware services.

Thus, in order to provide a more efficient design of the unified methodology, an alignment of the EBIOS concepts with the concepts of Secure Tropos and PriS was important in order to identify any conceptual conflicts or any similarities in the terms used. The alignment of the concepts is presented in Table 1. Since Secure Tropos and PriS have their origins from the Software Engineering world, there was no need to align their concepts as well. The necessary alignment was between EBIOS and the two other methods.

## 5 Conclusions

This work comprises the first step towards the development of a methodology for a security assurance framework, able to support connected vehicular technology, by addressing safety, security and privacy of the handled data. The first step of this work, presented in this paper, focuses on the identification of the common concepts of three already existing methodologies, namely EBIOS, Secure Tropos, and PriS. In order to provide a more efficient design of the unified methodology, an alignment of the EBIOS concepts with the concepts of Secure Tropos and PriS was important in order to identify any conceptual conflicts or any similarities in the terms used. This output will be the basis for the development of the methodology that facilitates the transition from a system description and threats knowledge, to a detailed, clearly justified and well-structured set of security requirements.

Assurance security evaluation methods always rely on the definition of a proper security target. Thus, it is an important aspect of the evaluation process to define a meaningful security target. It is often one of the most criticised parts of an evaluation, since there is no universal way to assess the relevance of such a document. But one thing that helps gain confidence in this part of the evaluation is the existence of elements of proof that the system and the real threats associated to it, are properly understood and justified. With this work, we aim to overcome the aforementioned limitations, by providing a methodology which will be able to facilitate the design process of the relevant security target, representing real-world security objectives for Intelligent Transportation Systems (ITS).

In the next steps of this work, the aim is to develop a new tool that will be able to make assurance of security, safety and privacy aspects for Connected Vehicles, measurable, visible and controllable by stakeholders, and thus, enhancing confidence and trust in Connected Vehicles.

| **Concept** | **Meaning** | **Example** | **Concept Alignment with Secure Tropos and PriS** |
|---|---|---|---|
| Entities | Main organisation elements | Hardware, Software, Network, etc. | Resources (Assets), Actors |
| Essential Elements | Functions and information providing added value to the entities. They are linked to the Entities | A computational parameter is an essential element that is linked with the computer A and Software Process B | – |

| | | | |
|---|---|---|---|
| Sensitivity | Security criteria that constraint an essential element. Avoiding the coverage of a security criterion there will be an impact on the organisation through the linked entity. | Integrity, Availability, Confidentiality | Security Constraint, Privacy Constraint |
| Threat Agents | Natural, human, environmental threats, either accidental or deliberate | Earthquake, loss of password | Threat |
| Attack Methods | The knowledge derived by the combination of the sensitivity of the organization and the respective threat agents | Availability and denial of service attack | Attack method |
| Vulnerability | Each entity has a number of vulnerabilities that can be exploited by threat agents using attack methods | A denial of service attack (attack method) exploited by a malicious actor (threat agent) on the web server (entity) due to lack of cryptographic protocol usage (vulnerability) | Vulnerability |
| Security Objectives | The way that vulnerabilities are reduced thus reducing the potential risk on the entities | Protect the integrity of users' data in order to avoid unauthorized alterations from malicious parties. | Security Objectives, Privacy Objectives |
| Security Requirements | The transformation of security objectives into security functionalities that are translated into functional requirements | – | Security Process patterns and plans, Privacy Process patterns and plans |

| | | | |
|---|---|---|---|
| Assurance Requirements | Specific requirements that will guarantee the required level of confidence for the realization of the security requirements expressed as functional requirements | – | Security mechanisms |

Table 1: EBIOS Concepts and Alignment with Secure Tropos and PriS

## Acknowledgment

This work is a part of the SAFERtec project. SAFERtec has received funding from the European Union's Horizon 2020 research & innovation programme under grant agreement no 732319. Content reflects only the authors' view and European Commission is not responsible for any use that may be made of the information it contains.

## References

1. Ebios — expression des besoins et identification des objectifs de sécurité (2019), https://www.ssi.gouv.fr/guide/ebios-2010-expression-des-besoins-et-identification-des-objectifs-de-securite/
2. Alberts, C., Dorofee, A., Stevens, J., Woody, C.: Introduction to the octave approach. Tech. rep., CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST (2003)
3. Biagio, A.R.M.G.M.D.J.F.M.E.G.M.C.K.A.K.J.L.L.M.M.A.S.B.T.C.C.: An analysis of possible socio-economic effects of a cooperative, connected and automated mobility (ccam) in europe. Eur - scientific and technical research reports, Publications Office of the European Union (2018)
4. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J.: Tropos: An agent-oriented software development methodology. Autonomous Agents and Multi-Agent Systems **8**(3), 203–236 (2004)
5. Chung, L., Nixon, B.A.: Dealing with non-functional requirements: three experimental studies of a process-oriented approach. In: Proceedings of the 17th international conference on Software engineering. pp. 25–37. ACM (1995)
6. Colombo, P., Ferrari, E.: Towards a modeling and analysis framework for privacy-aware systems. In: Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom). pp. 81–90. IEEE (2012)
7. Deng, M., Wuyts, K., Scandariato, R., Preneel, B., Joosen, W.: A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. Requirements Engineering **16**(1), 3–32 (2011)

8. Diamantopoulou, V., Kalloniatis, C., Gritzalis, S., Mouratidis, H.: Supporting privacy by design using privacy process patterns. In: IFIP International Conference on ICT Systems Security and Privacy Protection. pp. 491–505. Springer (2017)
9. Diamantopoulou, V., Pavlidis, M., Mouratidis, H.: Evaluation of a security and privacy requirements methodology using the physics of notation. In: Computer Security, pp. 210–225. Springer (2017)
10. European Commission: Communication from the commission to the european parliament, the council, the european economic and social committee, the committee of the regions, on the road to automated mobility: An eu strategy for mobility of the future (2018)
11. Faßbender, S., Heisel, M., Meis, R.: Functional requirements under security pressure. In: Software Paradigm Trends (ICSOFT-PT), 2014 9th International Conference on. pp. 5–16. IEEE (2014)
12. Faßbender, S., Heisel, M., Meis, R.: Problem-based security requirements elicitation and refinement with pressure. In: International Conference on Software Technologies. pp. 311–330. Springer (2014)
13. Fredriksen, R., Kristiansen, M., Gran, B.A., Stølen, K., Opperud, T.A., Dimitrakos, T.: The coras framework for a model-based risk management process. In: International Conference on Computer Safety, Reliability, and Security. pp. 94–105. Springer (2002)
14. Haley, C., Laney, R., Moffett, J., Nuseibeh, B.: Security requirements engineering: A framework for representation and analysis. IEEE Transactions on Software Engineering **34**(1), 133–153 (2008)
15. Islam, S., Ouedraogo, M., Kalloniatis, C., Mouratidis, H., Gritzalis, S.: Assurance of security and privacy requirements for cloud deployment models. IEEE Transactions on Cloud Computing **6**(2), 387–400 (2015)
16. Kalloniatis, C.: Designing privacy-aware systems in the cloud. In: International Conference on Trust and Privacy in Digital Business. pp. 113–123. Springer (2015)
17. Kalloniatis, C.: Incorporating privacy in the design of cloud-based systems: a conceptual meta-model. Information & Computer Security **25**(5), 614–633 (2017)
18. Kalloniatis, C., Kavakli, E., Gritzalis, S.: Pris methodology: incorporating privacy requirements into the system design process. In: Proceedings of the SREIS 2005 13th IEEE International Requirements Engineering Conference–Symposium on Requirements Engineering for Information Security, J. Mylopoulos, G. Spafford (Eds.) (2005)
19. Kalloniatis, C., Kavakli, E., Gritzalis, S.: Addressing privacy requirements in system design: the pris method. Requirements Engineering **13**(3), 241–255 (2008)
20. Kalloniatis, C., Kavakli, E., Kontellis, E.: Pris tool: A case tool for privacy-oriented requirements engineering. In: MCIS. p. 71 (2009)
21. Kleberger, P., Olovsson, T., Jonsson, E.: Security aspects of the in-vehicle network in the connected car. In: Intelligent Vehicles Symposium (IV), 2011 IEEE. pp. 528–533. IEEE (2011)
22. Labuschagne, W.B.L., et al.: A comparative framew ork for evaluating information security risk management methods. Standard Bank Academy for Information Technology, Rand Afrikaans University (2004)
23. McKeefry, H.L.: Consumers get on board with connected cars (2016)
24. Mead, N.R., Stehney, T.: Security quality requirements engineering (SQUARE) methodology, vol. 30. ACM (2005)
25. Mouratidis, H.: A security oriented approach in the development of multiagent systems: applied to the management of the health and social care needs of older people in England. Ph.D. thesis, University of Sheffield (2004)

26. Mouratidis, H.: Secure software systems engineering: the secure tropos approach. JSW **6**(3), 331–339 (2011)
27. Mouratidis, H., Giorgini, P.: Secure tropos: a security-oriented extension of the tropos methodology. International Journal of Software Engineering and Knowledge Engineering **17**(02), 285–309 (2007)
28. Mouratidis, H., Islam, S., Kalloniatis, C., Gritzalis, S.: A framework to support selection of cloud providers based on security and privacy requirements. Journal of Systems and Software **86**(9), 2276–2293 (2013)
29. général de la défense nationale Direction centrale de la sécurité des systèmes d' ?nformation, P.M.S.: The ebios method, expression of needs and identification of security objectives
30. Parliament, E.: Self-driving cars in the eu: from science fiction to reality (2019)
31. Petit, J., Shladover, S.E.: Potential cyberattacks on automated vehicles. IEEE Trans. Intelligent Transportation Systems **16**(2), 546–556 (2015)
32. Sabouri, A., Rannenberg, K.: Abc4trust: protecting privacy in identity management by bringing privacy-abcs into real-life. In: IFIP International Summer School on Privacy and Identity Management. pp. 3–16. Springer (2014)
33. Salini, P., Kanmani, S.: Application of model oriented security requirements engineering framework for secure e-voting. In: Software Engineering (CONSEG), 2012 CSI Sixth International Conference on. pp. 1–6. IEEE (2012)
34. Spiekermann, S., Cranor, L.F.: Engineering privacy. IEEE Transactions on software engineering **35**(1), 67–82 (2009)
35. Stolen, K., den Braber, F., Dimitrakos, T., Fredriksen, R., Gran, B.A., Houmb, S.H., Lund, M.S., Stamatiou, Y., Aagedal, J.: Model-based risk assessment-the coras approach. In: iTrust Workshop (2002)
36. Van Lamsweerde, A.: Goal-oriented requirements engineering: A guided tour. In: Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on. pp. 249–262. IEEE (2001)
37. Yazar, Z.: A qualitative risk analysis and management tool–cramm. SANS InfoSec Reading Room White Paper **11**, 12–32 (2002)
38. Yu, E.: Modelling strategic relationships for process reengineering. Social Modeling for Requirements Engineering **11**, 2011 (2011)
39. Zhang, T., Antunes, H., Aggarwal, S.: Defending connected vehicles against malware: Challenges and a solution framework. IEEE Internet of Things journal **1**(1), 10–21 (2014)