ThreMA: Ontology-based Automated Threat Modelling for ICT Infrastructures

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 September 2022, accepted 25 October 2022, date of publication 3 November 2022, date of current version 9 November 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3219063*

APPLIED RESEARCH

# ThreMA: Ontology-Based Automated Threat Modeling for ICT Infrastructures

**FABIO DE ROSA[1], NICOLÒ MAUNERO[1,2], PAOLO PRINETTO[1,2,3], (Senior Member, IEEE), FEDERICO TALENTINO[1], AND MARTINA TRUSSONI[1]**

[1]National Interuniversity Consortium for Informatics, 00185 Rome, Italy
[2]Department of Control and Computer Engineering, Polytechnic of Turin, 10129 Turin, Italy
[3]System Security Modelling and Analysis Group, IMT School for Advanced Studies, 55100 Lucca, Italy

Corresponding author: Nicolò Maunero (nicolo.maunero@polito.it)

**ABSTRACT** *Threat Modelling* allows defenders to identify threats to which the target system is exposed. Such a process requires a detailed infrastructure analysis to map threats to assets and to identify possible flaws. Unfortunately, the process is still mostly done manually and without the support of formally sound approaches. Moreover, *Threat Modelling* often involves teams with different levels of security knowledge, leading to different possible interpretation in the system under analysis representation. Threat modelling automation comes with two main challenges: (i) the need for a standard representation of models and data used in various stages of the process, establishing a formal vocabulary for all involved parties, and (ii) the requirement for a well-defined inference rule set enabling reasoning process automation for threat identification. The paper presents the *ThreMA* approach to automating threat modelling for ICT infrastructures, aiming at addressing the key automation issues through the use of ontologies. Specifically, a formal vocabulary for modelling an ICT infrastructure, a threat catalog and a set of inference rules needed to support the reasoning process for threat identification are provided. The proposed approach has been validated against actual significant case studies provided by different Stakeholders of the Italian Public Sector.

**INDEX TERMS** Automation, cybersecurity, ontology, threat modeling.

## I. INTRODUCTION

Cyber Risk Management, similarly to its business counterpart, is the process of forecasting and evaluating the risk related to cyber attacks and incidents, together with the procedures to mitigate or avoid their impact [1]. In the various proposed Risk Management frameworks [2], [3], [4], [5], one of the key parts of the whole process is the *Risk Assessment* [6]. It considers different phases of the system life cycle, from design to actual implementation, intending to assess the system cybersecurity posture and preparedness level against possible cyber attacks. The *Risk Assessment* process can be split into three main phases: (i) *Threat Modelling*, aimed at identifying the threats the system is exposed to, e.g., the possible attack vectors and how attackers could exploit any vulnerability or flaw in the system; (ii) *Vulnerability Assessment* (VA), which is the iterative process of identifying vulnerabilities, and (iii) *Penetration Testing* (PT), in which identified vulnerabilities and potential issues are thoroughly analysed by trying to exploit them. These three phases synergistically compute the level of cyber risk to which the system is exposed.

The *Threat Modelling* phase is fundamental and, if done effectively, provides guidelines for the later two phases, highlighting critical points while avoiding missing important parts during vulnerability assessment and penetration testing (VAPT).

Since Threat modelling is still today a mainly manual activity [7], it requires highly trained personnel, and it is often error-prone [8]. In addition, these problems are compounded by the rapid and constant growth of ICT infrastructure complexity [9] and related cyber threats [10], [11]. Therefore, it becomes crucial to create solutions aiming to automate the Threat Modelling process, in particular, to develop tools

The associate editor coordinating the review of this manuscript and approving it for publication was Jiafeng Xie.

able to prevent human errors and simplify the management of organisations' ICT infrastructures [12]. Moreover, automation enables a faster reaction to changes both in the infrastructure (e.g., technology, architecture) and in threats (e.g., new attack vectors), allowing the security operations team to plan remediation activities and improve the system security posture in a shorter time.

Over the years, several solutions trying to automate the threat modelling process have been proposed [13] in both academia and industry. Most of them are focused on specific vertical sectors, such as the world of IoT [14] or Industrial Control Systems (ICS) [15], which makes them harder to be adopted to more general ICT infrastructure scenarios. On the other hand, ''generic'' approaches and solutions available in the literature often either consider threats in a too broader way or are too tied to known system vulnerabilities, thus risking resulting in ineffective system analysis [16]. Moreover, supply chain threats are often marginally considered [17].

Generally speaking, automating the threat modelling process brings with it two main challenges: the former one being the need for a standard representation of models and data used in various stages of the process, establishing a formal and common vocabulary for all the involved parties. Often, threat modelling is performed by different specialists, with different levels of security knowledge; hence, different interpretations of models and data may take place, thus generating misunderstanding and ineffectiveness during the overall threat modelling process [18]. The latter challenge is the definition of a well-defined set of inference rules supporting the reasoning task during the threat identification process. This set of rules has to be based on formal models and data representing the target ICT infrastructure. Frequently, both these data and their threats lack the right context, thus leading to low accuracy in terms of domain knowledge compared with what a domain expert could do [19]. Such a problem can effectively be addressed by exploiting ontologies. These are often designed to support computation with structured data by providing a standardised descriptive vocabulary for concepts and machine-processable entity-relation semantics, thus, providing the possibility of performing automated reasoning to extract the desired information [19].

The work presented in this paper aims at addressing the key issues related to automating threat modelling through the use of ontologies, by providing a standard metamodel, named *ThreMA*, to describe generic ICT infrastructure and a well-defined set of rules able to support engines during the inference task. *ThreMA metamodel* is an ontology conceptually composed of two main parts: (i) the formal vocabulary to model ICT infrastructures (e.g., components, data flow, and so on), and (ii) threat descriptions and categorisations. These two parts are each other related via inference rules, the threat modelling logic, used by reasoners during the threat identification process.

The remainder of the paper is structured as follows: Section II provides some background concepts as well as a brief overview of related works. Section III describes the *ThreMA* approach, providing an overview of both the ontology implementation and the inference rules adopted to automate the threat modelling process. Section IV provides a validation example of the proposed solution, and, finally, Section V summarises the work and presents some possible future improvements.

## II. BACKGROUND AND RELATED WORKS

Threat modelling can be defined in different ways depending on the target application domain [13], but a general definition that captures the nuances of the overall process can be: ''a systematic way to identify threats that might compromise security'' [20]. Moreover, threat modelling has the following key advantages [21]: (i) when applied during the different stages of the system life cycle, from design to implementation, it allows ranking threats, prioritising the most important ones, and assuring resources be distributed effectively to develop and maintain adequate defences; (ii) iteratively applying threat modelling can assure proper mitigations be in place for newly discovered threats.

Several approaches and tools have been proposed in the literature to address automation issues in threat modelling. Next, an overview is presented.

### A. THREAT MODELLING METHODOLOGIES

A complete overview of methodologies for threat modelling can be found in [22], whereas an overview of threat modelling approaches and techniques is given in [13].

One of the fundamental steps, shared among most of the proposed threat modelling methodologies, is the definition of a *Data Flow Diagram* (DFD) [23]. As the name suggests, it describes how data are exchanged among the various components of the system under analysis, being it a software program or a whole ICT infrastructure, indifferently. Analysts take advantage of the data flow to understand which possible attacks threaten the system, how and which assets they impact, and so on. From that, a specific threat methodology is applied.

The most widely adopted threat modelling methodologies, both in academia and industry, are *STRIDE* and *Attack Trees*. Both methodologies do not make use of any automation mechanism but provide guidelines that security analysts can adopt to identify threats in a system. They are, most often, the basis on which threat modelling tools are built upon.

#### 1) STRIDE

STRIDE is a threat modelling methodology developed by Microsoft [24]. It is used along with the model of the target system and provides guidelines for system analysis and threat discovery. STRIDE divides threats into 6 main categories: (i) **Spoofing**: illegally accessing and then using another user's authentication information; (ii) **Tampering**: malicious modification of data; (iii) **Repudiation**: situations in which a malicious actor denies performing a certain operation without other involved parties having the possibility of

disproving it; (iv) **Information Disclosure**: revealing information to entities who are not supposed to have access to them; (v) **Denial of Service (DoS)**: breaking the availability of a certain service, making it not accessible to legitimate users; (vi) **Elevation of Privileges**: scenarios in which a malicious actor, without the proper level of privileges, obtains privileged access to a system, data, or assets in general. STRIDE is applied by analysing how each of the 6 threat categories affects all the system components, their interconnections, and relationships.

STRIDE comes with a related Threat Modelling Tool (TMT) [25] to support analysts in assessing the security level of the systems by identifying mitigation to be implemented. More specifically, the tool tries to find software design flaws starting from a DFD diagram and technologies adopted to implement the system; then, it extracts possible threats with their proposed mitigation, following the STRIDE categorisation.

#### 2) ATTACK TREES

One of the oldest and most popular threat modelling techniques is the usage of Attack Trees [26]. Initially used alone, it is now frequently combined with other approaches like STRIDE or PASTA [27].

The goal of Attack Trees is to identify all the possible attack goals within the system under analysis. For each of them, a graph is created that models how an attacker can achieve that goal, by describing the various steps of the attack and how it unravels within the system. This information can then be used to take security decisions, such as proposing the most effective remediation to be put in place to mitigate identified attacks. Attack Trees are very simple and easy to adopt, but require the involvement of analysts with high cybersecurity expertise and capable of clearly understanding the system composition and its security concerns.

### B. AUTOMATING THREAT MODELLING

An interesting approach to automating threat modelling is the one provided by the *AutoSEC* tool, developed by Frydman et al. [12]. It takes advantage of DFD, *identification trees*, and *mitigation trees* models to automate the threat modelling process. In particular, the DFD model is generated using the Microsoft TMT and is given to *AutoSEC* as input. *Identification trees* and *mitigation trees* are built from databases such as CAPEC[1] and CWE[2] and contain the information needed to identify threats within the DFD model as well as the related countermeasures. Starting from these models, the tool automatically derives the threat model by mapping the threats to the system under analysis.

Works in [28] and [29] extend Microsoft TMT to specific domains. In the former, authors present a system for automatic threat modelling of edge computing scenarios, by providing a specific library based on three main asset categories: *Physical/virtual processing nodes*, *Software components/modules*, and *Communication channels*. Data are divided into *user-related data*, *environmental data*, and *service data*, whereas IoT threats are modelled through a well-defined library.

In [29], the Microsoft TMT is extended to Smart-Grid threat modelling. A model, composed of nodes describing a Smart Grid context and a list of threats to be considered, is created to be imported into the tool. The threat categorisation is based on STRIDE, and, for each category, threats are identified for a given context.

In general, all of the above works are focused on specific vertical domains, such as software design or IoT/edge computing; which makes it difficult to adapt to generic ICT infrastructures. Moreover, in [12] authors highlight the limitation of using the Microsoft TMT, identifying them as having a low degree of expressiveness both in the system modelling and threat modelling logic.

### C. ONTOLOGY-BASED THREAT MODELLING APPROACHES

One of the most recent and promising works concerning ontology-driven threat modelling is the *Ontology Driven Threat Modelling* Framework, defined by the OWASP foundation.[3] The framework tries to automate the threat modelling process using ontology to collect all the information concerning the security of a specific domain. The project is still in progress, but the idea is to build a base threat model ontology (the *Base Threat Model*) and, from it, to provide more domain-specific ontologies and an ontology-driven threat rule engine. The *Base Threat Model* is an OWL ontology [30] that contains classes and properties representing the components of an ICT infrastructure, the threats, the mitigations, and the properties that bind all of these classes. Moreover, it is possible to model also the *DFD Diagrams*. Starting from the *Base Threat Model*, it is possible to build more *Domain Specific Models* that contain typical components, threats and countermeasures of a specific area [30], [31]. The project focuses primarily on cloud-based infrastructures [32].

Also, the approach proposed by Manzoor et al. [16] is designed for automating threat modelling of cloud infrastructures. The work is based on an ontology modelling the different components needed for security analysis and threat identification. The main drawback of the approach is that threats are identified as known vulnerabilities, thus going to limit the effectiveness of the threat modelling.

The same limitation of having an ontology only for cloud-based systems can be found in *Nemesis* [33] and the related Vulcan tool [34].

The work proposed by Välja et al. [19] defines an ontology framework to improve automation in threat modelling, by addressing two fundamental issues: the lack of domain knowledge and the mismatched granularity. However, the framework focuses only on software-related concerns and threats are identified only by known system vulnerabilities.

[1] https://capec.mitre.org
[2] https://cwe.mitre.org
[3] https://owasp.org

Salini and Shenbagam [35], instead, propose an ontology modelling web attacks, by combining threats and vulnerability information. The approach allows for deriving possible attacks through rules and related inference engines. The proposed approach has not been designed for threat modelling per se, but rather to allow an analyst, given a vulnerability (e.g., cross site scripting) to understand what the possible attacks are (e.g., XSS Attacks). Nevertheless, only threats related to web applications are taken into consideration.

Ekelhart et al. [18] simulate the impact that an informed incident (a threat that materialises) can have on corporate assets through a threat ontology. The proposed ontology allows the description of both cyber and physical threats, such as fires and natural disasters, by following a taxonomy defined by Landweher [36]. The ontology is very general, but it describes only the concepts of the CIA triad, without going into details about specific threats. The asset description is not very detailed and allows a very limited representation of an ICT infrastructure.

Also, the work of Luh et al. (TAON) [37] uses ontology to model cyber attacks, in particular APT-style attacks. TAON does not perform threat modelling and is aimed at supporting security analysts in identifying the proper mitigation(s) for the given attack(s). The ontology must be populated manually with known attacks information, the list of assets, and the catalog of commercial of-the-shelf (COTS) security solutions of interest. Once populated, the ontology is used to correlate information allowing security analysts to understand, given an attack, which are the assets at risk and which COTS security solutions can be used to mitigate that attack.

Finally, Sabbagh and Kowalski [38] provide a socio-technical framework to support the analysis of possible threats that may be generated in the supply chain. The proposed approach does not involve any kind of automation but is intended to provide a tool that analysts can use to reason, holistically, about the supply chain security issues and how to address them effectively.

Summarising, the works proposed in the literature mainly focus on specific vertical domains, making them hardly reusable in different contexts. In addition, they lack standard representations of models and data used in the various stages of the process. In addition, the inference rules supporting the reasoning task during the threat identification process are generally targeted to specific vulnerabilities and attacks.

Furthermore, the proposed approaches only automate part of the threat modelling process, still requiring human interaction in the definition of threats and resorting to external tools, in a not integrated way. Finally, aspects related to supply-chain threats are rarely considered.

The approach proposed in the present paper provides a standard metamodel to describe a complete ICT infrastructure, establishing a formal and common vocabulary for all the involved parties; threats are already categorised and formalised in the designed ontology, starting from common threat databases. In addition, a well-defined set of inference rules are provided to describe the threat modelling logic and to map infrastructure components to threats using ontology automatic reasoners. Finally, supply chain related security concerns are taken into consideration. The proposed approach can be easily extended, and no external tools or coding are required.

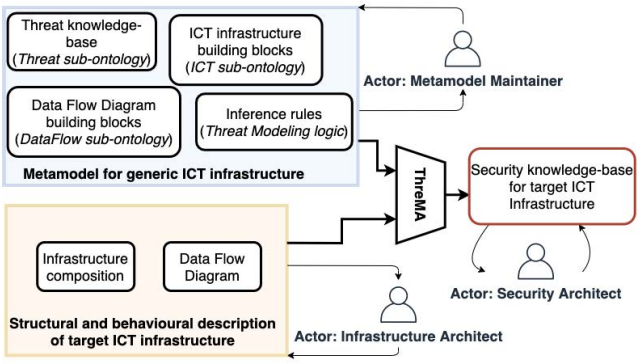


**FIGURE 1.** ThreMA architecture.

## III. THE ThreMA APPROACH

This section presents the ThreMA approach and its architecture, detailing the elements composing the underlying ontology *metamodel*. The metamodel is conceptually divided into three subontologies, two designed for modelling the target system, while the other provides the threat catalog used for threat modelling. Specifically, the `ICT sub-ontology` contains rules and vocabulary for modelling an ICT infrastructure; the `Data Flow sub-ontology` is intended to represent the data flow diagram and the `Threat sub-ontology` contains the characterisation of threats. These three parts are connected by means of relationships and the `Threat Modelling logic` is expressed using `inference rules` used by reasoners to map threats to infrastructure components.

The ThreMA ontology has been developed by resorting to *Protégé*,[4] a tool provided by Stanford University that allows for building ontology-based applications. It embeds the necessary features for describing an ontology, populating it, and applying automatic reasoners. The adopted ontology description language is OWL2 [39], while Semantic Web Rule Language (SWRL) [40] is used for specifying the inference rules.

ThreMA has been designed for modelling complete ICT infrastructures through a formal dictionary shared by all the experts involved in the threat modelling process. In addition, according to [41], a well-defined rule set has been defined for supporting system modelling, in order to prevent misconceptions that could lead to sub-optimal or erroneous threat identification. As it will be seen in the following sections, there is a tight link between how the threat catalog is organised and how the infrastructure components are described. Both are organised as a function of each other; threats are

[4]https://protege.stanford.edu

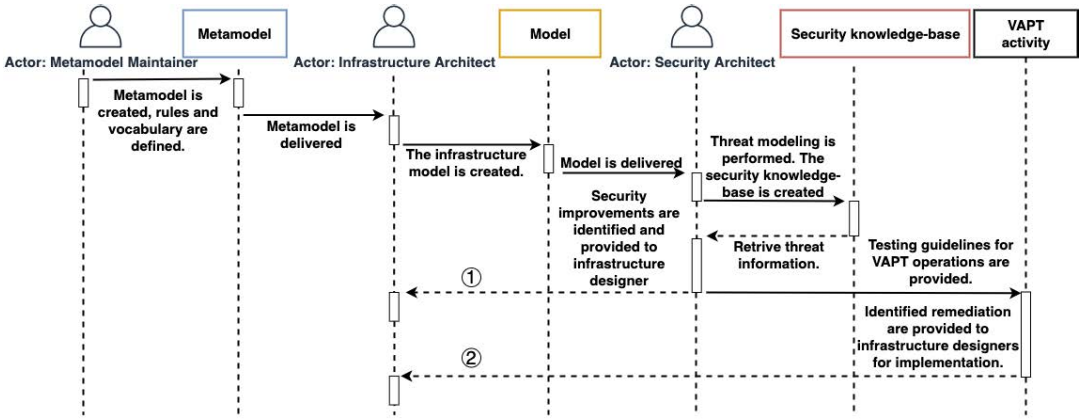


**FIGURE 2.** Sequence diagram and use cases.

categorised according to what types of infrastructure components or interactions they threaten; conversely, the various components are also organised according to what threatens them. This organisation allows a more efficient application of the automatic reasoning process, simplifying the creation of threat modelling logic rules.

In the following, first, the ThreMA architecture is presented to give a broader look at its components and functionalities, then the various parts composing ThreMA are analysed more in details.

### A. ThreMA ARCHITECTURE

Fig. 1 depicts the ThreMA architecture, showing main components and actors involved. Two main input are required: the *ICT infrastructure metamodel* and the *structural and behavioural descriptions* of the target ICT infrastructure. The ICT infrastructure metamodel (shortly, metamodel) is the ontology containing the set of formal modelling rules and vocabulary that allow for modelling ICT infrastructures and its threats. The threat knowledge-base provides the catalogue of threats organised by the type of infrastructure components they apply to, whereas rules, in the SWRL language, allow automatic reasoners to map the described infrastructure components to the affecting threats. The metamodel is maintained by the *Metamodel Maintainer*, e.g., the person or team in charge of developing and extending it.

The second input, on the other hand, represents the structural and behavioural descriptions, in terms of components and data flow, of the infrastructure under analysis. In this case, the *Infrastructure Architect*, e.g., the person responsible for the infrastructure under analysis, is in charge of designing, exploiting the rules described in the metamodel, the target infrastructure, by creating a model containing assets, connections, and interactions among the various components.

On the basis of the provided inputs, ThreMA, by using the internal ontology reasoner, is able to automatically extract the threat model by applying the rules defined in the metamodel to the described infrastructure. The output of the process is a *security knowledge-base* that contains all the information needed by a *Security Architect* to identify critical points in the infrastructure, to plan the implementation and adoption of mitigation and remediation actions, and so on.

Fig. 2 provides a sequence diagram showing the temporal sequence of steps performed by using ThreMA tool. Everything starts with the creation of the metamodel (rules and vocabulary to model an ICT infrastructure) and of the threat knowledge-base. The metamodel is delivered to the *Infrastructure Architect* who designs the infrastructure under analysis. The created *model* is then used to identify potential threats of the ICT infrastructure under analysis, through the ontology reasoner, by generating the relative *security knowledge-base*. The *Security Architect* can utilise the produced security knowledge-base either in the design phase or after the implementation.

Two use cases are shown in Fig. 2 as well. ① Represents threat modelling applied during the design phase; the *Security Architect* can use the *security knowledge-base* to identify security improvements and provide them to the *Infrastructure Architect* to modify the ICT infrastructure design. ② Represents threat modelling applied to an ICT infrastructure already in operation. The *Security Architect* can utilise the gathered threat information to provide testing guidelines for VAPT (Vulnerability Assessment and Penetration Testing) activities or to identify vulnerable points in the ICT infrastructure and relative remediation plan to be provided to the *Infrastructure Architect* for implementation.

### B. ICT SUB-ONTOLOGY

Fig. 3 depicts the metamodel used for modelling ICT infrastructures. Particular attention has been placed on representation and organisation of assets. As defined in [41], an asset can be either something valuable for the organisation or everything that may contain vulnerabilities that can be menaced by threats. Hence, not only the stored data, the networks and the digital services must be modelled but, more in general, every device and software connected to the ICT infrastructure that could be the target and/or the vector of possible attacks. Infrastructure components are organised according to

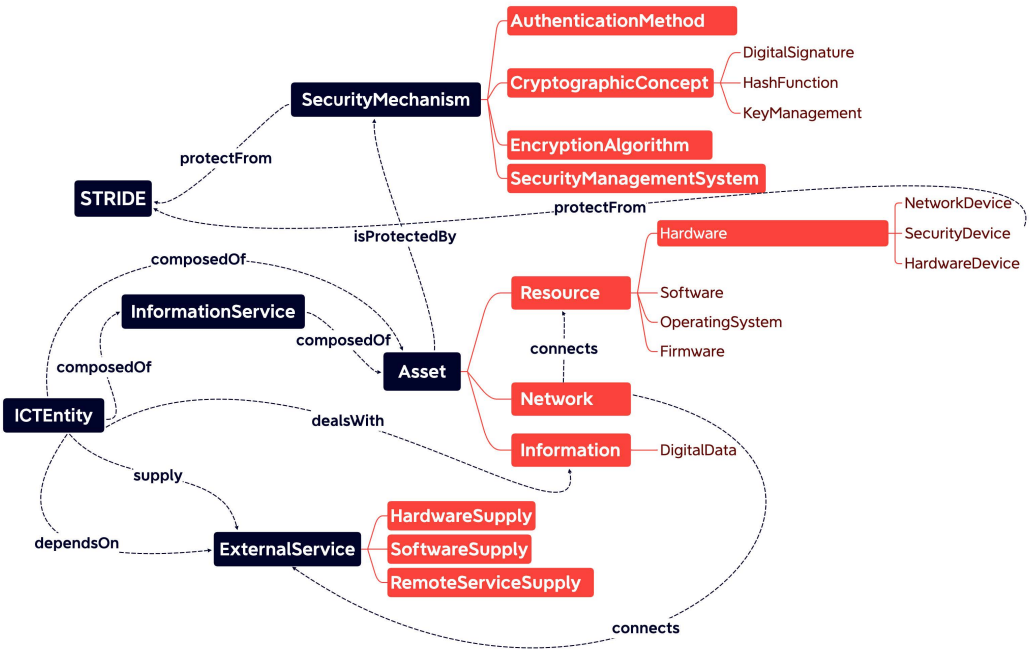


**FIGURE 3.** ICT sub-ontology.

the type of threats they can be targeted by. The sub-ontology is, thus, organised as detailed in the sequel of this section. For each Entity, we provide a brief description, its sub-entities and relationships as well as why they have been introduced and what they represent.

The main entity, `ICTEntity`, represents the infrastructure target of the analysis, which is related to entities `InformationService` and `Asset` by the `composedOf` relationship.

The `InformationService` entity represents a single digital service within the infrastructure. It can include, among the others, a user management service, a computational service, and so on. Although not directly used for the threat modelling process, this entity has been introduced to allow a better structural organisation of assets belonging to different parts of the infrastructure. It is related to the entity `Asset` by the `composedOf` relationship.

The `Asset` entity is used to model the assets of the infrastructure and includes three main sub-entities: (i) `Network` represents a network of the infrastructure and is related to the entity `Resource` by the property `connects`; (ii) `Information` represents the critical information managed by the infrastructure. It describes data at rest, such as databases of user data, and (iii) the entity `Resource` represents all other type of assets.

The `Resource` entity includes four main sub-entities: (i) `Hardware`, (ii) `Software`, (iii) `OperatingSystem` and (iv) `Firmware`. These sub-entities specialise the types of assets the infrastructure may contain and have been introduced to provide both a more formal modelling, avoiding possible misunderstanding, and a tool to better map specific threats to assets in order to prevent the inclusion of threats that are not meaningful within the analysed context.

The entity `Hardware` is further divided into: (i) `NetworkDevice`, (ii) `SecurityDevice` and (iii) `HardwareDevice`. The entity `SecurityDevice` represents security hardware devices, such as firewalls. It is related to the entity `STRIDE` by the relation `protectsFrom`, which represents the STRIDE threat category the device mitigates.

The `SecurityMechanism` entity is used to model security solutions used to protect the specific infrastructure component. It serves two main purposes: (i) describing which security mechanisms are in operation to allow understanding which threats are mitigated by what, but also (ii) identifying possible threats specific to these type of solutions, such as a faulty implementation of an authentication protocol. Hence, the `Asset` entity is related to the class `SecurityMechanism` by the property `isProtectedBy`, while the relationship `protectFrom` with the `STRIDE` entity models which STRIDE category the adopted security mechanism mitigates. `SecurityMechanism` is further specialised with four sub-entities. (i) `EncryptionAlgorithm` represents possible encryption algorithms adopted, while (ii) `CryptographicConcept` represents all those cryptographic concepts that are not encryption algorithms. (iii) `SecurityManagementSystem` represents security solutions such as Endpoint Detection Response (EDR) and monitoring systems, while (iv) `AuthenticationMethod` represents authentication protocols and mechanisms.

The entity `ExternalService` is used to model the supply chain and dependencies from external providers and suppliers. It is related to `ICTEntity` through two relationships `supply` and `dependsOn`. The former represents something the infrastructure provides externally, such as a web service others may utilise, while the latter represents something the infrastructure depends on. `ExternalService` includes three sub-entities: (i) `HardwareSupply`, (ii) `SoftwareSupply` and (iii) `RemoteServiceSupply`. `HardwareSupply` represents supply of hardware components, for example outsourced production of devices. `SoftwareSupply` represents supply of software components, such as outsourced development of software, and `RemoteServiceSupply` represents externally hosted digital services, such as web services or, in general, *something*-as-a-service.

### C. DATA FLOW DIAGRAM SUB-ONTOLOGY

Fig. 4 depicts the *Data Flow* metamodel, the part of the ontology devoted to model the DFD, e.g., how components of the infrastructure communicate each other.

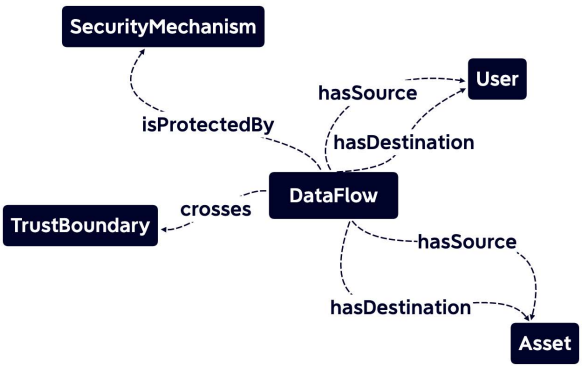


**FIGURE 4.** Data flow sub-ontology.

The entities and relationships added to describe the data flow are quite simple. The main entity is `DataFlow`; it represents a communication channel between specified source and destination and is modelled by using two relationships: `hasSource` and `hasDestination`. Target of these two relationships can be a `User` or an `Asset` entity. Moreover, the entity `DataFlow` can specify which security mechanism is in place for its protection using the relationship `isProtectedBy` with the entity `SecurityMechanism`.

The `TrustBoundary` entity represents a change in the level of privileges between source and destination of a data flow. This can be specified for a `DataFlow` entity using the relationship `crosses`.

Finally, the `User` entity models user interaction with the infrastructure.

### D. THREAT SUB-ONTOLOGY

Fig. 5 depicts the composition of the threat sub-ontology used for modelling threats.

The threats are organised in different classes based on the type of the infrastructure components they apply to and the interactions the components have with the surrounding environment.

The source of information used for modelling threats is the *MITRE CAPEC* knowledge-base, in particular the abstraction level adopted is the *CAPEC* view *Domains of Attack*, which is composed of six categories: (i) **Software**, (ii) **Hardware**, (iii) **Communications**, (iv) **Supply Chain**, (v) **Physical Security**, and (vi) **Social Engineering**. Among those, the first four categories were selected to create the list of threats, while *Social Engineering* threats and *Physical Security* were not considered. The former was excluded because its effective application will require thoughtful modelling of human related aspects, which are out of the scope of this work and will require further analysis. The latter was excluded for a similar reason; this category contains threats to physical security systems, such as locks and physical alarms, which are not taken into account in the proposed ICT infrastructure metamodel.

More specifically, CAPEC threats are organised following different levels of abstraction: each category contains *Meta Attack Patterns* which, in turn, contain *Standard Attack Patterns*. Each Standard Attack Pattern is composed of *Detailed Attack Patterns*. For sake of simplicity, the present paper focuses on the *Meta Attack Patterns* only, but the proposed reasoning and rules can be easily extended to the subclasses, by following the same described approach.

Starting from the *Domains of Attack* view, twelve categories have been identified and the corresponding ontology entities created. CAPEC's entries have been associated with each entity by providing the corresponding ID, the description, and the applicable countermeasures. These are taken from the CAPEC knowledge-base, as well.

- Entity `AuthenticationMechanism` refers to threats related to flaws in authentication systems;
- Entity `ClientServerInteraction` refers to threats related to flaws in client-server communication;
- Entity `CommunicationChannel` refers to threats affecting flaws in communication in general;
- Entity `Crypto` refers to threats affecting flaws in applied cryptography;
- Entity `Hardware` refers to threats affecting hardware devices;
- Entity `SoftwareInput` refers to threats affecting software components' input interfaces;
- Entity `HardwareInput` refers to threats affecting the hardware input/output system;
- Entity `NetworkCommunication` refers to threats affecting network flaws in particular;
- Entity `PrivilegeOrPermissionAbuse` refers to threat affecting flaws in the mechanism managing privilege levels;
- Entity `Software` refers to threats affecting software;
- Entity `SupplyChain` refers to threats affecting the supply chain, being them outsourcing production or supplying customers;

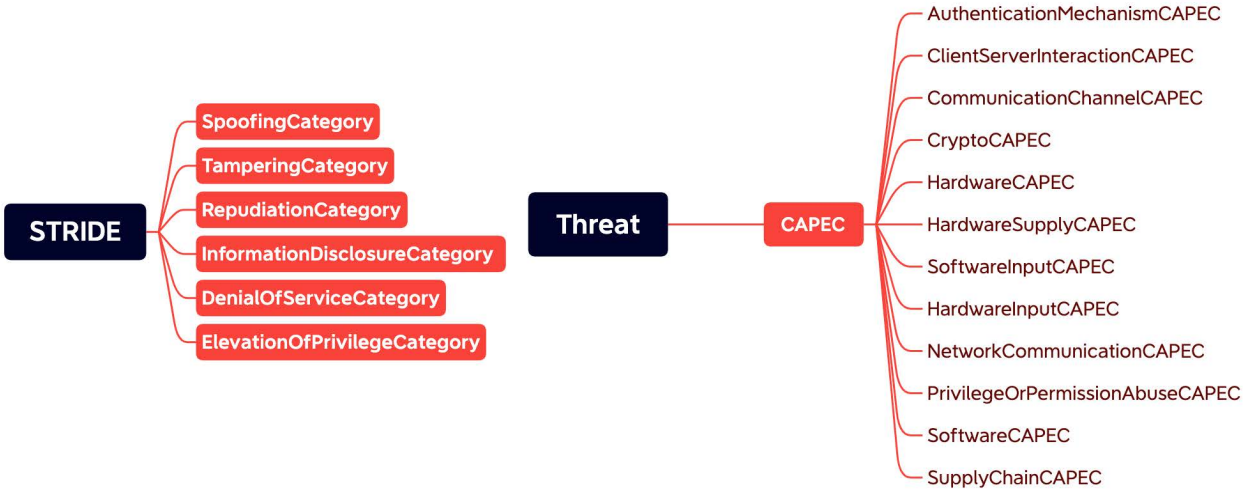


**FIGURE 5.** Threat sub-ontology.

- Entity `HardwareSupply` refers to threats affecting in particular the supply chain of hardware components.

Associating each threat with possible mitigations helps in providing a meaningful knowledge-base as a result of the threat modelling process. As described in Section III-A, the results obtained can be used to better individuate remediation action to be taken to improve the infrastructure security.

The second part of the threat sub-ontology is devoted to model STRIDE. Simply, a `STRIDE` entity was inserted with 6 sub-entities, each representing a STRIDE category. The `STRIDE` entity serves two purposes: (i) threats are associated with the STRIDE category they belong to, and (ii) security mechanisms adopted in the infrastructure are associated with the STRIDE category they mitigate.

Mapping between instances of the `CAPEC` entity and the `STRIDE` sub-entities was performed by following the *CAPEC-STRIDE Mapping* project [42]. The project consists of binding the *Meta Attack Patterns* contained in CAPEC to the corresponding STRIDE category. In the proposed ontology, the mapping is performed using the object property `isLabelledWithSTRIDE`.

Table 1 summarises the identified threat categories and the associated threats, represented using the corresponding CAPEC ID.

**TABLE 1.** Threat categories and associated threats.

| Threat Category | Associated CAPEC Threat IDs |
|---|---|
| AuthenticationMechanism | 114, 115, 122, 21, 560 |
| ClientServerInteraction | 22 |
| CommunicationChannel | 117, 125, 148, 151, 154, 161, 169, 192, 216, 224, 227, 272, 594, 607, 94 |
| Crypto | 112 |
| Hardware | 154, 188, 212, 440, 607, 624 |
| SoftwareInput | 28, 74, 113, 116, 123, 137, 153, 173, 240, 242, 248, 441, 549 |
| HardwareInput | 74, 113, 116, 441 |
| NetworkCommunication | 161 |
| PrivilegeOrPermissionAbuse | 233, 458, 554 |
| Software | 154, 188, 212, 607 |
| SupplyChain | 116, 184, 438, 439 |
| HardwareSupply | 440, 624 |

### E. THREAT MODELLING LOGIC

This section presents the inference rules adopted for performing automatic threat modelling. Rules have been written using the SWRL language [40] and represent the logic behind threat modelling automation. Automatic reasoners can apply the defined rules to the ICT infrastructure model to map components to corresponding threats.

SWRL rules can be expressed in a "human-readable" form:

$$antecedent \Rightarrow consequent$$

meaning, if the *antecedent* holds true, then also the *consequent* must also hold. *Antecedent* and *consequent* are expressed in terms of ontology entities and relations.

For each of the identified threat categories (see Section III-D), rules have been provided and summarised in Table 2 using simplified pseudocode. The first column of the table contains the reference threat category; the second column describes the rules, the condition and the logic, determining when an infrastructure component is threatened by that category of threats.

Referring to Table 2, the first row describe the rules (second column) determining the conditions upon which threats in the category `AuthenticationMechanism` (first column) are applied. In details, in the first rule presented the *antecedent* states that if exist an instance of the

**TABLE 2.** Inference rules.

| Threat Category | Rule(s) |
|---|---|
| AuthenticationMechanism | if `Asset isProtectedBy AuthenticationMethod` ⇒ `Asset` is threatened<br>if `DataFlow crosses TrustBoundary` ⇒ `DataFlow` is threatened |
| ClientServerInteraction | if `DataFlow hasSource Software(1)` and `hasDestination Software(2)` ⇒ `DataFlow` is threatened |
| CommunicationChannel | if `DataFlow` ⇒ `DataFlow` is threatened<br>if `Network` ⇒ `Network` is threatened |
| Crypto | if `Asset isProtectedBy EncryptionAlgorithm` ⇒ `Asset` is threatened<br>if `Asset isProtectedBy CryptographicConcept` ⇒ `Asset` is threatened<br>if `DataFlow isProtectedBy EncryptionAlgorithm` ⇒ `DataFlow` is threatened<br>if `DataFlow isProtectedBy CryptographicConcept` ⇒ `DataFlow` is threatened |
| Hardware | if `Hardware` ⇒ `Hardware` is threatened |
| SoftwareInput | if `DataFlow hasDestination Software` ⇒ `Software` is threatened<br>if `DataFlow hasDestination OperatingSystem` ⇒ `OperatingSystem` is threatened<br>if `DataFlow hasDestination Firmware` ⇒ `Firmware` is threatened |
| HardwareInput | if `DataFlow hasDestination Hardware` ⇒ `Hardware` is threatened |
| NetworkCommunication | if `Network` ⇒ `Network` is threatened |
| PrivilegeOrPermissionAbuse | if `DataFlow crosses TrustBoundary` ⇒ `DataFlow` is threatened<br>if `Firmware` ⇒ `Firmware` is threatened<br>if `OperatingSystem` then `OperatingSystem` is threatened |
| Software | if `Software` ⇒ `Software` is threatened |
| SupplyChain | if `ICTEntity dependsOn ExternalService` ⇒ `ExternalService` is threatened<br>if `ICTEntity supply ExternalService` ⇒ `ExternalService` is threatened |
| HardwareSupply | if `ICTEntity dependsOn HardwareSupply` ⇒ `HardwareSupply` is threatened<br>if `ICTEntity supply HardwareSupply` ⇒ `HardwareSupply` is threatened |

entity `Asset` which is connected to an instance of the entity `AuthenticationMechanism` by the relationship `isProtectedBy`, the *consequent* must hold, which state that the instance of the entity `Asset` is threatened by threats in the category `AuthenticationMechanism`. In the same way all the reported rules should be read.

Is important to notice that the presence of security solutions, in the list of rules, does not remove the threats they mitigate. That was a deliberate choice, since, problems related to improper use or implementation of security measures are often the source of attack vectors. Therefore, it was deemed more effective, in the threat modelling phase, to point out specific threat a security solution may introduce but not to remove threat it mitigates. In this way, analysts are provided with a complete set of issues, useful to understand whether the security solutions adopted are effective. Anyway, the possibility to specify the STRIDE categories a security solution mitigates, and the mapping performed between threats and STRIDE, still provides information of which threats are being mitigated by which security solution.

## IV. CASE STUDY

ThreMA was validated using real use cases taken by different Stakeholders of the Italian Public Sector. Fig. 6 shows one target infrastructure used to validate our approach; sensitive data and information have been removed for security and IP protection reasons. The use of ThreMA can be split into two main phases: *system modelling* and *threat modelling*. The former requires human intervention while the latter is completely automated. Automation is possible thanks to the threat catalog provided (see Section III-D) containing known threats affecting infrastructure components, and the inference rules (see Section III-E) defining the logic used in mapping threats to specific components.

Following the sequence diagram presented in Section III-A, once the construction of the metamodel was completed, the ontology was populated with information from the target infrastructure. Referring to Fig. 6, instances for the various hardware components, software, and networks were inserted and the data flow diagram was modelled. The legend in Fig. 6 shows which ontology entity is used to model the given infrastructure asset. Arrows define dataflow direction and, consequently, how it has been modelled with ontology entities and relationships (`DataFlow hasSource` and `hasDestination`). The model created consists of approximately 100 instances, each representing a component of the infrastructure.

Once the model was built, threat modelling was performed. The rules defined for the *Threat Modelling Logic* were applied resorting to the automatic reasoner Pellet [43]. The output represents the *Threat Model* of the infrastructure under analysis, and the process leads to the identification of more than 1,000 threats.

Table 3 summaries the main and most significant results obtained; the first column describes the infrastructure components considered; the second column the number of threats

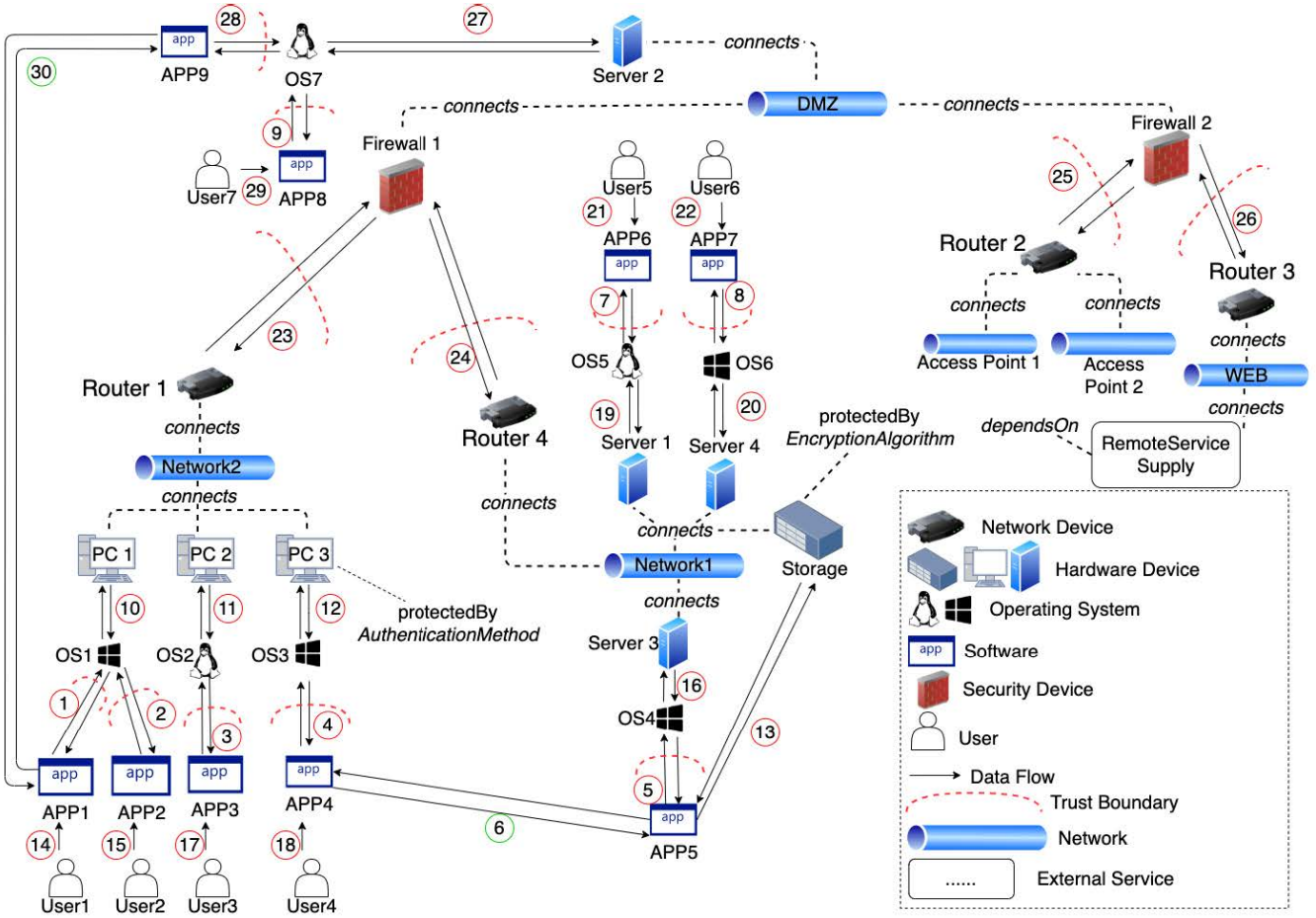


**FIGURE 6.** Use-case Infrastructure.

identified; the third column the categories of the identified threats, and the last column contains the reference to STRIDE categories.

More in detail, all *hardware* devices are affected by threats in the categories `Hardware`, given the type of component, and `HardwareInput`, since they have an input interface. Moreover, *network devices* and *security devices* are also affected by threats in `AuthenticationMechanism` and `PrivilegeOrPermissionAbuse` categories being targets of a dataflow crossing a trust boundary. Finally, instances such as *PC3* and *Storage* (both hardware devices), being protected respectively by an authentication mechanism and an encryption mechanism, are also affected by threats of the corresponding categories.

All *software* instances are affected by threats in categories `Software`, given the type of asset, and `SoftwareInput`, since they implement an input interface. Moreover, *software* instances are also affected by threats in `AuthenticationMechanism` and `PrivilegeOrPermissionAbuse` categories because targets of a dataflow crossing a trust boundary. The same is valid for *operating systems*.

All *dataflow* instances are affected by threats belonging to the `CommunicationChannel` category. Moreover, instances 6 and 29 of `DataFlow` are an example of what falls in the `ClientServerInteraction` category and are then associated with the corresponding threats.

All *network* instances, on top of threats in the `CommunicationChannel` category, are also affected by `Network` threats.

Finally, the *external service*, modelling the use of externally supplied services, is associated with threats in the `SupplyChain` category.

Each threat in the catalog is associated with possible countermeasures that can be adopted to mitigate attacks.

The complete model, the ontology containing both system and threat modelling, constitutes the *security knowledge-base* for the given ICT infrastructure. The generated knowledge base can be queried by security analysts to extract useful information regarding threats affecting the system and possible countermeasures. The mapping between the identified threats and the corresponding STRIDE categories provides an overall summary of the identified issues, while the association between threats and countermeasures allows easier

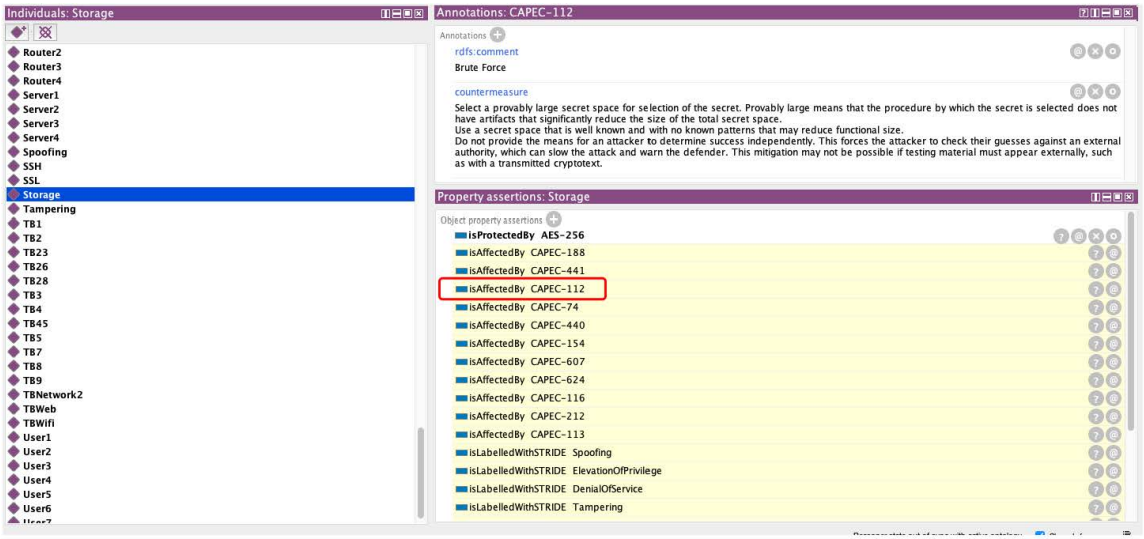


**FIGURE 7.** Protégé windows with extract of results obtained.

**TABLE 3.** Threat modelling results.

| Infrastructure Component | N. of Threats | Threat Categories | STRIDE |
|---|---|---|---|
| Hardware devices | 10 | Hardware, HardwareInput | S T I |
| Operating System | 17 | Software, SoftwareInput AuthenticationMechanism PrivilegeOrPermissionAbuse | S D T I |
| Network | 15 | CommunicationChannel NetworkCommunication | S E D T I |
| Software | 25 | Software, SoftwareInput AuthenticationMechanism PrivilegeOrPermissionAbuse | S E D T I |
| PC3 | 15 | Hardware, HardwareInput AuthenticationMechanism | S E D T I |
| Network devices | 18 | Hardware, HardwareInput Authentication Mechanism PrivilegeOrPermissionAbuse | S E D T I |
| Security devices | 18 | Hardware, HardwareInput Authentication Mechanism PrivilegeOrPermissionAbuse | S E D T I |
| DataFlow | 14 | CommunicationChannel | S E D T I |
| DataFlow 6 and 29 | 15 | CommunicationChannel ClientServerInteraction | S E D T I |
| Storage | 11 | Hardware, Crypto | S E D T I |
| External Service | 4 | SupplyChain | T I |

identification of remediation actions to be adopted and/or implemented.

Fig. 7 contains a screenshot taken from Protégé, the tool used to develop the ontology, populate it and perform threat modelling. It is an example of obtained results; the left window contains the partial list of infrastructure component instances added to the ontology. The bottom right window presents the inferred threat information for the given asset, in this case the *storage*, and the top right window provides information about the selected threat, in this case CAPEC-112, and the identified countermeasures.

## V. CONCLUSION AND FUTURE WORKS

The present paper introduced ThreMA, an ontology-driven threat modelling automation tool for ICT Infrastructures. The underlying ontology metamodel provides a formal vocabulary to model the target ICT infrastructure, a categorisation of threats tightly linked with infrastructure components, and the threat modelling logic to bind everything together. The solution can be easily integrated within an operation pipeline, providing a cybersecurity knowledge-base of the infrastructure under analysis that can be exploited in different phases of its life-cycle, from design to post-production security evaluations. The provided description vocabulary can be shared between different actors during the security evaluation and prevents the introduction of possible modelling errors or misunderstandings that would lead to sub-optimal results.

In ThreMA, the threat modelling logic is expressed and performed through SWRL rules and applied using ontology automatic reasoners. This is one of the major points of our work. The use of an ontology and inference rules provides a syntactical way of describing the problem that mimics the typical expert's way of thinking. Extensibility and maintainability greatly benefit from this aspect, no coding is required, and security experts can focus on describing the underlying logic of threat modelling providing faster integration in a rapidly changing context where new threats are constantly being discovered.

Concerning ThreMA, there are several aspects to be investigated in future works. Extend the threat portfolio and categories by considering other sources of information besides

CAPEC, by integrating Cyber Threat Intelligence (CTI) information in the process of threat identification [44]. Integrate solutions focusing on specific domain verticals, such as [28], to extend the applicability of the proposed solution.

Finally, a great challenge will be the modelling of two knowledge domains, often overlooked in threat modelling: the hardware domain with its peculiar threat [45] and the human factor [46].

## REFERENCES


[1] A. Refsdal, B. Solhaug, and K. Stølen, *Cyber-Risk Management*. Cham, Switzerland: Springer, 2015, pp. 33–47.

[2] *Information Technology—Security Techniques—Information Security Risk Management*, Standard ISO/IEC 27005, Int. Org. Standardization, Geneva, Switzerland, 2016. [Online]. Available: https://www.iso.org/standard/75281.html

[3] *Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy*, Standard SP 800-37 Rev. 2, Nat. Inst. Standard Technol., Gaithersburg, MD, USA, 2018. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-37/rev-2/final

[4] *Information Security Management Systems—Guidelines for Information Security Risk Management*, Standard BS 7799-3:2017, British Standards Institution, London, U.K., 2017. [Online]. Available: https://knowledge.bsigroup.com/products/information-security-management-systems-guidelines-for-information-security-risk-management-1/standard

[5] (2022). *Interoperable Eu Risk Management Framework*. European Union Agency for Cybersecurity. Attiki. GRC. [Online]. Available: https://www.enisa.europa.eu/publications/interoperable-eu-risk-management-framework

[6] I. Linkov and A. Kott, "Fundamental concepts of cyber resilience: Introduction and overview," in *Cyber Resilience of Systems and Networks*, 1st ed. Springer, Jun. 2019, pp. 1–25, doi: 10.1007/978-3-319-77492-3_1. and doi: 10.1007/978-3-319-77492-3.

[7] S. Moral-García, S. Moral-Rubio, D. G. Rosado, E. B. Fernández, and E. Fernández-Medina, "Enterprise security pattern: A new type of security pattern," *Secur. Commun. Netw.*, vol. 7, no. 11, pp. 1670–1690, Nov. 2014.

[8] J. A. Ingalsbe, L. Kunimatsu, T. Baeten, and N. R. Mead, "Threat modeling: Diving into the deep end," *IEEE Softw.*, vol. 25, no. 1, pp. 28–34, Jan. 2008.

[9] S. Aier, S. Buckl, U. Franke, B. Gleichauf, P. Johnson, P. Närman, C. M. Schweda, and J. Ullberg, "A survival analysis of application life spans based on enterprise architecture models," in *Proc. Enterprise Modelling Inf. Syst. Archit.*, vol. LNI P-152, Jun. 2009, pp. 141–154. [Online]. Available: https://www.alexandria.unisg.ch/id/eprint/213537

[10] M. Ulsch, *Cyber Threat!: How to Manage Growing Risk Cyber Attacks*. Hoboken, NJ, USA: Wiley, 2014.

[11] K. Bergmann, "The growing cyber threat," *Asia–Pacific Defence Reporter*, vol. 40, no. 2, pp. 12–13, 2014.

[12] M. Frydman, G. Ruiz, E. Heymann, E. César, and B. P. Miller, "Automating risk analysis of software design models," *Sci. World J.*, vol. 2014, pp. 1–12, Oct. 2014.

[13] W. Xiong and R. Lagerström, "Threat modeling—A systematic literature review," *Comput. Secur.*, vol. 84, pp. 53–69, Jul. 2019.

[14] V. Casola, A. De Benedictis, M. Rak, and U. Villano, "Toward the automation of threat modeling and risk assessment in IoT systems," *Internet Things*, vol. 7, Sep. 2019, Art. no. 100056.

[15] I. Zografopoulos, J. Ospina, X. Liu, and C. Konstantinou, "Cyber-physical energy systems security: Threat modeling, risk assessment, resources, metrics, and case studies," *IEEE Access*, vol. 9, pp. 29775–29818, 2021.

[16] S. Manzoor, T. Vateva-Gurova, R. Trapero, and N. Suri, "Threat modeling the cloud: An ontology based approach," in *Proc. Int. Workshop Inf. Oper. Technol. Secur. Syst.* Cham, Switzerland: Springer, 2018, pp. 61–72.

[17] A. Yeboah-Ofori and S. Islam, "Cyber security threat modeling for supply chain organizational environments," *Future Internet*, vol. 11, no. 3, p. 63, Mar. 2019.

[18] A. Ekelhart, S. Fenz, M. D. Klemen, and E. R. Weippl, "Security ontology: Simulating threats to corporate assets," in *Proc. Int. Conf. Inf. Syst. Secur.* Cham, Switzerland: Springer, 2006, pp. 249–259.

[19] M. Välja, F. Heiding, U. Franke, and R. Lagerström, "Automating threat modeling using an ontology framework," *Cybersecurity*, vol. 3, no. 1, pp. 1–20, Dec. 2020.

[20] A. Marback, H. Do, K. He, S. Kondamarri, and D. Xu, "A threat model-based approach to security testing," *Softw., Pract. Exper.*, vol. 43, no. 2, pp. 241–258, Feb. 2013.

[21] Exabeam. *Top 8 Threat Modeling Methodologies and Techniques*. Accessed: Jul. 30, 2022. [Online]. Available: https://www.exabeam.com/information-security/threat-modeling/

[22] N. Shevchenko, T. A. Chick, P. O'Riordan, T. P. Scanlon, and C. Woody, "Threat modeling: A summary of available methods," Carnegie Mellon Univ., Softw. Eng. Inst., Pittsburgh, PA, USA, Tech. Rep., 2018. [Online]. Available: https://apps.dtic.mil/sti/citations/AD1084024

[23] Q. Li and Y.-L. Chen, "Data flow diagram," in *Modeling and Analysis of Enterprise and Information Systems*. Cham, Switzerland: Springer, 2009, pp. 85–97.

[24] Microsoft. *Microsoft Threat Modeling Tool—Threats*. Accessed: Jul. 30, 2022. [Online]. Available: https://docs.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats

[25] Microsoft. *Getting Started With the Threat Modeling Tool*. Accessed: Jul. 30, 2022. [Online]. Available: https://docs.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-getting-started

[26] B. Schneier, "Attack trees," *Dr. Dobb's J.*, vol. 24, no. 12, pp. 21–29, 1999.

[27] T. UcedaVelez and M. M. Morana, *Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis*. Hoboken, NJ, USA: Wiley, 2015.

[28] V. Casola, A. D. Benedictis, C. Mazzocca, and R. Montanari, "Toward automated threat modeling of edge computing systems," in *Proc. IEEE Int. Conf. Cyber Secur. Resilience (CSR)*, Jul. 2021, pp. 135–140.

[29] L. H. Fla, R. Borgaonkar, I. A. Tøndel, and M. Gilje Jaatun, "Tool-assisted threat modeling for smart grid cyber security," in *Proc. Int. Conf. Cyber Situational Awareness, Data Anal. Assessment (CyberSA)*, Jun. 2021, pp. 1–8.

[30] B. Andrei and O. Evgeny, "Framework for ontology-driven threat modelling of modern computer systems," *Int. J. Open Inf. Technol.*, vol. 8, no. 2, pp. 14–20, 2020.

[31] A. Brazhuk, "Towards automation of threat modeling based on a semantic model of attack patterns and weaknesses," 2021, *arXiv:2112.04231*.

[32] B. Andrei, "Threat modeling of cloud systems with ontological security pattern catalog," *Int. J. Open Inf. Technol.*, vol. 9, no. 5, pp. 36–41, 2021.

[33] P. Kamongi, M. Gomathisankaran, and K. Kavi, "Nemesis: Automated architecture for threat modeling and risk assessment for cloud computing," in *Proc. 6th ASE Int. Conf. Privacy, Secur., Risk Trust (PASSAT)*, 2014, pp. 1–10.

[34] P. Kamongi, S. Kotikela, K. Kavi, M. Gomathisankaran, and A. Singhal, "VULCAN: Vulnerability assessment framework for cloud computing," in *Proc. IEEE 7th Int. Conf. Softw. Secur. Rel.*, Jun. 2013, pp. 218–226.

[35] P. Salini and J. Shenbagam, "Prediction and classification of web application attacks using vulnerability ontology," *Int. J. Comput. Appl.*, vol. 116, no. 21, pp. 42–47, Apr. 2015.

[36] C. E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi, "A taxonomy of computer program security flaws," *ACM Comput. Surveys*, vol. 26, no. 3, pp. 211–254, Sep. 1994.

[37] R. Luh, S. Schrittwieser, and S. Marschalek, "TAON: An ontology-based approach to mitigating targeted attacks," in *Proc. 18th Int. Conf. Inf. Integr. Web Appl. Services*, Nov. 2016, pp. 303–312.

[38] B. A. Sabbagh and S. Kowalski, "A socio-technical framework for threat modeling a software supply chain," *IEEE Secur. Privacy*, vol. 13, no. 4, pp. 30–39, Jul. 2015.

[39] W. O. W. Group. *OWL 2 Web Ontology Language Document Overview (Second Edition)*. Accessed: Jul. 30, 2022. [Online]. Available: https://www.w3.org/TR/owl2-overview/

[40] H. Boley, M. Dean, B. Grosof, I. Horrocks, P. F. Patel-Shcneider, and S. Tabet. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. Accessed: Jul. 30, 2022. [Online]. Available: https://www.w3.org/Submission/SWRL/

[41] N. Messe, V. Chiprianov, N. Belloir, J. El-Hachem, R. Fleurquin, and S. Sadou, "Asset-oriented threat modeling," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Dec. 2020, pp. 491–501.

[42] B. C. Jonathan. *Capec-Stride Mapping*. Accessed: Jul. 30, 2022. [Online]. Available: https://ostering.com/blog/2022/03/07/capec-stride-mapping/

[43] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical OWL-DL reasoner," *J. Web Semantics*, vol. 5, no. 2, pp. 51–53, 2007.

[44] W. Xiong, E. Legrand, O. Åberg, and R. Lagerström, "Cyber security threat modeling based on the MITRE enterprise ATT&CK matrix," *Softw. Syst. Model.*, vol. 21, no. 1, pp. 157–177, Feb. 2022.
[45] W. Hu, C.-H. Chang, A. Sengupta, S. Bhunia, R. Kastner, and H. Li, "An overview of hardware security and trust: Threats, countermeasures, and design tools," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 6, pp. 1010–1038, Jun. 2021.
[46] L. Hadlington, "The 'human factor' in cybersecurity: Exploring the accidental insider," in *Research Anthology on Artificial Intelligence Applications in Security*. Hershey, PA, USA: IGI Global, 2021, pp. 1960–1977.

**FABIO DE ROSA** received the B.S., M.S., and Ph.D. degrees in computer science from University La Sapienza, Italy.

He is currently a DPO and CIS Architect and a Researcher at the CINI–Cybersecurity National Laboratory. He is a Seasoned Enterprise and a Cyber and Information Security (CIS) Architect, with over 20 years of experience in research, analysis, design of distributed systems, adaptive business process management, pervasive computing, service oriented computing, service oriented architecture, and SOA and system testing; and CIS methodologies, techniques, and technologies that he applied to national and international real use cases. He has acquired competences on different enterprise and methodological frameworks (NIST and Italian Cybersecurity and Privacy Frameworks, RUP, OMT, Zachman–NAF only theoretical knowledge), languages (SBVR and UML) and tools about distributed systems and network architecture design and development, skills on very large suite of proprietary and open source components, and products on different technological platforms in supporting EIS and EIA. He has been an official trainer of ORACLE SOA Suite 10g and 11g for Oracle University Italia.

He was the coauthor of Simple Engineering SOA/BPM methodological frameworks: simpleSOAD and simpleSOA. His current research interests include consulting, advising, cyber and information security, data breach, data loss prevention issues, and interdisciplinary aspects of cybersecurity and privacy. In the past, he has actively participated in several working groups of Health Level Seven (HL7), Italy, and indirectly to international ones.

**NICOLÒ MAUNERO** received the M.S. degree in embedded systems from the Polytechnic of Turin, Italy, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Control and Computer Engineering. He is a member of the Cybersecurity National Laboratory of CINI and the Italian National Consortium for Informatics. He started his research career during his master's degree. He started his research activity on hardware and embedded system security. Since 2020, he has been focusing on cyber risk management and cybersecurity ontologies.

**PAOLO PRINETTO** (Senior Member, IEEE) received the M.S. degree in electronic engineering from the Polytechnic of Turin, Italy, in 1997.

He is a Full Professor of computer engineering at the Polytechnic of Turin, (50%) and the IMT–Institute for Advanced Studies Lucca, Italy, (50%). He is currently serving as the Director at the CINI–Cybersecurity National Laboratory, a Coordinator of the programs CyberChallenge, OliCyber, and CyberHighSchools. He is serving as an Italian Representative with the Civil Security for Society Sub-Group, EU Shadow Strategic Programme Committee of Horizon Europe–the Framework Programme for Research and Innovation. His research interests include hardware security, digital systems design and test, and system dependability.

Dr. Prinetto was honored of the title "Doctor Honoris Causa" of the Technical University of Cluj-Napoca (Romania), in 2012. From 2010 to 2014, he served as an Appointed Member of the Scientific Committee of the French "Centre National de la Recherche Scientifique" (CNRS). From 2013 to 2019, he served as the President for CINI (Italian National Inter–University Consortium for Informatics). From 2013 to 2019, he served as the Vice-Chair for the International Federation for Information Processing (IFIP) Technical Committee TC 10–Computer Systems Technology. In 2000 and 2003, he served as the Elected Chair for the IEEE–Computer Society Test Technology Technical Council (TTTC).

**FEDERICO TALENTINO** received the B.S. degree in computer engineering and the M.S. degree in cybersecurity from the Polytechnic of Turin, Italy, in 2022. He is a Junior Researcher at the CINI–Cybersecurity National Laboratory, Italy. His research interests include vulnerability assessment and cybersecurity ontologies.

**MARTINA TRUSSONI** received the B.S. degree in computer engineering and the M.S. degree in cybersecurity from the Polytechnic of Turin, Italy, in 2022. She is a Junior Researcher at the CINI–Cybersecurity National Laboratory, Italy. Her research interests include cyber risk management and cybersecurity ontologies.

Open Access funding provided by 'Politecnico di Torino' within the CRUI CARE Agreement