An Ontology-based Approach for Automatic Specification, Verification, and Validation of Software Security Requirements: Preliminary Results

Dimitrios Tsoukalas | tsoukj@iti.gr
Centre for Research and Technology Hellas
• Introduction and Problem Statement

• IoTAC SSD Platform

• Security Requirements Specification

• Security Requirements V&V

• Test Cases Demonstration

• Conclusions and Future Work
• Problem:
  • A significant portion of software vulnerabilities are introduced during the Design and Requirements phase, mainly due to incorrect, unclear, or missing security requirements.
  • There is a strong need in the community for tools that facilitate software engineers in specifying, verifying, and validating security requirements.
  • Most of the existing methods either i) rely on theoretical methodologies, reusable templates, and formal languages, or ii) require a lot of manual effort, which hinders their practicality.

In the present work we introduce two mechanisms, able to automatically:

• Facilitate the correct Specification of software security requirements expressed in natural language

• Verify and Validate (V&V) the defined software security requirements and recommend potential refinements and/or additions
Introduction and Problem Statement

IoTAC SSD Platform

Security Requirements Specification

Security Requirements V&V

Test Cases Demonstration

Conclusions and Future Work
The present work comprises preliminary results in the context of the ongoing research project IoTAC, funded by the European Commission (EC) under Horizon 2020.

**IoTAC Software Security by Design (SSD) Platform**

Provides novel solutions for monitoring and optimizing the security of IoT Software Applications throughout their SDLC, and certifying their security level.
• Introduction and Problem Statement

• IoTAC SSD Platform

• Security Requirements Specification

• Security Requirements V&V

• Test Cases Demonstration

• Conclusions and Future Work
Security Requirements Specification

• **Purpose:**
  - Facilitating the correct definition/specification of software security requirements.

• **The overall pipeline is summarized in the following four steps:**
  - **Input:** a set of Software Security Requirements (Natural Language)
  - **Syntactic analysis:** identify the grammatical category of each word (e.g. noun, verb, etc.), as well the grammatical relations among them (subject, verb, object) using the Mate tools
  - **Semantic analysis:** identify the main requirements concepts (Action, Actor, Properties, etc.) using semantic role labelling
  - **Output:** Machine-readable version of the requirements and storage into an Ontology

• An Ontology for storing Software Security Requirements has been set up and a schema has been devised.
• An **Ontology** for storing Software Security Requirements has been set up

• A **schema** has been devised based on our research

• The same schema was also used to build the **Security Requirements Knowledge Base**, which contains a carefully curated set of security requirements used as a reference point for V&V tasks

• More than 200 curated software security requirements have been collected (up to now) from i) **repositories** of well-defined security requirements and ii) relevant security **standards**
Security Requirements Specification Component

**Syntactic Analysis**
- Tokenization
- Lemmatization
- Part-of-speech Tagging
- Dependency Parsing

**Semantic Analysis**
- Semantic Role Labeling
- Identify instances of Actions for each Action
- Identify instances of Actor, Objects and Properties

Input: Security Requirement (in NL)
Output: Ontology Instances
Security Requirements Specification

• Requirement:
  • “The system shall authenticate users prior to accessing an application or data.”

• Syntactic Analysis:

• Semantic Analysis:
• Introduction and Problem Statement
• IoTAC SSD Platform
• Security Requirements Specification
• Security Requirements V&V
• Test Cases Demonstration
• Conclusions and Future Work
Security Requirements V&V

• **Purpose:**
  - Verifying and validating the correctness and completeness of the defined software security requirements, as well as for recommending potential refinements

• **The overall pipeline is summarized in the following three steps:**
  - **Input:** the set of user-defined security requirements expressed in a machine-readable form (i.e., the result of the SSRS component)
  - **Similarity Checks:** employ word similarity models for comparing the requirements provided as input with a curated security requirements set (i.e., the Security Requirements Knowledge base) in order to assess their correctness and completeness
  - **Output:** recommendations regarding the improvement of the initial set of user-defined requirements, including similar or additional curated requirements and suggestions of alternative priority terms
Security Requirements Verification & Validation Component

Input
Ontology Instances of Sec. req. provided by the user

If Similarity > threshold

Similarity between new Ontology Instances and stored Instances of the curated Requirements

Obtain corresponding Curated Requirements

Obtain corresponding Priority Term

Output
Recommendations for Improvement

QRS 2021 - STV Workshop, December 6-10, 2021
**Security Requirements V&V**

- **Requirement (loosely specified):**
  - "The system could authenticate users."

- **Similarity Checks:**
  - Uses natural language processing toolkits (e.g., WordNet with NLTK) to find the word similarity between the concepts (i.e., Actors, Actions, etc.) of the user-defined requirement and the corresponding concepts of each curated requirement stored into the Security Requirements Knowledge base.

  If similarity scores (sim) between req. concepts are higher than a predefined threshold, then the curated requirement is presented to the user as recommendation.
Security Requirements V&V

• Final Output:

“The system could authenticate users.”

Rec. 1 - Consider **replacing** the initial loosely specified security requirement with the following:
✓ “The system shall authenticate users prior to accessing an application or data.”

Rec. 2 - Consider **to complement** the initial security requirements set with the following:
✓ “The system shall authenticate users using at least one of the following authentication mechanisms: username/password, digital certificate, secure token or biometrics.”
✓ “The system shall allow security administrators to grant authorization to users.”
✓ “The system shall provide the ability to extract sensitive record information only to authorized users.”

Rec. 3 - Consider the priority term “SHALL” that better reflects the severity level
• Introduction and Problem Statement

• IoTAC SSD Platform

• Security Requirements Specification

• Security Requirements V&V

• Test Cases Demonstration

• Conclusions and Future Work
• Introduction and Problem Statement

• IoTAC SSD Platform

• Security Requirements Specification

• Security Requirements V&V

• Test Cases Demonstration

• Conclusions and Future Work
Conclusions and Future Work

• **Conclusions:**
  - We introduced two new mechanisms for facilitating the specification, verification, and validation of software security requirements
  - We implemented the proposed mechanisms as prototype tools in the form of web services and demonstrated their capabilities through test cases

• **Future Work:**
  - Evaluation of the accuracy of the proposed mechanisms through a large-scale study, utilizing real security requirements retrieved from documents of actual software products
  - Involvement of security experts in order to enhance the reliability of the evaluation results
  - Further refinement of the semantic role labelling rules and similarity thresholds integrated into the two proposed mechanisms
  - Integration of the proposed mechanisms to a dedicated user interface, facilitating their usage in practice
Thank you!

Q&A

Dimitrios Tsoukalas | tsoukj@iti.gr
Centre for Research and Technology Hellas