

SecTro: A CASE Tool for Modelling Security in Requirements Engineering using Secure Tropos

Michalis Pavlidis and Shareeful Islam

School of Computing, IT and Engineering, University of East London, UK
m.pavlidis@ieee.org, shareeful@uel.ac.uk

Abstract. Secure Tropos is an extension of Tropos methodology, which considers security throughout the whole development process. The main concept of Secure Tropos is the security constraint that captures constraints regarding security. Similarly, the concepts of dependency, goal, task, resource, and capability were also extended with security in mind. In this paper we present the SecTro tool, a CASE tool that guides and supports the developers in the construction of the appropriate models of Secure Tropos.

Keywords: Security, goal modelling, requirements engineering, Secure Tropos, CASE tools.

1 Introduction

As the use of information systems is increasing rapidly everyday in finance, military, education, health care, and transportation, the need of security is increasing respectively. The stored information in many cases is sensitive and has to be secured by protecting it from any attack. In other words, there should be cost effective and operationally effective protection from undesirable events [1].

It is already agreed by the industry and research community, that security has to be considered from the early phases of the software development process [2]. Having defined the security requirements along with the functional requirements will enable the better comprehension of the system's security issues and limit the conflicts between the security and functional requirements for more secure information systems [1].

Secure Tropos is a security requirements engineering methodology that considers security throughout the whole development process [1]. The approach identifies, models and analyses the security issues from the early stages of software development within the organization and social settings [2]. But, the fact that it considers security from the early stages of software development, results in a serious increase of the activities in the software development stages and therefore requires the existence of a software tool to support the development process [2]. This paper demonstrates a tool, named SecTro, which assists the security analysts in constructing the relevant Secure Tropos diagrams that are required in order to identify, model and analyze the security issues.

The rest of the paper is structured as follows. Section 2 is a review on Secure Tropos. Section 3 illustrates the tool that supports Secure Tropos. Section 4 discusses the related work while section 5 concludes the paper and presents future work.

2 Secure Tropos Methodology

Secure Tropos is an extension of Tropos methodology that takes security into account and is based on the concept of security constraint. Also, the Tropos concepts of dependency, goal, task, resource, and capability were also extended with security in mind and formed the secure entities [1, 3]. Secure Tropos includes the following modelling activities, the security reference modelling, the security constraint modelling, the secure entities modelling, and the secure capability modelling. In addition, it consists of four stages, the early requirements, the late requirements, the architectural design, and the detailed design stages. The metamodel of Secure Tropos [4] is shown in Fig. 1 and for a more detailed description of Secure Tropos please refer to [1], [3].

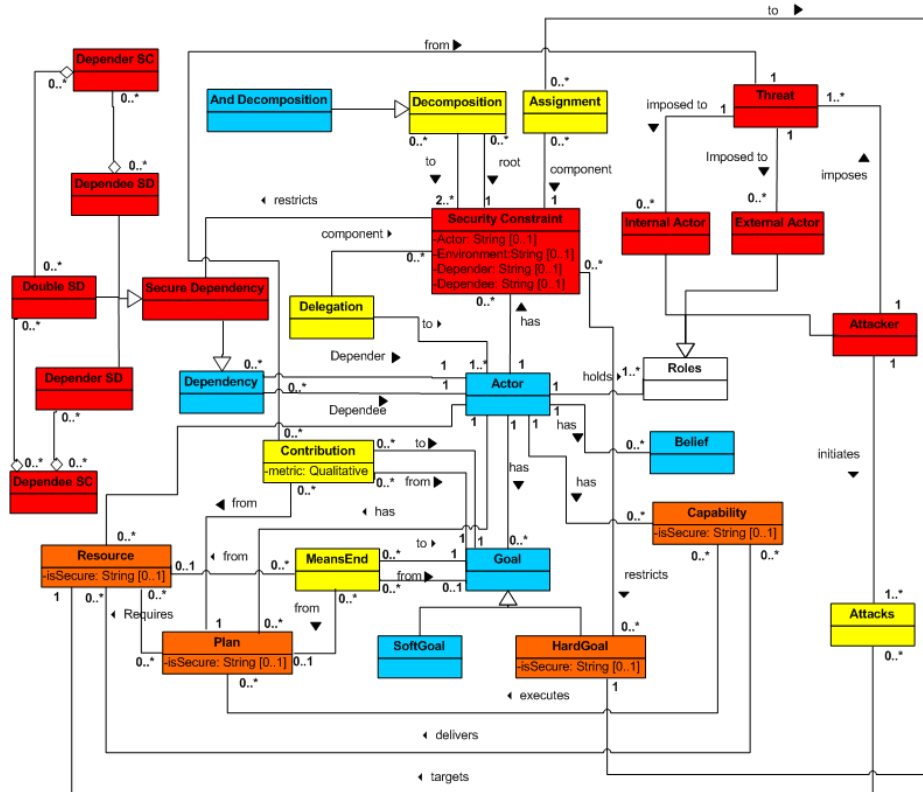


Fig.1. Secure Tropos metamodel.

3 The SecTro Tool

3.1. SecTro Architecture

SecTro is a standalone application that was built with the Java programming language making it a portable application across different platforms. The package diagram is shown in Fig. 2 and descriptions of the packages are given in Table 1. The class diagram of the classes that are responsible for the drawing functionality of the tool is shown in Fig. 3. In the ElementType class belong all the elements that can be drawn, such as an actor and a hard goal, and in the LinkType class belong all the links between the elements, such as the “plays” link and the “satisfies” link. The class diagram of the graphical user interface (GUI) package is shown in Fig. 4.

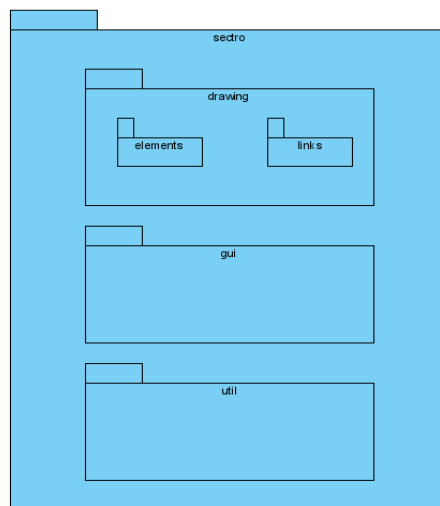


Fig. 2. Package diagram of SecTro.

Table 1. Description of the SecTro packages.

Package	Description
sectro	The parent package that includes the main class and all the sub packages
sectro.drawing	Contains the generalized class for all the drawing objects (DrawingObject) and the elements and links packages
sectro.drawing.elements	Contains the classes for all the drawing elements (Actor, HardGoal, Resource, Plan, etc.)
sectro.drawing.links	Contains the generalized class for all the Links (Link) and the classes for all the drawing links (LinkDependency, LinkRestricts, LinkPlays, etc.)
sectro.gui	Contains all the classes related to the user interface (MainForm, ToolBar, MenuBar, etc.)
sectro.util	Contains all the utility classes (ImageUtil, XMLUtil, FileUtil, etc.)

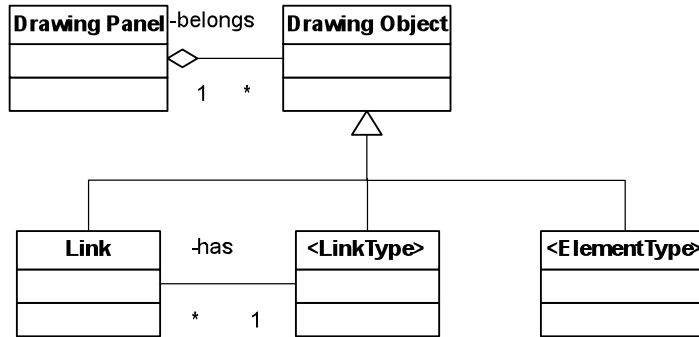


Fig. 3. Class diagram of the SecTro drawing functionality.

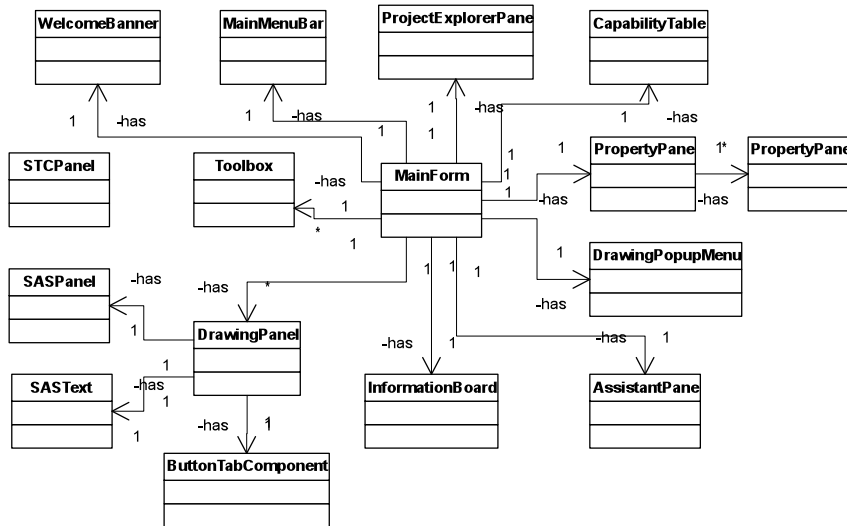


Fig. 4. SecTro GUI class diagram.

3.2. SecTro Layout and Functionalities

SecTro’s workspace (Fig. 5) consists of the drawing canvas in the centre, on the top there is a series of tabs for showing the developed diagrams for each stage of Secure Tropos, the project explorer and the properties panel are on the right side, the toolbox (Fig. 6) is on the left side, and the SecTro assistant at the bottom of the workspace. The graphical representations of all the concepts of Secure Tropos by the SecTro tool are shown in Fig. 7 and the graphical representation of the secure dependency is shown in Fig. 8.

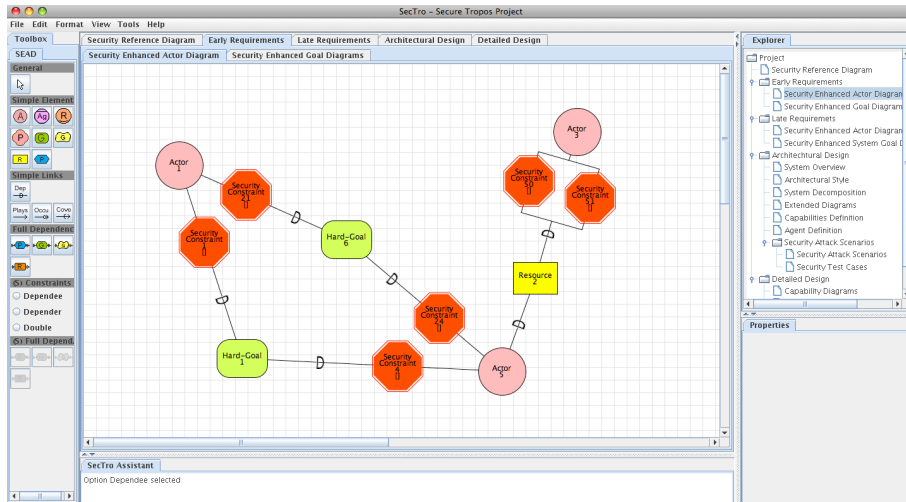


Fig. 5. SecTro workspace.

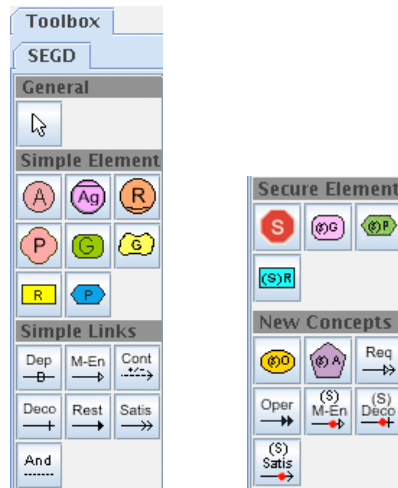


Fig. 6. SecTro toolbox.



Fig. 7. Secure Tropos notation.

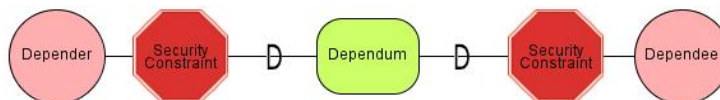


Fig. 8. Secure Dependency.

The main functionalities of the SecTro are to support the developer in the modelling activities of Secure Tropos. Therefore, the tool enables the developer to perform security reference modelling (Fig. 9), security constraint modelling (Fig. 10), secure entities modelling (Fig. 11), and secure capability modelling. During these activities the tool has a mechanism for checking the rules and constraints and informs the developer for any error. Also, the SecTro assistant panel shows more information about the rules and constrains, the concepts and the meta-models. In this way it assists the developer in the learning process of Secure Tropos methodology. Furthermore, the tool enables the developer to export the diagrams as images and in XML format.

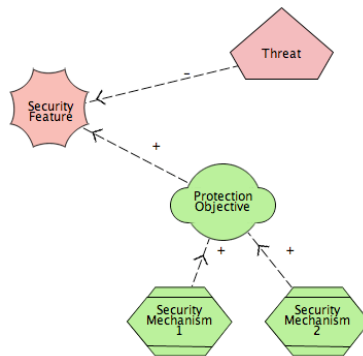


Fig. 9. Security reference modelling.

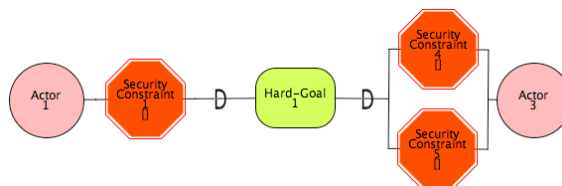


Fig. 10. Security constraint modelling.

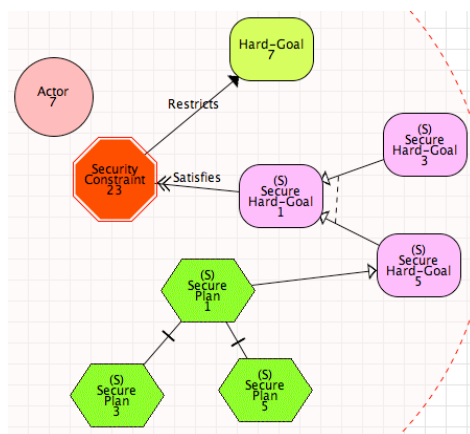


Fig. 11. Secure entities modelling.

During the architectural design the architecture of the system is defined. The tool can automatically generate the architecture style and the system decomposition. However, the activities of the architectural design can be a very difficult task for a developer without knowledge of security. Finally, in most cases, during the end of the architectural design the security attack testing takes places, where the design of the system is tested against the security requirements [5]. The tool automatically generates for the developer the security attack scenario template and the security test case template.

4 Related Work

Although Secure Tropos is still in research and it is difficult to develop a CASE tool for a methodology that is still in research, the i* modelling framework has been out for some years and a number of related CASE tools were developed to support it. OME [6], OpenOME [7], REDEPEND-REACT [8], TAOM4e [9], GR-Tool [10], T-Tool [9], ST-Tool [11], J-PRiM [12], jUCMNav [13], SNet Tool [14], and DesCARTES [15] are some examples of such tools.

The aforementioned tools, although they were developed for different ultimate purposes, they all provide support for the i* modelling framework, which is the modelling framework that was adopted by Secure Tropos as well. But, Secure Tropos introduces new concepts that none of the previous tools enables their graphical representation, i.e. security constraint, secure goal, secure plan, secure resource, and secure capability. Also, the previous tools don't provide support for the modelling activities that Secure Tropos introduces, i.e. security constraint modelling, secure entities modelling, and secure capability modelling. So, despite the fact that experienced users with Secure Tropos can make conventions and use the previous tools to construct single diagrams; these tools are not adequate to support the Secure Tropos methodology.

5 Conclusions and Future Work

The tool supports the developers in the modelling activities of the early and late requirements and architectural design stages of Secure Tropos by assisting them in the construction of the relevant concepts and models that are required during the new modelling activities. Its user-friendly interface makes it easy to use and assists security analysts who are not familiar with the methodology, by providing them with information about the methodology concepts, stages, and metamodels. Also, it enforces rules and constraints and provides valuable feedback on various actions of the developers in an interactive way. The tool has already been used by the students of university of East London to model and analyse security issues of a real industry case study. However, the tool does not support the modelling activities of the detailed design stage and we consider this as future work. In addition, future work includes the extension of the XML Schema in order to validate more models of the methodology.

References

1. Mouratidis, H., Giorgini, P.: Secure Tropos: A Security-Oriented Extension of the Tropos Methodology. *International Journal of Software Engineering and Knowledge Engineering* 17(2), pp. 285-309 (2007)
2. Mouratidis, H., Giorgini, P.: Integrating Security and Software Engineering: Future Vision and Challenges. In: Mouratidis, H., Giorgini, P. (eds.) *Integrating Security and Software Engineering: Advances and Future Visions*. Idea Group Publishing, London (2007)
3. Giorgini, P., Mouratidis, H., Zannone, N.: Modelling Security and Trust with Secure Tropos. In: Mouratidis, H., Giorgini, P. (eds.) *Integrating Security and Software Engineering: Advances and Future Visions*. Idea Group Publishing, London (2007)
4. Matulevicius, R.: Summary of Secure Tropos Metamodel. Internal Report, University of Namur (2008)
5. Mouratidis, H., Giorgini, P.: Security Attack Testing (SAT) – Testing the Security of Information Systems at Design Time. *Journal of Information Systems* 32, pp. 1166-1183 (2007)
6. OME3, <http://www.cs.toronto.edu/km/ome/>
7. OpenOME, <https://se.cs.toronto.edu/trac/ome/>
8. Grau, G., Franch, X., Maiden, N.: REDEPEND-REACT: An Architecture Analysis Tool. In: 13th IEEE International Conference on Requirements Engineering, pp. 455-456. Paris (2005)
9. Morandini, M., Nguyen, C.D., Perini, A., Siena, A., Susi, A.: Tool-supported Development with Tropos: The Conference Management System Case Study. In: Luck, M., Padgham, L. (eds.) *AOSE 2007*. LNCS, vol. 4951, pp.182-196, Springer, Heidelberg (2008)
10. Giorgini, P., Mylopoulos, J., Sebastiani, R.: Goal-Oriented Requirements Analysis and Reasoning in Tropos Methodology. *Journal of Engineering Applications of Artificial Intelligence* 18(2), pp. 159-171 (2005)
11. Massaci, F., Mylopoulos, J., Zanone, N.: Computer-Aided Support for Secure Tropos. *Journal of Automated Software Engineering* 14(2), 341-364 (2007)
12. Grau, G., Franch, X., Avila, S.: J-PRiM: A Java Tool for a Process Reengineering i* Methodology. In: 14th IEEE International Conference on Requirements Engineering, pp. 359-360. Minneapolis (2006)
13. Mussbacher, G., Amyot, D.: Assessing the Applicability of Use Case Maps for Business Process and Workflow Description. In: 2008 International MCETECH Conference on e-Technologies, pp. 219-222. Montreal (2008)
14. Gans, G., Lakemeyer, G., Jarke, M., Vits, T.: SNet: A Modeling and Simulation Environment for Agent Networks Based on i* and ConGolog. In: Proceedings of the 14th International Conference on Advanced Information Systems Engineering, pp. 328-323 (2002)
15. UCL/ISYS - DesCARTES Architect, <http://www.isys.ucl.ac.be/descartes/index.php>