# Holistic Security Requirements Analysis: An Attacker's Perspective

Tong Li, Elda Paja, John Mylopoulos
University of Trento, Italy
{tong.li, paja, jm}@disi.unitn.it

Jennifer Horkoff
City University London, UK
horkoff@city.ac.uk

Kristian Beckers
Technische Universität München, Germany
beckersk@in.tum.de

***Abstract*—The ever-growing complexity of systems makes their protection more challenging, as a single vulnerability or exposure of any component of the system can lead to serious security breaches. This problem is exacerbated by the fact that the system development community has not kept up with advances in attack knowledge. In this demo paper, we propose a holistic attack analysis approach to identify and tackle both atomic and multistage attacks, taking into account not only software attacks but also attacks that are targeted at people and hardware. To bridge the knowledge gap between attackers and defenders, we systematically analyze and refine the malicious desires of attackers (i.e., anti-goals), and leverage a comprehensive attack pattern repository (CAPEC) to operationalize attacker goals into concrete attack actions. Based on the results of our attack analysis, appropriate security controls can be selected to effectively tackle potential attacks.**

## I. Introduction

Socio-Technical Systems (STSs), which consist of people, business processes, software applications, and physical infrastructure, have suffered from a variety of attacks, as attackers are able to breach system security by targeting any of those components. Take a smart meter system as an example [1]. An attacker can access energy consumption data by performing social engineering against the stakeholders, by intercepting communication data transmitted between software applications, or even by probing the physical smart meter device. The larger attack surfaces of STSs are exposed to multistage attacks, which are harder to defend against.

Thinking like an attacker constitutes an effective way to discover attacks in order to produce secure systems [2]. Many approaches have been proposed to analyze security requirements from an attacker's perspective, such as anti-goal analysis [3] and misuse cases [4]. However, those approaches do not explicitly capture interrelations among various system components (e.g., people, software, and hardware), and cannot holistically analyze attacks for socio-technical systems.

Another obstacle to STS security is that attack analysis lacks knowledge of impending attacks. Barnum and Sethi have pointed out that the software engineering community has not kept up with advances in attack knowledge, which leads to less effective or even useless security designs [2]. Attack patterns are provided as solutions to this problem, which are developed to document reusable attack knowledge in support of system security solutions. Specifically, CAPEC (Common Attack Pattern Enumeration and Classification) is a comprehensive attack knowledge repository, which includes 463 attack patterns[1]. However, without an efficient method to utilize the large amount of attack patterns, analysts are reluctant to adopt them in practice [5].

In this paper, we propose to analyze attacks from a holistic viewpoint based on a three-layer requirements framework [6], which involves a social layer, a software layer, an infrastructure layer, as well as connections among these layers. In particular, our approach makes the following contributions:

- takes into account threats across all the three layers to provide a holistic security analysis.
- systematically analyzes attacker malicious desires in order to explore not only atomic attacks within a specific layer, but also multistage attacks that compose atomic attacks from different layers.
- proposes a method to efficiently select and apply CAPEC attack patterns in order to operationalize attacker's malicious desires into concrete attack actions and identify the most relevant and effective security controls.

## II. A Holistic Attack Analysis Technique

The proposed process for holistic attack analysis is shown in Fig. 1. Our approach takes a three-layer system requirements model and an attacker's malicious intentions as input, and produces a list of security controls that effectively protect the system from possible attacks.

### A. Anti-Goal Modeling

To identify and explore attacks against a system from an attacker's perspective, we capture and model attacker intentions as anti-goals. In particular, we characterize an anti-goal by four attributes (*Asset*, *Threat*, *Target*, and *Interval*) in order to systematically explore alternative attack scenarios. An *Asset* is anything of value to stakeholders. Attackers can benefit from attacking assets. A *Threat* indicates an undesired condition for an asset, which attackers try to achieve to fulfill their malicious intentions. In this work, we leverage the STRIDE threat categories [5] to specify threats. A *Target* is a component of a system, which involves assets and has vulnerabilities that are exploitable by attackers. Within the three-layer system structure, targets vary from layer to layer. An *Interval* represents the time period, during which attackers

[1] https://capec.mitre.org

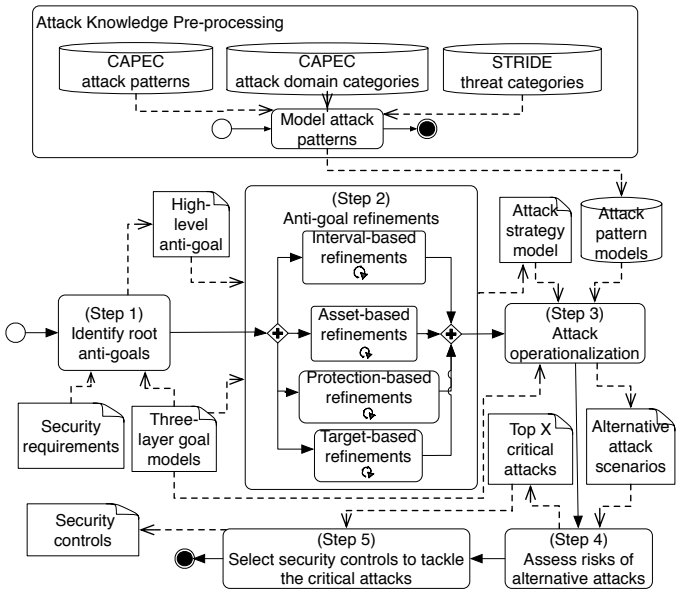


Fig. 1: The overall analysis process

carry out attacks. As shown in Fig. 2, the anti-goal *AG1* means "Tampering *[Threat]* energy consumption data *[Asset]* during the energy collection *[Interval]* by attacking smart meter firmware *[Target]*".

### B. Anti-Goal Refinements

After capturing an attacker's root anti-goal, we proceed to explore attack strategies that shed light on how to achieve the anti-goal by attacking various system components across layers. To this end, we model seven real attack scenarios, which are reported in [7] and [8], in order to understand how attackers generate attack strategies to achieve their anti-goals. Then, we investigate the anti-goal models and propose a set of anti-goal refinement methods. By systematically applying such refinement methods to an attacker's root anti-goal, we generate attack strategies. An example of the application of the asset-based refinement method is shown in Fig. 2: given an anti-goal *AG1* which is intended to harm the asset *Energy consumption data* that consists of two parts, the asset-based refinement method will "or-refine" *AG1* into two sub-anti-goal *AG2* and *AG3*, which are intended to harm the asset *Water consumption data* and *Electricity consumption data*, respectively. Note that the refinement methods that use "and-refinement" allow us to identify multistage attacks.

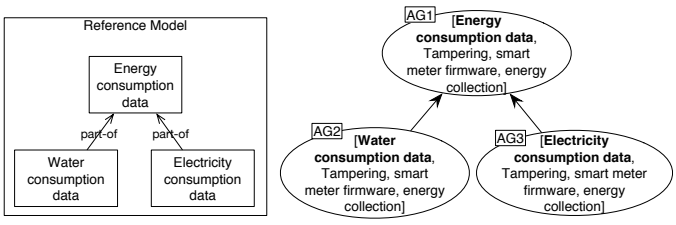


Fig. 2: An example of the asset-based refinement

### C. Anti-Goals Operationalization

Once the attack strategies have been discovered, we scrutinize each leaf anti-goal in the refined anti-goal model to determine whether and how operational attack actions can be carried out to achieve the anti-goal. To this end, we leverage the reusable attack knowledge documented in the CAPEC attack patterns. In particular, if the *Motivation* of an attack pattern matches an anti-goal, and the *Prerequisite* of the attack pattern complies with the context of the system that is targeted by the anti-goal, then the anti-goal can be operationalized into specific attack actions as specified in the attack pattern. As the attack patterns are specified in text, manually matching anti-goals against 463 attack patterns is impossible in practice. In order to efficiently leverage the CAPEC patterns to support our analysis, we have proposed a systematic way of pre-processing the textual attack patterns and modeling them as contextual goal models, which allow us to semi-automatically match and apply attack patterns.

### D. Generate Security Controls

The operationalization of anti-goals can disclose a set of alternative attack scenarios, which should be assessed and prioritized. We make use the knowledge provided by the CAPEC attack patterns, such as the *Typical Severity* and *Typical Likelihood of Exploit* of an attack pattern, to evaluate the risk of alternative attack scenarios, producing a list of the top X critical attacks. According to the results, the identified critical attack scenarios should be treated with corresponding security controls, which are also provided by the CAPEC attack patterns.

## III. Conclusions

In this paper we present ongoing research on a holistic attack analysis technique, which takes an attacker's viewpoint by capturing their malicious intents as anti-goals. The approach considers threats to various system components and the interrelations among those components, and carries out a backwards analysis on the root anti-goal to discover alternative attack scenarios for achieving the anti-goal, and finally provides a list of security controls to effectively protect the system.

**Acknowledgements** This work was supported by ERC advanced grant 267856, titled "Lucretius: Foundations for Software Evolution".

## References

[1] T. Flick and J. Morehouse, *Securing the smart grid: next generation power grid security*. Elsevier, 2010.

[2] S. Barnum and A. Sethi, "Attack patterns as a knowledge resource for building secure software," in *OMG Software Assurance Workshop: Cigital*, 2007.

[3] A. V. Lamsweerde, "Elaborating security requirements by construction of intentional anti-models," in *ICSE*, 2004, pp. 148–157.

[4] G. Sindre and A. L. Opdahl, "Eliciting security requirements with misuse cases," *Requirements Engineering*, vol. 10, no. 1, pp. 34–44, 2005.

[5] A. Shostack, *Threat Modeling: Designing for Security*. John Wiley & Sons, 2014.

[6] T. Li and J. Horkoff, "Dealing with security requirements for socio-technical systems: A holistic approach," in *CAiSE'14*, 2014.

[7] K. D. Mitnick and W. L. Simon, *The art of deception: Controlling the human element of security*. John Wiley & Sons, 2011.

[8] E. Skoudis and T. Liston, *Counter hack reloaded: a step-by-step guide to computer attacks and effective defenses*. Prentice Hall Press, 2005.