Poster: Automating Defence Generation for Risk Assessment

Olga Gadyatskaya
SnT, University of Luxembourg
olga.gadyatskaya@uni.lu

***Abstract*—Efficient risk assessment requires automation of its most tedious tasks: identification of vulnerabilities, attacks that can exploit these vulnerabilities, and countermeasures that can mitigate the attacks. E.g., the attack tree generation by policy invalidation approach looks at systematic automatic generation of attack trees from a socio-technical model of an organization. Attack trees succinctly represent the ways to attack the system. They are useful for identifying the most dangerous attacks, and can be explained to the stakeholders. We now propose a technique to generate attack-defence trees from a socio-technical model. Generated trees incorporate not only attacks, but also defences already present in the system. Furthermore, they can be further used as a basis for risk treatment.**

1. Motivation

Today risk assessment is a critical element of organizational security. Traditionally, it is performed by security analysts (think consultants that charge you per hour) jointly with domain experts by brainstorming on possible threats to important assets. After relevant threat agents and vulnerabilities are identified, the experts analyze unwanted incidents and attack scenarios, and decide on potential impact of those incidents to the assets. It is quite clear that for a large organization this manual risk assessment process is tedious and error-prone (and expensive), due to the huge variety of scenarios and issues to be considered.

Security researchers have started to look at automating the risk assessment process, or at least some of its elements, e.g., identification of vulnerabilities and attack scenarios. The basic approach for automatically extracting a relevant attack model, used, e.g., in [1], is to generate attack tree models from a model that represents the system in question (in our case, the organization). As the risk assessment process focuses simultaneously on physical (buildings and offices), cyber (networks, computers, data) and social (employees and attackers) domains, it is practical to model the organization as a socio-technical system that incorporates elements of all these domains.

However, the risk assessment activity also includes *risk treatment*: identification of countermeasures that need to be introduced in the organization in order to reduce risks to acceptable levels. There is currently a lack of approaches that automatically propose countermeasures based on the socio-technical model of the organization and the set of automatically found attack scenarios.

2. Approach

To address the issue of automating the risk treatment process for an organization, we propose to start by *generating attack-defence trees from the socio-technical model*. Attack-defence trees include both attack and defence nodes, and they can succinctly represent various scenarios of attacker and defender interactions [2]. By generating attack-defence trees we are able to capture the security controls already considered in the model, while generation of attack trees loses this valuable information.

Attack tree generation from a socio-technical model To generate attack-defence trees we follow the approach of *attack tree generation by policy invalidation* [1], which works, roughly, as follows. The system is modelled as a graph with nodes representing locations, actors and assets. The process starts by choosing an asset and an attacker among the entities in the system. The main goal of the attacker is to invalidate the security policy, e.g., confidentiality or integrity policy associated with this asset. Based on the reachability reasoning, the process systematically searches for the ways for the attacker to access the asset. For example, consider the asset to be a secret located in a computer in the manager's office, and the attacker to be an employee working on the same floor. To access the secret, the attacker can try to get to the computer and log into it (an AND-decomposition [2]). This might require possession of the password to the computer that needs to be obtained elsewhere in the system. Alternatively (an OR-decomposition with the previous attack), also the manager knows the secret. Thus, the attacker can access the secret by influencing the manager. This can be implemented, e.g., by gaining trust of the manager. To summarize, the process of attack trees generation by policy invalidation will systematically identify reachable steps, add them to the attack tree, and refine those steps further into the subsequent actions, producing an attack tree complete with respect to the model in the end [1].

Attack-defence tree generation The basic idea to generate attack-defence trees relies on modelling the encountered countermeasures as defence nodes. In the socio-technical model considered, e.g., in [1] the only countermeasures are the enforcement mechanisms for the access control policies that specify which credentials (e.g., password or key) or conditions (e.g., trust relationship) are required for access. Therefore, attack-defence tree generation is started by again choosing an asset and a threat agent among the actors in the system. Then, by applying the reachability reasoning, the algorithm proceeds by identifying all locations

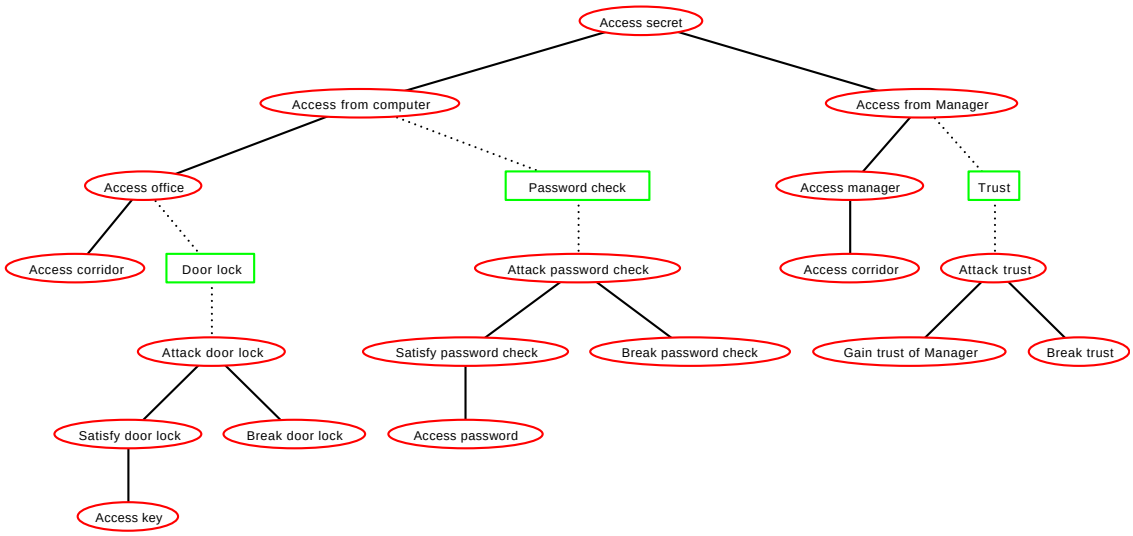

Figure 1: Illustrative example of a generated attack-defence tree (actually generated models are less human-readable; see examples in [1], [3])

in the system from which this asset can be accessed. If any location is protected by an access control policy, the algorithm adds a corresponding defence node. This countermeasure can be attacked by collecting the required credentials or achieving the necessary conditions, or by breaking the enforcement mechanism. Fig.1 gives an illustrative example of a generated tree. More details about the approach are available in [3].

3. Open Problems: Ongoing and Future Work

While we are able to generate attack-defence tree models, the quest of automated defence generation for risk assessment is not completed yet. There are many open problems and challenges ahead.

Security controls not considered in the model. Generated attack-defence trees can only incorporate controls already captured in the socio-technical model. In our own approach, only access control policies in the model yield defence nodes. This is not enough to be a practical risk treatment approach. A large variety of security controls and treatments is known to the security practitioners. While preventive mechanisms (such as encryption or firewalls) can still be captured by access control mechanisms, and thus introduced in our attack-defence trees, detective (e.g., cameras or security guards) or administrative (e.g., security awareness training) controls are currently not supported by socio-technical modelling languages.

Integration with knowledge bases. A practical way to overcome the limitations of socio-technical models is to consider the various knowledge bases developed by security experts in order to support risk assessment (e.g., IT Baseline Protection Catalogs, CAPEC, etc.). These knowledge bases already include mappings of attacks and countermeasures. Therefore, they are complementary to the socio-technical model, and they can be used to extend the generated attack-defence tree with more countermeasures (and attacks).

Integration with the model. Considering an extended attack-defence tree, it will be necessary to map the added defences back to the socio-technical model (and to the organization itself). The challenge here is that not all attacks and defences can be represented in the socio-technical modelling language, as we noted before. Thus, practically, the extended attack-defence tree will be nominated an attack-defence model to be maintained in parallel with the socio-technical model, and traceability links will need to be designed [3].

Matching with manually designed attack-defence trees. Last, but not least, generated attack trees and attack-defence trees do not conform with the basic intuition of traditional attack trees designed by human experts. It is currently challenging for human analysts to comprehend a large automatically generated attack tree or attack-defence tree, because their labels are based on the notation of the modelling language (not the natural language). At the same time, manually designed attack-defence trees have a different structure from the automatically generated ones, because the notion of *refinement* is different in these cases. We will need to bridge this gap by, for example, transforming generated trees into human-readable ones (e.g., by applying compression and aggregation of nodes).

References

[1] M. G. Ivanova, C. W. Probst, R. R. Hansen, and F. Kammuller, "Transforming graphical system models to graphical attack models," in *Proc. of GraMSec 2015*, ser. LNCS, vol. 9390. Springer, 2016.

[2] B. Kordy, S. Mauw, S. Radomirovic, and P. Schweitzer, "Attack-defense trees," *Oxford Univ. Press J. Logic and Computation*, vol. 24, no. 1, pp. 55–87, 2014.

[3] O. Gadyatskaya, "How to generate security cameras: Towards defence generation for socio-technical systems," in *Proc. of GraMSec 2015*, ser. LNCS, vol. 9390. Springer, 2016.

Acknowledgment. This work was partially supported by the European Union FP7 Programme under grant agreement ICT-318003 (TREsPASS).