TRE$_S$PASS: Plug-and-Play Attacker Profiles for Security Risk Analysis

Wolter Pieters*†, Dina Hadžiosmanović*, Aleksandr Lenin‡, Lorena Montoya†, and Jan Willemson‡
*Delft University of Technology, The Netherlands, {w.pieters, d.hadziosmanovic}@tudelft.nl
†University of Twente, The Netherlands, a.l.montoya@utwente.nl
‡Cybernetica AS, Estonia, {aleksandr.lenin,janwil}@cyber.ee

***Abstract*—Existing methods for security risk analysis typically estimate time, cost, or likelihood of success of attack steps. When the threat environment changes, such values have to be updated as well. However, the estimated values reflect both system properties and attacker properties: the time required for an attack step depends on attacker skill as well as the strength of a particular system component. In the TRE$_S$PASS project, we propose the separation of attacker and system properties. By doing so, we enable "plug-and-play" attacker profiles: profiles of adversaries that are independent of system properties, and thus can be re-used in the same or different organisation to compare risk in case of different attacker profiles. We demonstrate its application in the framework of attack trees, as well as our new concept of attack navigators.**

I. Introduction

Providing meaningful metrics for operational security risk is hard [1]. There are different reasons for this: ever-changing threat landscape, difficulties in validation and updating of risk estimations, inability of performing comprehensive quantification of the threat environment, etc. We believe that the difficulty of dealing with these challenges, among other reasons, is due to properties of the *system* and properties of the *threat* having not been separated in common risk analyses in organisations.

Existing approaches for security risk analysis use estimates of time and cost to evaluate attack steps. For example, it is said that a particular attack step costs $ 10,000 or has a 0.2 likelihood of success. Such annotations can then be used for calculating the properties of complex, multi-step attacks from the values associated with the individual steps (e.g., in the framework of attack trees [2], [3], [4], [5]). For example, if access to sensitive data first requires cracking a password and then exploiting a vulnerability, and both have 0.2 likelihood of success, the likelihood of success of the overall attack is 0.04 (assuming independence and single attempts).

The problem arises when the threat and/or vulnerability landscape changes, which is a realistic scenario in dynamic organisations. The threat environment can change due to agent behaviour (e.g., increase in attacker resources), while the vulnerability landscape can change due to infrastructure updates (e.g., applying patches to decrease system vulnerability, but also unintentional events). In either case, the estimated annotations need to be updated. However, these values reflect jointly both system and agent properties: the time required for an attack step depends on attacker skill as well as difficulty of the step. By using a joint estimation, it is unclear how to update the values if only one of the components changes. For example, one may have assigned a 0.2 likelihood of success assuming a script kiddie as attacker. However, how should this value be updated if one faces a national security agency instead and the system has been patched in the meantime?

In the context of atomic threat events, some standards already acknowledge the distinction between attacker and system properties. For example, the FAIR risk taxonomy [6] distinguishes between Threat Capability and Control Strength to determine likelihood of success. However, FAIR does not consider changing conditions in environments with multi-step attacks. In earlier work, we proposed the use of Item Response Theory to take both attacker and system properties into account in quantitative penetration testing [7], [8].

In the ongoing TRE$_S$PASS project (www.trespass-project.eu), we primarily focus on the risk analysis perspective. In particular, we tackle the challenge of operational security risk metrics by analysing multi-step attacks in the context of complex socio-technical systems. We see our work as a step forward in dealing with challenges of security metrics.

II. Attack navigators

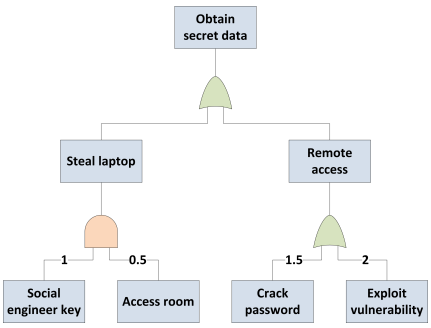
The separation of attacker and system properties enables running risk calculations for different combinations of attacker profiles and system configurations, resulting in comprehensive risk analyses that are further evaluated by the organisation. Thus, we enable "plug-and-play" attacker profiles: profiles of adversaries that are independent of system properties, implying that (1) attacker profiles can be used for different systems, and (2) the risk analysis of a system can be done with a different attacker profile without the need for updates of time, cost or likelihood values. To demonstrate this approach, the TRE$_S$PASS project developed the concept of *attack navigator*, which consists of a *map* of the system components and properties (e.g., by using socio-technical annotations such as system configuration, user policies, network access controls), and an attacker profile which is traversing the *map*. Our tool simulates situations in which different attackers may have different goals, skills and resources, and may therefore prefer different attack paths on the map. Combinations of attacker profiles may be used to reflect the threat environment of a system, and these can be updated when needed. In such a case, a new picture emerges of the risk situation of the organisation based on the new threat environment.

To illustrate the approach, we show how attacker profiles reflect on calculations of the likelihood of success of attacks. For this we use attack trees, an industry standard for adversarial

analysis. Our attack navigator tool is able to generate attack trees from maps of the system, based on a chosen target asset. We focus on the situation where attacker properties (e.g., skill) and system properties (e.g., difficulty) together determine the likelihood of success of an attack step. Different functions are possible to denote such a relation:

- a constraint-based approach, indicating that the attacker should have a skill level at least as high as the difficulty (as an extension of [2]);
- a logistic function, indicating that the likelihood of success is 0.5 when skill (β) and difficulty (δ) are equal (as in Item Response Theory [7]): in its simplest form $P = (e^{\beta-\delta})/(1+e^{\beta-\delta})$.

In attack trees, the system properties (e.g., difficulty) will be annotations on the leaves in the tree. The attacker properties (e.g., skill) will be included in the attacker profiles. When a particular attacker profile is selected, the likelihood of success can be determined for each node based on the combination of difficulty and skill. The resulting likelihood of success can then be used in traditional attack tree calculations, as well as in security risk analysis. Fig. 1 shows in a simplified scenario how to calculate risk properties for an attacker with skill 1. To adapt to changing environments, the same calculations can be performed with different skill levels.



Attacker skill (β): 1
Constraint-based: only left branch (AND-node) is feasible.
Logistic model: the likelihoods of success of the leaf nodes are respectively 0.5; 0.62; 0.38; 0.27. The all-in likelihood of success for the left branch is 0.5 * 0.62 = 0.31. For this attacker, cracking the password would be the most attractive option (i.e., 0.38).

Fig. 1. Example plug-and-play attack tree analysis. The bottom left node is an AND-node; others are OR-nodes. The attack tree is annotated with difficulty (δ) of the steps.

In the project, we work with more complex (multi-parameter) infrastructure maps and attacker profiles. One way to further extend the analysis is to use a three-parameter function to take the dependency between invested time and success into account. One can then derive likelihood of success from (1) attacker skill, (2) step difficulty, and (3) time invested by the attacker. We are currently working on such extensions, based on timed probability distributions [9].

III. Plug-and-Play in TRE$_S$PASS

One of the main bottlenecks of practical security risk analysis is the unclear attribution of properties to the threat environment (attackers) or the system being analysed. In this work, we have presented a way forward based on results from the TRE$_S$PASS project. In particular, we analyse how attacker profiles can be used as plug-and-play components in the risk analysis. In the broader context of the project, we envision that different properties can be used as plug-and-play components: user profiles, system configurations, etc. For example, user profiles could be used for evaluating the likelihood of success of social engineering steps of the attacks, which could differ depending on the cultural environment.

To derive meaningful maps of complex socio-technical systems, we leverage different techniques. Next to scalable formal modelling techniques, several visualisation techniques have been developed in TRE$_S$PASS to support model development and analysis. This ranges from specific representations of importance of branches in attack trees to physical system maps built by stakeholders using *Lego*. Such "thinking tools" are essential in capturing the relevant knowledge about system architecture, potential attackers, and associated parameters.

Acknowledgment

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement ICT-318003 (TRE$_S$PASS). This publication reflects only the authors' views and the Union is not liable for any use that may be made of the information contained herein.

References

[1] V. Verendel, "Quantified security is a weak hypothesis: A critical survey of results and assumptions," in *Proceedings of the 2009 New Security Paradigms Workshop*. New York, NY, USA: ACM, 2009, pp. 37–50. [Online]. Available: http://doi.acm.org/10.1145/1719030.1719036

[2] A. Buldas and A. Lenin, "New efficient utility upper bounds for the fully adaptive model of attack trees," in *Decision and Game Theory for Security*, ser. LNCS, S. K. Das, C. Nita-Rotaru, and M. Kantarcioglu, Eds., vol. 8252. Springer, 2013, pp. 192–205. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-02786-9_12

[3] B. Kordy, P. Kordy, S. Mauw, and P. Schweitzer, "ADTool: Security analysis with attack-defense trees," in *Quantitative Evaluation of Systems*, ser. LNCS, K. Joshi, M. Siegle, M. Stoelinga, and P. D'Argenio, Eds. Springer, 2013, vol. 8054, pp. 173–176. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40196-1_15

[4] S. Mauw and M. Oostdijk, "Foundations of attack trees," in *Proc. 8th Annual International Conference on Information Security and Cryptology, ICISC'05*, ser. LNCS, D. Won and S. Kim, Eds., vol. 3935. Springer, 2006, pp. 186–198. [Online]. Available: http://www.icisc.org/

[5] B. Schneier, "Attack trees: Modeling security threats," *Dr. Dobb's journal*, vol. 24, no. 12, pp. 21–29, 1999.

[6] The Open Group, "Risk taxonomy," The Open Group, Tech. Rep. C081, 2009. [Online]. Available: www.opengroup.org/pubs/catalog/c081.htm

[7] W. Pieters, S. H. G. Van der Ven, and C. W. Probst, "A move in the security measurement stalemate: Elo-style ratings to quantify vulnerability," in *Proceedings of the 2012 workshop on New security paradigms*, ser. NSPW '12. New York, NY, USA: ACM, 2012, pp. 1–14. [Online]. Available: http://doi.acm.org/10.1145/2413296.2413298

[8] F. Arnold, W. Pieters, and M. Stoelinga, "Quantitative penetration testing with item response theory," in *Proceedings of Information Assurance and Security (IAS) 2013*. IEEE, 2013.

[9] F. Arnold, A. Belinfante, F. Berg, D. Guck, and M. Stoelinga, "Dftcalc: A tool for efficient fault tree analysis," in *Computer Safety, Reliability, and Security*, ser. LNCS, F. Bitsch, J. Guiochet, and M. Kaâniche, Eds. Springer, 2013, vol. 8153, pp. 293–301. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40793-2_27