Quantifying the Odds in Real World Attack Scenarios

Paul Tavolato
Faculty of Computer Science
University of Vienna
Vienna, Austria
paul.tavolato@univie.ac.at

Robert Luh
Department of Computer Science
University of Applied Sciences
St. Pölten, Austria
robert.luh@fhstp.ac.at

Sebastian Eresheim
Faculty of Computer Science
University of Vienna
Vienna, Austria
sebastian.eresheim@univie.ac.at

Simon Gmeiner
Faculty of Computer Science
University of Vienna
Vienna, Austria
simon.gmeiner@univie.ac.at

Sebastian Schrittwieser
Faculty of Computer Science
University of Vienna
Vienna, Austria
sebastian.schrittwieser@univie.ac.at

2024 IEEE International Conference on Cyber Security and Resilience (CSR) | 979-8-3503-7536-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/CSR61664.2024.10679461

***Abstract*—In cyber security an important part of risk analysis for IT systems is threat analysis. Threat analysis is an indispensable prerequisite for the planning and budgeting of efficient defense measures. This paper describes a strictly formal method for modeling realistic cyber-attack scenarios. These scenarios are modeled as Discrete Time Markov Decision Processes, opening the opportunity for the application of formal methods to calculate quantitative success probabilities of cyber threats depending on the attacker's skill level and the victim's infrastructure and defense measures. Techniques and tools of probabilistic model checking are applied to find quantitative answers to important questions relevant in threat analysis, such as the attacker's minimum and maximum success probabilities in the victim's IT environment. This provides valuable decision support for security managers when they are forced to assess different security measures in the course of deciding which measure to implement under given budget constraints. To guarantee the practical relevance of the method, a list of 159 attack actions and a list of 118 defense actions are compiled, where the information is gained from several proven tactical and technical knowledge bases. An example – stealing confidential data – shows the application of the method. For calculating probabilities, the model checking tool PRISM is used.**

Keywords:* *Security Management, Threat Analysis, Formal Methods, Model Checking, Formal Security Models

I. Introduction

Today every enterprise, be it a small company or a large organization, is confronted with an ever-growing number of cyber threats that become more and more sophisticated. No matter whether an attacker wants to steal valuable data, hold the enterprise for ransom by encrypting data, deface an organization's website for reasons of hacktivism, or interrupt the enterprise's smooth flow of operation for whatever reason, providing effective cyber security measures has become a must. Planning such measures requires risk assessment, which goes hand in hand with threat analysis – the systemic enumeration and evaluation of the various threats the IT systems and IT processes are facing. A thorough analysis of possible threats is necessary to plan effective countermeasures and thus minimize the risk of falling victim to a devastating cyber-attack [1].

To know about the effects of various possible investments in cyber security as precisely as possible, at best in a quantitative way, would be a great benefit in the decision process of planning a cyber security strategy. Such a process consists of the definition of possible attack scenarios (threats) and the planning of defense measures. Usually the budget for the implementation of security measures is limited, thus making it necessary to select the most effective measures for the given situation. A method that is able to calculate the effects of defense measures on the success probabilities of potential attack scenarios will provide a valuable decision basis.

Various methods of threat analysis have been proposed in the literature throughout the last years. Typical examples are STRIDE, Pasta, and others; see e.g. [2,3,4]. The main problem with these approaches is that they are of an informal or at most semi-formal nature. Hence, they are not suited for precise mathematical analysis and the application of formal methods. By formal methods we generally understand the modeling of systems with mathematically rigorous techniques as a basis for further analysis. In this paper we show how to model attack scenarios formally as Discrete Time Markov Decision Processes and how to formulate statements or questions of interest about the system, usually by means of a temporal logic. (For example: What is the attacker's success probability in a certain situation). These important questions for threat analysis can then be answered by using methods of probabilistic model checking [5] with the help of available tools.

It will be a great advantage for threat analysis if it contains an analysis of the probability that a certain threat – a certain attack scenario – will be successful depending on the current state of defense measures in effect in the victim infrastructure and organization. This will point to the most endangered parts of the IT system and can help to plan the budget for additional defense measures: varying these defense measures in the analysis can quantify the effects an investment in specific measures will have, and thus will support threat analysis (and subsequently risk analysis) significantly. In this paper we describe a method for this purpose that is very close to practice

and is nevertheless viable in typical situations faced by security managers.

To achieve this goal the method must meet two challenges: a) the definition of a formal model of possible attack scenarios and b) a quantitative way to calculate the success probability of attacks. The first challenge is met by constructing attack scenarios based on a most complete list of elementary attack actions and formalizing the scenarios with the help of Discrete Time Markov Decision Processes (DTMDP). The second challenge is overcome by using probabilistic model checking. In detail the following subtasks have been solved:

- A list of elementary attack actions is compiled.
- Each elementary attack action is attributed with the skill level necessary for its operation, its success probability (depending on the current state of defense measures in place), and the damage caused by a successful execution.
- Examples of possible attack scenarios are defined combining the elementary attack actions mentioned above to concerted attacks.
- To define the security level of the victim system, a list of elementary defense actions is compiled.
- The effects of each defense action on the success probabilities of correlated attack actions are defined.
- These attack scenarios are modelled formally by means of a Discrete Time Markov Decision Process.
- Methods of probabilistic model checking are used to calculate the overall success probability of each scenario.

The main contribution of this paper is the application of strictly formal, nevertheless practicable, methods – Markov Decision Processes and probabilistic model checking – within a realistic environment of today's threat landscape. Such a method will constitute a great support for security managers giving them reliable quantitative information about the effectiveness of planned defense measures.

II. Related Work

Threat analysis in general is discussed in various papers from the last 30 years; a recent overview covering all aspects of it is given in [8] in form of a comprehensive self-assessment test; other general descriptions can be found in [9, 10, 11].

A well-known concept for attack modeling are attack trees (AT), which were originally introduced by Schneier [12], further explained by him in [13] and described in more detail in [14] and in [15]. In [6] and especially in [16] attack trees are tackled in a more formal way. Further treatment of attack trees expands the idea to attack-defense trees (ADT) by inserting defense actions, too, to counteract the attack actions [17]. The nodes of attack trees (and ADTs) can be attributed with various parameters in order to compute attack metrics; such parameters could represent e.g. the cost of the attack actions, the time needed for the action, or the probability of the action. In [18] some algorithms for the analysis of different variations of attack trees are described. Aslanyan et al. [19] describe the analysis and metric computation of such quantified properties of attack-defense trees. Quantitative analysis of attack trees or attack-defense trees and the combination of multiple parameters are described in [20, 21, 22, 23, 24, 25] and [27]. A stochastic analysis of attack trees is discussed in [26]; and [27] uses constraint programming to analyze attack trees with incomplete information.

One possibility for computing overall characteristics of ATs and ADTs is the use of formal methods. Generally, the concept of "formal methods" is understood as a process that uses a formal model of a situation and a formal definition of some properties (usually formulated as a proposition in temporal logic) to prove the properties by enumerating all possible paths through the model and checking its validity in each state; this attempt to prove the proposition is called model checking; see [5] for a thorough description. An overview of the use of formal methods in attack tree analysis is given in Widel et al. [7]. In papers by Gadyatskaya et al. [28] and by Kumar et al. [29] Priced Timed Automata are used as formal model for attack trees. In [37], [38] and [39] formal methods together with model checking are used for threat modeling in connection with the development of secure software.

In case the basic formal model contains probabilities (with regard to the success of actions), the method is called probabilistic model checking. In this case the model checker does not output a simple true/false on the proposition checked, but rather minimum and maximum probabilities for the proposition to be true. Relevant descriptions of probabilistic model checking can be found in Kwiatkowska et al. [41, 42] and in [43]; a model checking approach integrating the costs of an attack is described in [16]. The suitability of Discrete Time Markov Processes as a model for ATs and ADTs in connection with probabilistic model checking is obvious and is described in [19] and [36]. In [34] and [35] Markov Processes are combined with game theory. In [36], an algorithm to translate an attack tree into a Markov Decision Process is given, which is then analyzed by model checking techniques; however, the example given is very simple, hence having only limited practical relevance.

Exhaustive sets of attack actions can be found in several documents issued by cyber security organizations and companies such as CISA or MITRE. Elementary attack actions must then be combined to build concerted attacks with specified goals, usually aiming to break one of the three key concepts of cyber security: confidentiality, integrity, availability. Attack scenarios include kill chains. Descriptions and discussions of attack scenarios and kill chains and can be found among others in [30], in the CISA report "Cybersecurity Scenarios" [31], in [32], and in [33].

Unlike the applications described in the literature this paper uses formal methods and probabilistic model checking for threat analysis in realistic settings.

III. Attack Definition

A. Attack Actions

By the term "attack action" we will mean an elementary action that an attacker can execute against the infrastructure or the organization of the victim. The first step in attack definition is the compilation of a list, as complete as possible, of elementary attack actions. This list should comprise actions covering diverse attacks such as malware-based attacks (ransomware, trojans, etc.), phishing attacks (spear phishing,

whaling, etc.), man-in-the-middle attacks, denial of service attacks (DOS and DDoS), SQL injection attacks, DNS tunneling attacks, password attacks, drive-by download attacks, cross-site scripting (XSS) attacks, DNS spoofing or poisoning attacks, session hijacking, URL manipulation, and others. There is quite a number of problems when attempting to compile such a list: there is no general standard of nomenclature and the delimitation of the actions from each other is not trivial as they might be on different abstraction levels. To overcome these problems and in order to keep the approach as close to practice as possible several proven and generally accepted data sources were consulted, and an integrated list of attack actions was synthesized. The data sources involved where: STIX (Structured Threat Information eXpression language) [44], the APT kill chain by Hutchinson [30], the CAPEC (Common Attack Pattern Enumeration and Classification) attack patterns [45], and the MITRE ATT&CK (Adversarial Tactics, Techniques & Common Knowledge) attack and mitigation patterns [46].

The result is a consolidated list of 159 attack actions that are on a similar abstraction level and well distinct from each other. To have a better structure the attack actions can be grouped into categories according to attack phases. In the literature several such attack phases can be found: Lockheed Martin proposes 7 phases: Reconnaissance, Weaponization, Delivery, Exploitation, Installation, Command and Control, Actions on Objectives [48]. MITRE defines 14 so called tactics [49]: Reconnaissance, Resource Development, Initial Access, Execution, Persistence, Privilege Escalation, Defense Evasion, Credential Access, Discovery, Lateral Movement, Collection, Command and Control, Exfiltration, Impact. CISA reduces this in their risk and vulnerability assessment to 11 [50]: Initial Access, Execution, Persistence, Privilege Escalation, Defense Evasion, Credential Access, Discovery, Lateral Movement. Collection, Command and Control, Exfiltration. Such a distribution on many groups, however, leads to the disadvantage of attributions of one action to multiple groups. To make attribution of actions unique, we decided to restrain to 3 phases only: Reconnaissance, Initial Access and Execution.

Reconnaissance comprises all actions related to gaining information about the victim, such as the details of its configuration including the defense measures currently in place or the existence of vulnerabilities. There are 17 such actions in the list. Initial Access contains 31 actions that can be used for getting access to the victim's system by various means. Execution is the largest class with 111 actions, all capable of executing some payload on the victim's system. A list of all 159 actions can be downloaded from https://www.pen.quest/wp-content/uploads/2024/07/Actionlists.pdf

Beside a description of the action in plain text, three values must be attributed to each attack action:

1) The necessary skill level the attacker must have to be capable of conducting the action.

2) The success probability of the action (depending on the defense level of the victim).

3) The damage a successful completion of the action will cause.

Ad 1) The level of an actor's technical abilities constrains action use; a higher skill rating unlocks more complex attacks. A hacker with skill rating 3 can attack vulnerabilities. However, they are not able to perform supply chain attacks or develop zero-day attacks. Skill levels are ranked from 1 to 4; the relationship between a certain skill level and the actions available for an attacker at this level have been derived as a combination of the level of privileges required as defined by MITRE ATT&CK (and some information taken from CVSS base scores).

Ad 2) The definition of success probabilities is a complex task. They largely depend on the amount and quality of defense measures in place at the victim's infrastructure and organization. The proposed method works with a baseline set of success probabilities that assumes only a low basic security level at the victim's site. The main information for these baseline success probabilities of attack actions was taken from [50], where CISA reports such probabilities computed from risk and vulnerability assessments on a yearly basis. Other sources used to this end were sources compiled by MITRE: STIX—Structured Threat Information Expression | STIX Project Documentation [44], CAPEC—Common Attack Pattern Enumeration and Classification [45] and ATT&CK [46]. Furthermore, an impact matrix (attack actions x defense actions) is set up that for each implemented defense action defining the impact it has on the success probability of attack actions (no impact – small impact – medium impact – large impact – full annulation).

Ad 3) Damage is categorized in three dimensions: Confidentiality, Integrity, and Availability. For each dimension the damage level is defined as a number between 0 (no damage) and three (full compromise). The damage attribute of an elementary action is defined as the amount of increase of the damage level in each dimension (plus 1, 2 or 3); additionally, certain defensive actions may decrease a damage level (healing). Changes in damage levels are part of the effects of an action.

B. Attack Scenarios

An attack scenario describes a concerted combination of various elementary attack actions an attacker could perform to achieve her/his predefined goal. Attack scenarios can of course comprise multiple ways of pursuing the attacker's goal: the scenario must contain alternative paths to the goal, such that when one special sequence of attack actions does not work, others are possible, too, and might be tried. In case an action fails, it is possible for the attacker to repeat this action.

The definition of the "success" of an attack scenario depends on the point of view: for the attacker the scenario is successful, when s/he fully reaches the predefined goal of the attack; this means to raise the damage level in the intended dimension to 3. From the point of view of the defender, however, not the attacker's success, but the impact triggered by the attack scenario is relevant; where impact is defined as "the magnitude of harm that can be expected to result from the consequences of unauthorized disclosure of information, unauthorized modification of information, unauthorized destruction of information, or loss of information or information system availability" (according to the definition of NIST [51]). A defender may suffer a considerable impact, even if the attacker does not reach the goal; this means that raising the damage level in any of the three dimensions by at least one point is relevant for the defender.

An attack scenario is predetermined by five parameters – the configuration of the defender, the defense measures in place at the defender's site, the skill level of the attacker, the attacker's intended goal (Confidentiality, Integrity, Availability) and the attacker's initiative; the last term defines the number of steps, after which an attacker gives up if s/he hasn't achieved the goal yet, maybe due to time or budget restrictions or just because the attacker's options are exhausted (and s/he does not want to repeat unsuccessful actions over and over again). The attacker's skill level reduces the available actions; the same holds for the defender's configuration – not all actions make sense in a specific configuration (e.g. when no mobile phones are involved in a configuration the respective actions can be neglected; another condition limiting the choice of actions is their necessary sequence: only if reconnaissance actions are successful, actions from the next phase (initial access) are available; similarly for the initial access phase and the execution phase. Given these parameters, possible combinations of attack actions are defined that would reach the defined goal when executed successfully. To this end we can resort to cyber kill chains discussed in the literature [30] and adapt them with regard to the attack actions compiled for our purpose. (Some of the attack actions from the literature have only supportive character and make sense only when used together with a "real" attack action.)

C. Formalizing Attack Scenarios

To model possible attacker behavior in a specific scenario we use Discrete Time Markov Decision Processes (DTMDP). A scenario is formalized as a 6-tuple (S, A_{att}, AL, P, E, C) where:

$S = \{s_0, s_1, \dots s_n\}$	is a finite set of states; s_0 is defined as the starting state.
$A_{att} = \{a_1, a_2, \dots a_n\}$	is a finite set of available attacker actions.
AL: S x S → A_{att}	is a function labeling a subset of state transitions with an action
P: S x A_{att} x S → [0..1]	P(s,a,s') is the probability that action a executed at the transition from state s will lead to state s'.
$E = \{ e_0, e_1, \dots e_n\}$	is a finite set of effects of an action
C: S x S → P(E)	is a function defining the effects of a transition from state s to state s'.

Generally a path through the DTMDP is characterized by alternating transitions: there are transitions labeled with an action followed by transitions characterizing the success or failure of this action. Figure 1 shows 3 action transitions a1, a2 and a3 emanating from state S leading to states T, U and V; from each of these states there are only two possible successor states depending on the success or failure of the respective action, which is determined by the probability function P that defines the success probability of the action, respectively 1-P for a failure. The next possible action is chosen either by moving to the next step in the kill chain in case of success (states W and X) or by going back to state S and trying another action (including the possibility of repeating the failed action). If in a certain state several actions are available, one is chosen non-deterministically.

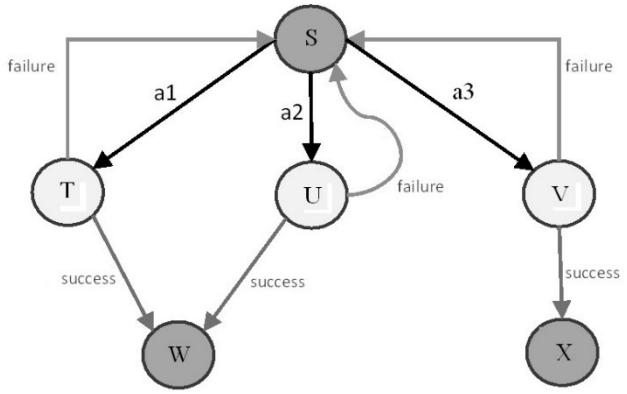

Fig. 1. DTMDP detail with 3 possible actions

Finally, the function C(s,s') defines the consequences of a state transition: state transitions labeled with an action always reduce the attacker's initiative by 1; state transitions representing the success of an action may increase some of the damage levels; state transitions representing the failure of an action have no consequences. The attack ends if either the attacker's ultimate goal is achieved, or her/his initiative is exhausted.

D. Calculating the Success Probability of Attack Scenarios

This model is then implemented in a model checking tool for further analysis. To this end we used the probabilistic model checker PRISM [52]. PRISM provides a language to formulate Markov Decision Processes as well as propositions in a temporal logic. Based on that, it implements probabilistic model checking algorithms to calculate among others minimum and maximum probabilities for the proposition to be evaluated to true. To resolve non-deterministic choices, PRISM uses a uniform scheduler, effectively resolving non-determinism in a uniformly-distributed probabilistic manner.

IV. Defense Measures

As the success probability of an attacker action is dependent on the defense measures in place at the victim's site, we must have a closer look at defense actions. The problems when compiling such a list are similar to those encountered with the list of attack actions: there is no general standard of nomenclature and the actions might be on different abstraction levels. Again, we resorted to proven and generally accepted data sources, mainly MITRE D3FEND [47].

Analogous to the attacker actions a list of 118 possible defense actions is compiled. The actions are subdivided into three categories: Prevention, Detection, and Response. Prevention actions are set in advance to prevent or at least impede certain attack actions. There are 43 prevention actions comprising technical (e.g. one-time passwords, data encryption, limit logon-attempts, run a decoy service, validate input, etc.) as well as organizational (e.g. security awareness training, setup

security response procedures, etc.) measures. The 49 detection actions are aimed at detecting ongoing attacks. Examples are: analyze DNS traffic, detect connection attempts, monitor input devices, detect self-modifications, etc. And the 26 response actions can be used to counter or mitigate attack actions that have already caused some damage; these comprise actions such as disabling accounts, isolating processes, terminating a connection, or shutting down the whole or parts of the system.

Furthermore, the effect of the various defense actions on the success probabilities of selected attack actions is defined as an impact matrix that specifies the reduction of the success probabilities of the attack actions in relation to the defense actions. As an example, let's look at the matrix entries two such prevention actions:

- Limit logon attempts: Limiting logon attempts would have a massive effect on the attacker action "Brute force password". The success probability of the brute force password action without any defense is 0.4. With the defense action limit logon attempts the success probability of brute forcing is decreased to 0.01 (a small success probability remains because of automated limited attempts with time intervals or of a right guess by chance).
- Encrypt data: Encrypting sensitive data on all storage media would render the result of data stealing actions worthless. The success probability of the respective attack actions (steal local data, steal network share data) could be set to a lower value. The baseline values is 0.4 for both actions (as access was already successful) and can be reduced to 0.05 (again we do not set the success probability to 0 because the attacker might be capable of decrypting the data).

V. Example

A. Example setting

Here is a small, nevertheless realistic, example to show how the system works: An attacker wants to steal information from the victim's system. The goal is the violation of the victim's confidentiality dimension. The defender's system is a network of servers and workstations with a network share; it is possible to log into the system via a website; directly accessing the system via an external device (e.g. a mobile phone) is not possible and there is no cloud infrastructure involved. The information the attacker is interested in, is stored in files available as network shares and in local files. We will restrict the attacker's skill level to 2. This has the consequence that only a limited set of actions is available for the attacker. In this scenario in the reconnaissance phase the attacker can gather information about the victim's website and users and/or execute various scans of the victim's system (scan for unknown systems or for vulnerabilities, sending packages and analyzing the response). If successful, the attacker enters the next phase – initial access – and tries to get access to the system either by exploiting a vulnerability detected or, based on the information gathered, hijacks a legitimate account or tries to access the system by using brute force password cracking. Having gained initial access s/he has arrived at the execution phase of the attack and can search for interesting files, either locally or on network shares. If something interesting is found s/he downloads the files. Successfully downloading local files awards one point on the confidentiality scale, downloading files from network shares rewards another 2 points. The attacker is successful if he reaches the value 3 on the confidentiality scale – which means that he was able to download files from local storage and from network shares.

We define the defense level of the victim by the set of success probabilities of the attack actions involved. We assume a baseline defense level with no special defense measures in place within the defender's infrastructure. Later we will vary the set of success probabilities in accordance with planned defense actions to analyze the effect of specific defense actions on the overall success probability of an attack. The baseline defense level is set by information taken from the sources mentioned above. Table 1 shows the baseline of success probabilities of the reduced set of possible attack actions of the scenario (reduced due to the victim's infrastructure and the attacker's skill level).

Another parameter having direct influence on the overall success probability of an attack is the attacker's initiative: How long (in terms of the number of failed or successful attack actions) is the attacker willing to pursue the goal before s/he gives up. This apparently depends – among others – on the attacker's budget.

TABLE I. BASELINE SUCCESS PROBABILITIES OF ATTACK ACTIONS WITHOUT SPECIFIC DEFENSE ACTIONS

Reconnaissance actions	Success probability
Scan system	0,7
Vulnerability scan	0,6
Discovery scan	0,7
Collect user info	0,7
Search victim website	0,7
Hijack external account	0,3
Pretexting	0,7
Initial access actions	
Exploit bug (for access)	0,4
Remote access	0,5
Brute force password	0,4
Execution actions	
Search network shares	0,6
Search local files	0,6
Steal local data	0,4
Steal network share data	0,4

B. Formalization

The DTMDP constructed for the scenario contains one initial state, two final states (attack successful and attack failed), and a set of intermediate states; three states (including the initial state) mark non-deterministic choices of the attacker. State transitions can be of two types: either they represent an attacker action (so-called action transitions) or they represent the outcome of the previous action (success or failure transitions). These transition types occur alternatingly: from a state reached by an action transition, only a success and a failure transition lead to another state.

A success transition leads to the next state of the kill chain, and a failure transition leads back to a state, where either the failed action can be repeated or another action might be chosen

non-deterministically. The choice between success and failure transition is controlled by the success probability p_a of the previous action transition.

The DTMDP has three levels according to the three attack stages Reconnaissance – Initial Access – Execution. A level can be entered only if the preceding level was completed successfully. In the first level – Reconnaissance – the attacker can either scan the victim's system for possibilities to enter (e.g. a vulnerability) or try to hijack an existing account to harvest the necessary information for the next level. If successful, the attacker enters the next level – Initial Access – where s/he must try to access the system based on the information gained during Reconnaissance: by exploiting a vulnerability or by either using collected access information or by brute forcing. If successful the attacker enters the third level – Execution – where s/he tries to steal the desired data.

The function C defines the consequences of a state transition: At transitions labeled with an action the initiative of the attacker is reduced by 1. If the attacker's initiative is exhausted (it reaches a value of 0) and the attacker's goal has not been achieved so far, the DTMDP goes to the final state "attack failed". At certain success transitions the damage level of the attacked system may increase. If the damage level defined as the attacker's goal reaches the value of 3 (that means he has successfully downloaded all files), the attack is successful, and this leads to the final state "attack successful". For failure transitions no consequences are defined.

C. Results and Discussion

Having defined the DTMDP and implemented it in the tool (PRISM) we can now define simple temporal propositions in PRISM's language to calculate the minimum and maximum success probabilities of the attacker.

$$Pmin =? [F state = AttSuccess] \quad (1)$$

$$Pmax =? [F state = AttSuccess] \quad (2)$$

This states that the minimal (1) and the maximal (2) probability P, such that the DTMDP finally (F) ends up in state AttSuccess, should be computed. Table 2 shows the results of the analysis for the example in case of a non-specific baseline defense level of the victim with two different values for the attacker's initiative (20 and 50 indicating that the attacker will lose interest and will quit his activities after executing 20 or 50 attack actions without reaching her/his goal).

TABLE II. MAXIMUM AND MINIMUM SUCCESS PROBABILITIES OF ATTACKS WITH LOW DEFENSE FOR INITIATIVE I = 20 AND I = 50

Low Defense Level	**Attacker Initiative**	
	I = 20	***I = 50***
Min success probability	0,66443	0,99385
Max success probability	0,79653	0,99952

From table 2 one can see that there is a significant difference in the overall success probability of an attack depending on the attacker's initiative. Setting a reasonable value for the initiative is not simple as we usually do not know anything about the attacker's motivation and background. One can predict the attacker's initiative a bit more substantially by considering her/his supposed budget: time invested in an attack is connected with the available budget of an attacker and that can be related to the value the stolen data might have for the attacker.

If the victim now decides to encrypt locally stored data as well as data on network shares, we can calculate the effect of this measure on the overall success probability of the attacker in the defined scenario. To this end we must adjust the success probabilities of the two attacker actions "steal local data" and "steal network share data" from 0.4 to 0.05. Rerunning the model checker with the adjusted success probabilities yields the result in Table 3, again for the two different values of the attacker's initiative (20 and 50).

TABLE III. TABLE 3: MAXIMUM AND MINIMUM SUCCESS PROBABILITIES OF ATTACKS WITH ENCRYPTED DATA FOR INITIATIVE I = 20 AND I = 50

Encrypted data	**Attacker Initiative**	
	I = 20	***I = 50***
Min success probability	0,06239	0, 53483
Max success probability	0, 08090	0, 57682

We see a dramatic reduction in the attacker's success probability. Nevertheless, if the attacker is more motivated and tries harder, he raises his chances.

We could now do the same with another defense measure instead, say limiting the logon attempts to impede brute force password attacks. We reset the success probabilities of the two data stealing actions to 0.4 and reduce the success probability of the brute force password action from 0.4 to 0.01 and then redo the scenario calculation. The results are shown in Table 4.

TABLE IV. MAXIMUM AND MINIMUM SUCCESS PROBABILITIES OF ATTACKS WITH LIMITED LOGON ATTEMPTS FOR INITIATIVE I = 20 AND I = 50

Limited Logon Attempts	**Attacker Initiative**	
	I = 20	***I = 50***
Min success probability	0, 00086347	0, 0038225
Max success probability	0, 79653	0, 99952

Here we see an interesting effect: the difference between minimum and maximum overall success probabilities is much larger than in the other scenario with the encrypted data. While the minimum success probability is very low, the maximum probability did not change at all and has the same value as in the situation without the limitation of logon attempts (see Tables 2). This is due to the fact, that password brute forcing is only one (out of three) possibilities to access the system. So the minimum success probability reflects a situation where the attacker tries to access the system only by using password brute forcing. In this situation the limiting of logon attempts is an effective way to impede the attacker, as is reflected in the minimum probability. But as password brute forcing is not the only way to access the victim' system, as there are other ways to achieve this, too (e.g. exploiting a vulnerability or using a legitimate access), the

maximum success probabilities are the same as in the case of no specific defense in place, because these other ways are not impeded by just limiting the number of false logons. In the alternative scenario, where data encryption was implemented as defense measure, the attacker had no alternative way: stealing (downloading) the data is the only way to achieve the goal.

From the results of the scenario calculations a security manager gets quantified information about the consequences of specific defense measures on the overall success probability of the attacker in this scenario. This will give her/him more insight into the security level of her/his system and will be a valuable decision support for selecting appropriate security controls within a given budget.

If the defender is interested in the impact the attacker's activities inflict on the defender, too, the propositions can be modified accordingly. The states of the model that increment a damage level can be defined as the attacker's "reward". If e.g. the state that increments the confidentiality damage level from 0 to 1 is called ConfDam1, then a proposition to calculate the minimum probability that such damage is inflicted would be:

$$Pmin =? [F state = ConfDam1] \quad (3)$$

Sometimes security managers are faced with the assessment of the impact of sets of defense measures (as opposed to single measures only). The necessary extension of the method for this requirement is straight forward

VI. Conclusion and Future Work

The goal of this project is to support tactical threat analysis by introducing strictly mathematical methods, which can be used to produce quantitative information about the success probabilities of attackers in certain scenarios. This is accomplished by using formal methods, namely DTMDPs and probabilistic model checking. To keep the analysis as close to practice as possible we collected attack actions from accepted data sources and vocabularies combining them to a comprehensive list of 173 different attack actions, subdivided in the phases Reconnaissance, Initial Access, and Execution. These actions can be put together to construct scenarios relevant for the situation analyzed. The scenario is then formally modelled as a DTMDP, which is implemented in a model checking tool, rendering possible a mathematically strict analysis of the success probability of the attacker in this scenario

For analyzing different situations one can vary the defense level of the victim. The defense level is characterized as a set of success probabilities for the attack actions. An implemented defense action decreases the success probabilities of certain attack actions. For different defense levels (with different defense actions in place) maximum and minimum overall success probabilities of the attack can be computed. The results will give valuable hints on the consequences of various security investments. Taking into account the costs of these investments one gets a quantified measure of the relationship between the security costs of certain defense activities and their consequences on the overall security of the system in terms of attacker success probabilities for specific attack scenarios. This yields a quantitative basis for decisions on security investments and will help to make sure to spend the security budget wisely.

The main problems when using this method in a real world setting are the assumptions the security manager must make about the potential unknown attacker: his/her skill level and initiative. If no information about the attacker is available, one can play around with the tool – simulating attackers with different skill level and different initiative. As can be seen from the example above, especially the attacker's initiative (maybe a result of his/her budget) has a relevant impact on the overall success probabilities of an attack. The attacker's skill level on the other hand influences mainly the available attack actions, which will requires more sophisticated defense actions (which depend on the skill level and the budget of the defender).

Future work includes real costs of defense as well as attack actions to bring the scenarios even closer to practice. Furthermore, the definition of a set of predefined typical attack scenarios would greatly facilitate tactic threat analysis for practitioners.

ACKNOWLEDGMENTS

This research was funded in whole, or in part, by the Austrian Science Fund (FWF) P 33656-N. For the purpose of open access, the author has applied a CC BY public copyright license to any Author Accepted Manuscript version arising from this submission.

REFERENCES

[1] James D. McCabe. 2007. Network Analysis, Architecture, and Design. Morgan Kaufmann

[2] Frank Swiderski and Window Snyder. 2004. Threat Modeling. Microsoft Press

[3] Adam Shostack. 2014 Threat Modeling: Designing for Security. Wiley

[4] Izar Tarandach and Matthew J. Coles. 2020. Threat Modeling: A Practical Guide for Development Teams. O'Reilly

[5] Edmund M. Clarke Jr., Orna Grumberg, Daniel Kroening, Doron Peled and Helmut Veith. 2018. Model Checking, second edition. MIT Press

[6] Tarik Guelzim and Mohammad S. Obaidat. 2015. Formal methods of attack modeling and detection. Modeling and Simulation of Computer Networks and Systems - Methodologies and Applications, 841-860. Elsevier. https://doi.org/10.1016/B978-0-12-800887-4.00029-8

[7] Wojciech Wideł, Maxime Audinot, Barbara Fila and Sophie Pinchinat. 2019. Beyond 2014: Formal Methods for Attack Tree-Based Security Modeling. ACM Computing Surveys. 52/4, Article 75. https://doi.org/10.1145/3331524

[8] Gerardus Blokdyk. 2023. Cyber Threat Analysis – A Complete Guide. The Art of Service - Cyber Threat Analysis Publishing

[9] Izzat Alsmadi. 2019. Cyber Threat Analysis. The NICE Cyber Security Framework, 205-242. Springer International Publishing. https://doi.org/10.1007/978-3-031-21651-0_17

[10] James Baine. 2012. An Overview of Threat and Risk Assessment. SANS Institute

[11] Michael Muckin and Scott C. Fitch. 2019. A Threat-Driven Approach to Cyber Security. Lockheed Martin Corporation

[12] Bruce Schneier. 1999. Attack Trees. Dr. Dobbs Journal, Vol. 24/12

[13] Bruce Schneier. 2015. Attack Trees. Secrets and Lies. Wiley Online Books. https://doi.org/10.1002/9781119183631.ch21

[14] Vinet K. Saini, Qiang Duan and Vamsi Paruchuri. 2008. Threat Modeling Using Attack Trees. Journal of Computing Sciences in Colleges, Vol. 23

[15] Sjouke Mauw and Martijn Oostdijk. 2006. Foundations of Attack Trees. Lecture Notes in Computer Science, Vol 3935. Springer Verlag. https://doi.org/10.1007/11734727_17

[16] Zaruhi Aslanyan, Flemming Nielson. 2017. Model Checking Exact Cost for Attack Scenarios. Proc. f the 6th International Conference on Principles of Security and Trust. https://doi.org/10.1007/978-3-662-54455-6_10

[17] Barbara Kordy, Sjouke Mauw, Saša Radomirović and Patrick Schweitzer. 2010. Foundations of Attack–Defense Trees. Formal Aspects of Security and Trust. Lecture Notes in Computer Science, Vol 6561. Springer. doi:10.1007/978-3-642-19751-2_6

[18] Milan Lopuhaä-Zwakenberg, Carlos E. Budde and Mariëlle Stoelinga. 2023. Efficient and Generic Algorithms for Quantitative Attack Tree Analysis. IEEE Transactions on Dependable and Secure Computing 20/5, 4169-4187. doi: 10.1109/TDSC.2022.3215752.

[19] Zaruhi Aslanyan, Flemming Nielson and David Parker. 2016. Quantitative Verification and Synthesis of Attack-Defence Scenarios. IEEE 29th Computer Security Foundations Symposium, CSF. doi: 10.1109/CSF.2016.15

[20] Barbara Fila and Wojciech Wideł. 2019. Efficient attack-defense tree analysis using Pareto attribute domains. In Proc. of the IEEE 32nd Computer Security Foundation Symposium, 200–215. IEEE

[21] Andrea Bobbio, Lavinia Egidi, and Roberta Terruggia. 2013. A methodology for qualitative/quantitative analysis of weighted attack trees. IFAC Proc. Volumes 46/22, 133–138

[22] Barbara Kordy and Wojciech Wideł. 2018. On quantitative analysis of attack–defense trees with repeated labels. Principles of Security and Trust. POST. Lecture Notes in Computer Science, Vol. 10804. Springer. doi.org/10.1007/978-3-319-89722-6_14

[23] Aivo Jürgenson and Jan Willemson. 2008. Computing exact outcomes of multi-parameter attack trees. In Proc. of the OTM Confederated International Conference on Move to Meaningful Internet Systems, 1036–1051

[24] Ahto Buldas, Peeter Laud, Jaan Priisalu, Maert Saarepera, and Jan Willemson. 2006. Rational choice of security measures via multi-parameter attack trees. International Workshop on Critical Information Infrastructures Security, 235–248

[25] Zaruhi Aslanyan and Flemming Nielson. 2015. Pareto efficient solutions of attack-defence trees. Principles of Security and Trust, POST. Lecture Notes in Computer Science, Vol. 9036. Springer. doi.org/10.1007/978-3-662-46666-7_6

[26] Nihal Pekergin, Sovanna Tan and Jean-Michel Fourneau. 2016. Quantitative Attack Tree Analysis: Stochastic Bounds and Numerical Analysis. International Workshop on Graphical Models for Security, GraMSec. doi:10.1007/978-3-319-46263-9_8

[27] Ahto Buldas, Olga Gadyatskaya, Alexandr Lenin, Sjouke Mauw and Rolando Trujillo-Rasua. 2020. Attribute Evaluation on Attack Trees with Incomplete Information. Computers and Security 88/101630. doi:10.1016/j.cose.2019.101630

[28] Olga Gadyatskaya, René R. Hansen, Kim G. Larsen, Axel Legay, Mads C. Olesen and Danny B. Poulsen. 2016. Modelling Attack-defense Trees Using Timed Automata. FORMATS 2016. Lecture Notes in Computer Science, Vol. 9884. doi: 10.1007/978-3-319-44878-7 3

[29] Rajesh Kumar, Enno Ruijters and Mariëlle Stoelinga. 2015. Quantitative Attack Tree Analysis via Priced Timed Automata. In Proc. of the 13th International Conference on Formal Modeling and Analysis of Timed Systems, FORMATS. doi: 10.1007/978-3-319-22975-1 11

[30] Eric M. Hutchins, Michael J. Cloppert and Rohan M. Amin. 2011. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Lead. Issues Inf. Warf. Secur. Res. 1/80

[31] CISA - Cybersecurity and Infrastructure Security Agency. 2023. Cybersecurity Scenarios. US Government Cybersecurity and Infrastructure Security Agency

[32] Dietmar P. Möller. 2020. Attack Models and Scenarios. Cybersecurity in Digital Transformation: Scope and Applications, Springer International Publishing, 89-98. https://doi.org/10.1007/978-3-030-60570-4_6

[33] Stephan Moskal, Shanchieh J. Yang and Michael E. Kuhl. 2018. Cyber threat assessment via attack scenario simulation using an integrated adversary and network modeling approach. The Journal of Defense Modeling and Simulation, Vol. 15/1, 13-29. doi.org/10.1177/1548512917725408

[34] Praveen Bommannavar, Tansu Alpcan and Nicholas Bambos. 2011. Security risk management via dynamic games with learning. In Proc of the IEEE International Conference on Communications (ICC). IEEE. doi: 10.1109/icc.2011.5963330

[35] Xiaolin Cui, Xiaobin Tan, Yong Zhang and Hongsheng Xi. 2008. A markov game theory-based risk assessment model for network information system. In Proc. of the International Conference on Computer Science and Software Engineering, CSSE. doi: 10.1109/CSSE.2008.949

[36] Saif U. Malik, Adeel Anjum, Syed A. Moqurrab and Gautam Srivastava. 2022. Towards enhanced threat modelling and analysis using a Markov Decision Process. Computer Communications, Vol. 194, 282-291. https://doi.org/10.1016/j.comcom.2022.07.038

[37] Shafiq Hussain, Harry Erwin and Peter Dunne. 2011. Threat modeling using formal methods: A new approach to develop secure web applications. In Proc. of the 7th International Conference on Emerging Technologies, 1-5. doi: 10.1109/ICET.2011.6048492

[38] Quentin Rouland, Brahim Hamid and Jason Jaskolka. 2020. Reusable Formal Models for Threat Specification, Detection, and Treatment. Lecture Notes in Computer Science, Vol 12541. Springer. https://doi.org/10.1007/978-3-030-64694-3_4.

[39] Quentin Rouland, Brahim Hamid and Jason Jaskolka. 2021. Specification, detection, and treatment of STRIDE threats for software components: Modeling, formal methods, and tool support. Journal of System Architecture 117, C. https://doi.org/10.1016/j.sysarc.2021.102073

[40] Abdelhakim Baouya, Samir Ouchani and Saddek Bensalem. 2022. Formal Modelling and Security Analysis of Inter-Operable Systems. Lecture Notes in Computer Science vol 13343. Springer. https://doi.org/10.1007/978-3-031-08530-7_47

[41] Marta Kwiatkowska, Gethin Norman and David Parker. 2007. Stochastic Model Checking. In: Proceedings of the 7th international conference on Formal methods for performance evaluation, 220-270. Springer. doi:10.1007/978-3-540-72522-0_6

[42] Marta Kwiatkowska, Gethin Norman and David Parker 2017. Probabilistic model checking: Advances and applications. Formal System Verification, 73-121. Springer. https://doi.org/10.1007/978-3-319-57685-5_3

[43] Gul Agha and Karl Palmskog. 2018. A Survey of Statistical Model Checking. ACM Transactions on Modeling and Computer Simulation, Vol. 28/1, 1-39. doi:10.1145/3158668

[44] MITRE Corporation. STIX—Structured Threat Information Expression | STIX Project Documentation. Retrieved February 21, 2024 from https://oasis-open.github.io/cti-documentation/

[45] MITRE Corporation. CAPEC—Common Attack Pattern Enumeration and Classification. Retrieved February 21, 2024 from https://capec.mitre.org/

[46] MITRE Corporation. MITRE ATT&CK. Retrieved February 21, 2024 from https://attack.mitre.org/

[47] MITRE Corporation. MITRE D3FEND - A knowledge graph of cybersecurity countermeasures. Retrieved February 21, 2024 from https://d3fend.mitre.org/

[48] Lockheed Martin Corporation. Cyber Kill Chain. Retrieved February 21, 2024 from https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

[49] MITRE Corporation. Enterprise tactics. Retrieved February 21, 2024 from https://attack.mitre.org/tactics/enterprise/

[50] CISA. 2023. FY22 Risk and Vulnerability Assessments (RVA) Results. Retrieved February 21, 2024 from https://www.cisa.gov/news-events/alerts/2023/07/26/cisa-releases-analysis-fy22-risk-and-vulnerability-assessments

[51] NIST. Computer Security Resource Center – Glossary. Retrieved February 21, 2024 from https://csrc.nist.gov/glossary/term/impact

[52] Marta Kwiatkowska, Gethin Norman and David Parker. 2011. PRISM 4.0: Verification of Probabilistic Real-time Systems. In Proc. 23rd International Conference on Computer Aided Verification (CAV'11), Lecture Notes in Computer Science 6806, 585-591, Springer