Computers & Security 132 (2023) 103328

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Game-theoretic APT defense: An experimental study on robotics

Stefan Rass[a,c,*], Sandra König[b], Jasmin Wachter[c], Víctor Mayoral-Vilches[c], Emmanouil Panaousis[d]

[a] Johannes Kepler University Linz, Austria and Universitaet Klagenfurt, Austria
[b] Austrian Institute of Technology, Vienna, Austria
[c] Universitaet Klagenfurt, Austria
[d] University of Greenwich, London, UK

ARTICLE INFO

Article history:
Received 19 December 2022
Revised 8 May 2023
Accepted 7 June 2023
Available online 9 June 2023

Keywords:
Cyber security
Cyber risk
Advanced persistent threats
Cyber physical system
Attack graph
Attack tree
Game-theory
Stealthy intrusion
Attacker-defender games

ABSTRACT

This paper proposes a novel game-theoretic framework for defending against Advanced Persistent Threats (APTs). It applies the original Cut-The-Rope model into an experimental study extending the previously studied attacker movements beyond the Poisson distribution to a realistic set of attack actions. More importantly, it demonstrates the value of this framework on an experimental study of an APT defense game on attack graphs, which lets a security officer establish an optimized defense policy against stealthy intrusions. The security model and algorithm under study is designed for practical use with attack graphs as threat models, possibly including vulnerability information if available. The game-theoretic optimization delivers a proactive defense policy under the following assumptions or requirements: first, we do not need to assume that the system is, or has been, clean from adversaries at any time. At the moment when the defender computes the defense policy, the attacker is assumed to already be in the system (also having penetrated it until an unknown depth). Second, the defender does not rely on any signaling or other indicators of adversarial activity, nor is there a reliable feedback mechanism to tell the defender if its actions were successful or not. Third, the model can use information on exploits, such as Common Vulnerabilities and Exposures (CVE) numbers, to refine the defense game, but can also operate without such information. We corroborate our findings on publicly documented attack graphs from the robotics domain; without and with CVE information. We run experiments against two different types of defense regimes, and compare the results against an intuitive baseline defense heuristic. The results show that the optimized defense strongly outperforms simple heuristics, like taking the shortest or easiest attack paths.

1. Introduction

Contemporary Advanced Persistent Threats (APTs) undergo a sequence of common "phases", which in the simplest instance boils down to three: (i) the initial infection (where the attacker makes the initial contact, e.g., by sending a successful spam or phishing email, usually after a reconnaissance phase of finding out details about the victim system to break in); (ii) a silent phase of penetration and learning (where the attacker gets into the system as deep as it can; often slowly and stealthy to avoid detection); and (iii) a weaponization and damaging phase. Security models can relate to a specific phase or be heterogeneous models spanning multiple phases (Rass and Zhu, 2016). This work is concerned with the daily business of defense, under the assumption that the infection has already happened, but there has not been any damage so far. Thus we are in the "incubation" phase in the APT life-cycle.

A refined view on the evolution of an APT is the *kill chain* (Kamhoua et al., 2018). This consists of *reconnaissance, exploit, command & control, privilege escalation, lateral movement* and *objective/target*, in the sequential order just given. This work proposes a game-theoretic model intended to support the daily business of a Chief Security Officer (CISO), seeking a proactive defense against an invisible intruder and more elaborated attacks. We assume that the adversary is already in the system,[1] and the CISOs duty is pre-

* Corresponding author at: Johannes Kepler University Linz, Austria and Universitaet Klagenfurt, Austria.
E-mail address: stefan.rass@jku.at (S. Rass).

[1] Following the famous quote of Robert S. Mueller: "There are only two types of companies: Those that have been hacked and those that will be hacked."

https://doi.org/10.1016/j.cose.2023.103328

venting damage. The battlefield on which the CISO, hereafter called the defender, matches the stealthy adversary, is an attack graph.

This is a graph-theoretic model of a system, relating individual system components to threats and exploits on them thereby visualizing possible attack paths towards defined target nodes. Such attack graphs can be compiled from threat modeling activities and further refined with penetration tests and the help of topological vulnerability scans (see, e.g., Jajodia et al., 2011). On the attack graph, we play a stealthy intrusion game with infinite time horizon and repetitions, in which the defender takes action to keep the attacker away from the critical assets in the enterprise, while the adversary is on its way towards some target asset, along one or more attack paths. The game instantly terminates if the attacker has reached the critical asset, in which case the defender (permanently) lost.

The interaction between the two players is constrained as follows:

Assumption 1. The attack graph $G = (V, E)$ consists of V nodes and E are the edges (e.g., exploits); see Fig. 2 for an abstract example and the use-case Section 4 for concrete practical instances. We assume a single starting node (for all attacks), and a single target node, representing some critical asset to be captured. The graph is assumed to be acyclic (thus, the attacker will never enter infinite loops along accomplished exploits), and all paths (more precisely the attack paths/vectors) lead to the target asset. We denote this target as $v_0 \in V$ hereafter. The nodes in V represent system threats and vulnerabilities, or system states, while edges represent threats or exploits to get from one component/state into the next component/state. For example, the adversary may jump from a desktop computer v_{pc} to a server v_{server}, or may gain root privileges, denoted as $v_{pc,root}$ from user-level privileges $v_{pc,user}$ on the same computer. In either case, we would have a directed edge $v_{pc} \to v_{server}$, or $v_{pc,user} \to v_{pc,root}$ to express this attack path in the graph model.

Assumption 2. The adversary is already somewhere in the system at an unknown location when the defender enters the gameplay (i.e., we are past the event of infection/reconnaissance). We also assume that the defender has no indication of adversarial activity (e.g. there is no Intrusion Detection System (IDS) or the IDS has not detected the activity); the adversary is *stealthy*. In absence of adversarial signals, the defender may assume all possible locations of the adversary as uniformly distributed (the inclusion of signals is discussed in Section 7.5).

Assumption 3. The adversary may run *parallel* or *concurrent attacks*, thereby exploiting several, up to all, attack paths simultaneously to maximize its chances to conquer v_0.

This general setting was converted into a game-theoretic model in Rass et al. (2019), named Cut-The-Rope: Like on a *chess-board*, the adversary in Cut-The-Rope runs parallel attacks, one on each attack path available. Since the attacker's location on each attack path is unknown to the defender, this player imagines a whole "cohort" of *avatars* starting from all possible locations in the network and moving towards v_0. The strategic choice of the adversary, from the defender's perspective, is about existing attack paths, but the defender does not know where the attacker is, equivalently, how far the adversary has already come down an attack path π. To tackle this uncertainty, the defender plays the game as if the adversary would first (strategically) choose the path π, and move *all* avatars on π simultaneously towards v_0. In other words, it moves avatars at all *possible, not* necessarily also *probable*, locations. The target asset (and security game) is lost (to the attacker), if at least one of the adversary's avatars reaches v_0.

For the adversarial movement, Cut-The-Rope lets the defender assume a random distribution on how many steps an avatar can take when it is on the move. This *random movement pattern* is a model design choice, and the original work (Rass et al., 2019) about Cut-The-Rope assumed one specific movement regime, in which the defender acts periodically, and the attacker has some "aggressiveness level" λ, interpretable as an "expected number exploits per day" or within a defined unit of time. This amounts to a Poisson-distributed number of steps taken along the attack path.

In this paper, we go beyond the first Cut-The-Rope implementation by studying (Section 3) three other patterns of adversarial movement besides the original Poisson model, and evaluate the defense level that a CISO can obtain from using Cut-The-Rope as a method to determine a security policy.

Research questions and contribution

Some companies have CISOs with a fixed working schedule, which corresponds to a periodically active defender (working days, day/night-shifts, etc.). What if there is a 24/7 continuous response team available, like in bigger (globally distributed) companies that run their own security operations center? These (multiple) defenders may become active at random time intervals and at any time. For the security game, it means that the defender will not be active periodically, but rather at random times and possibly at any time. This is the *first new* movement contributed and studied in this work.

Both, the original Poisson movement and the just described continuous security response policy are agnostic of the particular details of exploits, like their difficulty or severity. However, many attack graphs do carry additional information about exploits, and if so, it is desirable to use it in the security model. These additional details can range from a security threat research and risk assessment or scoring like Common Vulnerability Scoring System (CVSS), up to proof of concept implementations for each exploit. The *second new* movement pattern proposed in this work makes the attacker's traversal dependent on exploit complexities (threats assigned a higher complexity would thereby be probabilistically less feasible), as far as they are known, and studies the defense performance against a defender that is again periodically active.

The *third new* movement pattern is a combination of a defender that can take action at any time in a 24/7 continuous security provisioning, against an attacker that has to deal with threats and exploits of different complexity. We stress that none of these cases assumes a purely reactive defense, i.e., we do not study security response patterns when the incident has been noticed. This is due to the assumption of stealthiness of the intrusion; once the attacker becomes visible, it is because the target asset v_0 was lost (permanently).

The other contributions of this work are two case studies:

We first provide an experimental study and illustration of how to use Cut-The-Rope in different settings, based on two documented attack graphs for industrial robots. Specifically, we look at the Modular Articulated Robotic Arm (MARA) and Mobile industrial Robotics MiR100 robots (AcutronicRobotics, 2021; Alias Robotics, 2019; 2020; 2021), for which attack graphs have been compiled by security experts. On these, we instantiate Cut-The-Rope and compute results in the aforementioned settings of a periodically/randomly active defender versus an attacker that traverses an attack path with uniform speed at an average number of exploits per time unit, or mounts attacks with individually distinct exploit complexities, thus being slower or faster, depending on the chosen path.

Experiments are conducted on the MARA robot, for which the threats and exploits are known, but without a CVSS rating or further details. In that case, the original Poisson model from (Rass et al., 2019) and the first of the new patterns announced above are usable. The other case study is on the MiR100 robot,

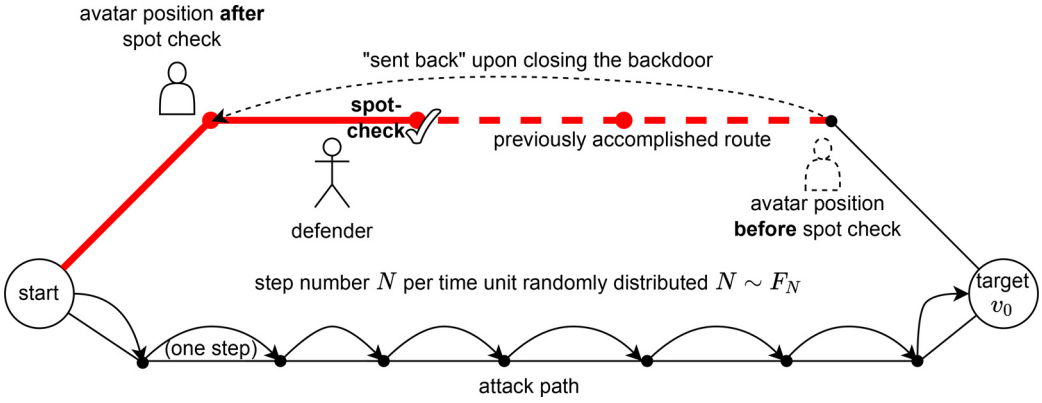


Fig. 1. Basic Gameplay of Cut-The-Rope.

whose attack graph carries additional vulnerability and exploit details, so that the other two new movement patterns, described below in Section 3, become applicable.

2. The model

In the following, we let sets appear as upper case letters, and vectors and matrices in boldface font. Given a finite set X, the symbol $\Delta(X)$ denotes all (categorical) probability distributions supported on X, i.e., an element $\mathbf{x} \in \Delta(X)$ has the elements $(p_1, \ldots, p_{|X|})$ with $p_i = \Pr(x_i \in X \text{ is chosen})$. The symbol $|X|$ is the cardinality of the set X.

We refrain from replicating the full formal description of Cut-The-Rope, and instead summarize its concept in Fig. 1. The game is played entirely from the defender's viewpoint: the defender knows the attack graph $G = (V, E)$ and can enumerate the attack paths on which the adversary can be. For simplicity, we assume that the number of these routes is tractably small. Generally, the number of routes can be exponential in the cardinality $|V|$ of nodes, but by strategic domination and other heuristics, some routes may be safely excluded from consideration. We will revisit this point later in Section 7.4.

To express the uncertainty about where exactly the adversary is, the defender acts as if the attacker would move a whole cohort of avatars towards v_0, each avatar starting from another possible location in the attack graph with uniform probability. The game is round-based, where the exact meaning of a round depends on the moving patterns of the defender and the attacker:

- If the defender acts periodically in fixed intervals (e.g., daily), a round of the game is one period of activity for the defender (e.g., one day). During this period of time, the attacker can take a random (unlimited) number of steps along the attack path towards the goal.
- If the defender is taking action at random, e.g., taking exponentially distributed pause times, then a round of the game is, in each instant, the random idle time of the defender. Again, during these periods, the attacker can take any number of actions, depending on its "configuration" and/or the attack path. In the terminology of the FlipIt game (van Dijk et al., 2013), this is called an *exponential defense strategy*.

In both cases, we do not explicitly model the time to complete a spot-check and merely assume this completion to be possible within one unit of time. Including the defender's costs for spot-checking as a separate goal (to minimize) makes the game multi-criteria and calls for Pareto-optimization, which we leave out of our scope in this work (and up to future considerations). We will come back to the exact meaning of a "round" or "unit of time" in Section 3.1. Let us first complete the description of the gameplay: Fig. 1 displays two attack paths, with the lower path showing the step-by-step traversal of an avatar towards the goal v_0. Every possible action of the defender is here called a *spot-check* at any node in $V \setminus \{v_0\}$, where the target node is excluded to avoid trivialities.[2] A *spot-check* can mean any action that, for example, (i) cleans a component from malware, (ii) disables certain services that an exploit would rely on, (iii) changes in the security policy or implementation that invalidates the adversary's knowledge (e.g., access control mechanisms), or similar. Common to all actions of the defender is their *transient* efficacy, which means that the effect of such an action is *not* permanent (the opposite case is discussed later in Section 7.2). After the action, and not necessarily known to the defender, the attacker is sent back on the attack path to an earlier position (upper part of Fig. 1). For example, if the so-far accomplished route has at some point used access credentials for a computer, and the defender has just changed them, the route is essentially closed at this point, and the attacker has to re-try just before this point[3] The avatars can go unaffected by the defender's action in two cases: (i) if it travels on a different route that the defender did not inspect in this moment (e.g., lower attack path in Fig. 1), or (ii) the attacker started from a location after the cut point (e.g., if the attacker is left to the cut point ✓ in Fig. 1). This assumption implicitly accounts for "out of attack graph" ways of the attacker having reached this location In either case, the avatar's journey is not intercepted.

The attacker may at any point decide to try a different route instead. This is called *lateral movement*. It is naturally included in this modeling by having avatars on all attack routes, which makes lateral movement nothing else than moving other avatars on another route. Cut-The-Rope is played under the assumption that an avatar can be thrown back to an earlier point by the defender, but will in any case re-try its current attack path, until it (or any of its clones) has reached the goal.

The payoffs in the game are zero-sum, and come to the probability of reaching v_0 in a single round of the game. This is the payoff to the attacker, and likewise the loss of the defender, who seeks to minimize this probability. Its computation depends on the probability distribution law that governs how many steps N can be taken during the defender's idle periods. This is the main ingredient whose influence is studied in this work, relative to a heuristic best-practice defense.

[2] If the defender would not move away from the target, there would be nothing to accomplish here for the adversary and there would be nothing to analyze.

[3] We herein assume that there is no direct way to just get back to the later point: if there would be such a shortcut route bypassing the just-closed backdoor, this would be another attack path, taken by a respectively other avatar.

The *payoff* to the adversary is the chances for any of its avatars to reach, from its current position, the target v_0 within N steps and within the time-limit W, during which the defender is idle.[4] This can be fixed (for a periodic defender) or random (for an exponential defense strategy). We collect all avatars in a set $\Theta \subseteq V \setminus \{v_0\}$, and denote individual avatars as $\theta \in \Theta$. The exclusion of v_0 from this set is to avoid the trivial case where the attacker has already reached v_0 before the defense game starts. The payoffs to both players are:

$$u_{\text{attacker}} = -u_{\text{defender}} = \Pr(\text{adversary reaches } v_0) \tag{1}$$

We will analytically determine this quantity in Section 2.2 in expressions (7) and (8), which make the dependency on the strategic choices of the defender and attacker visible and explicit.

2.1. Strategies

We now turn to the description of how the defender's and attacker's action determine the probability to reach v_0. The strategic choices of both players towards maximizing or minimizing Pr(adversary reaches v_0) are the following:

- The defender has a choice from the set $AS_1 := V \setminus \{v_0\}$ to spot-check, giving a total of $n = |AS_1|$ actions. We will write $\mathbf{x} \in \Delta(AS_1)$ for a randomized such spot-checking rule.
- The attacker can likewise use a total of $m = |AS_2|$ attack paths in G, collected in the set AS_2. Each avatar starts from a different location $\theta \in \Theta = V \setminus \{v_0\}$ and traverses one of the (perhaps many) routes from θ towards v_0. The adversary solution in the game is the best choice of attack paths from AS_2. Likewise, we will write $\mathbf{y} \in \Delta(AS_2)$ for a random choice from the set of attack paths.

Every avatar takes action by being moved forward along the attack path that it is on, and draws/samples a random number N from a fixed step-distribution f_N. This is *not* a strategic choice, but rather a part of the game's *payoff mechanism*. Low-level procedures of how the avatar technically mounts exploits are not expressed nor modeled in the game itself (due to the heterogeneity and sheer number of possibilities of exploits in a real-life attack graph).

2.2. Definition of payoffs

For the sake of rigor, let us concretize (1) by showing how it is practically obtained. This will also display the role of the movement patterns (periodic, exponential) in the experimental analysis. Working out the adversary's utility is a matter of conditioning the attack step distribution F_N on the current situation in the network, i.e, the position of the avatar and where the defender took action.

Let $\pi_1, \pi_2, \ldots, \pi_m$ be an (exhaustive) enumeration of all attack paths, each starting from another location $\theta \in \Theta \subseteq V \setminus \{v_0\}$. Each starting location is thus identified with one avatar, and the adversary moves all of them towards v_0. Let m be the total number of all attack paths.

Each such path is again a sequence of nodes, written as $\pi = (\theta, w_1, w_2, \ldots, v_0)$ with all $w_i \in \{v_1, v_2, \ldots\} = V$ and $\theta \in \Theta$ being the starting point of the route, one-to-one corresponding to an adversarial avatar. The set of nodes constituting π is $V(\pi)$. Furthermore, let $d_\pi(u, v) \in \mathbb{N}$ count the edges on the path π from u to v. It is a graph-theoretic distance.

[4] Here, we simplified the payoff representation from a vector-valued distribution in Rass et al. (2019) over the attacker distance to the goal, to the probability of reaching the goal. This does not affect the solution of the model, but facilitates readability.

Then, the *location distribution* for the attacker assigns to each node $v \in V$ the mass

$$\Pr(\text{avatar location} = v | V(\pi)) = \frac{f_N(d_\pi(\theta, v))}{\Pr_N(V(\pi))}, \tag{2}$$

in which $f_N(n) = \Pr(N = n)$, where $N \in \{0, 1, 2, 3, \ldots\}$ is the random number of steps undertaken by the avatar, and

$$\Pr_N(V(\pi)) = \sum_{x \in V(\pi)} \Pr_N(d_\pi(\theta, x)) = \sum_{x \in V(\pi)} f_N(d_\pi(\theta, x)). \tag{3}$$

The probability density f_N will be the main element to vary when describing different attacker-defender scenarios (such as announced in the introduction under the contributions). We will give various options to define f_N in Eqs. (10), (11), (12) and (13).

Now, the defender attempts to break the attacker's chain of exploitation ("cut the rope" in the wording of (Rass et al., 2019)). Let $c \in V$ be the checked node, then the possibly truncated path is

$$\pi|_c = \begin{cases} (\theta, w_1, w_2, \ldots, w_{i-1}), & \text{if } c = w_i \text{ for some } w_i \text{ on } \pi \\ (\theta, w_1, \ldots, v_0), & \text{otherwise.} \end{cases} \tag{4}$$

The closing of a backdoor here becomes a conditioning of the distribution of the avatar's location on the shorter (cut) path $\pi|_c$. The formula is the same as (2), only with π replaced by $\pi|_c$ now. Since $c \sim \mathbf{x}$ follows the defender's mixed spot checking strategy (possibly degenerate), and the set of paths π along which avatars proceed, the defender can determine the possible locations of the attacker, based on the imagined avatars, as the vector of probabilities

$$U = (\Pr(\text{adversary's location} = v | V(\pi|_c)))_{v \in V}, \tag{5}$$

which depends on the random choices of the defender ("where to cut?") and the attacker ("which route to take?"). This is what the implementation of Cut-The-Rope computes.

The actual quantity of interest for the game, coming back to (1), is the mass that U assigns to v_0. This is the utility for the adversary and conversely the loss of the defender. Since the game is, from the attacker's perspective, a strategic choice $\mathbf{y} \in \Delta(AS_2)$ of an attack path, the payoffs in the game are obtained from the following consideration:

$$\begin{aligned} &\Pr(\text{attacker reaches } v_0) = \Pr(\text{at least one avatar reaches } v_0) \\ &= \sum_{\theta \in V \setminus \{v_0\}} \Pr(\text{avatar reaches } v_0 \text{ starting from } \theta) \cdot \Pr(\theta) \\ &= \sum_{c,\pi} \sum_{\theta \in V \setminus \{v_0\}} \underbrace{\Pr[\text{avatar } \theta \text{ has location } v_0 \mid V(\pi|_c)]}_{\text{from eq. (2) and (4)}} \\ &\quad \cdot \underbrace{\Pr(\text{path } \pi \text{ is chosen and defender cuts at } c)}_{\text{strategic choices to optimize}} \cdot \underbrace{\Pr(\theta)}_{=1/|AS_2|} \end{aligned} \tag{6}$$

$$\begin{aligned} &= \sum_{c,\pi} \sum_{\theta \in V \setminus \{v_0\}} \Pr\Big[\text{avatar } \theta \text{ has location} \\ &\quad v_0 \mid V(\pi|_c)\Big] \cdot \Pr_{\mathbf{x}}(c) \Pr_{\mathbf{y}}(\pi) \cdot \Pr(\theta) =: u_{\text{attacker}}(\mathbf{x}, \mathbf{y}) \end{aligned} \tag{7}$$

$$= -u_{\text{defender}}(\mathbf{x}, \mathbf{y}) \tag{8}$$

The equality in the second line herein follows from the fact that the attacker will move one avatar at a time, so that no two avatars will simultaneously reach v_0. The first avatar to reach v_0 will make all others stop, so that the respective events become disjoint.

2.3. Solution concept

An *instance* of Cut-The-Rope is a quintuple $(G, v_0, AS_1, AS_2, f_N)$, containing: the attack graph $G = (V, E)$, the target node $v_0 \in V$, the defender's possible spot check locations $AS_1 \subseteq V \setminus \{v_0\}$, the possible locations $AS_2 \subseteq V \setminus \{v_0\}$, for the attacker's avatars. These

avatars will move towards v_0 along the attack paths encoded in G, taking a random number N of steps distributed according to the probability density f_N. This density determines the particular behavior of the attacker, relative to the defender's actions, and will be generally given in Section 3, and instantiated for the two real-life use-cases in Section 5.

A *solution* for a given instance is obtained with standard techniques to compute Nash equilibria: With both players having a finite set of choices, and the utility $\Pr(\text{adversary's location} = v \mid V(\pi|_c))$ derived from the location distribution (5) that depends on the attack path π, movement pattern f_N and spot-check location c, we end up with a (normal-form) matrix game that we can analyze for an equilibrium using known techniques. The solution concept used in this work is a *security strategy* for the defender, having the following (informal) semantics: it is the best randomized choice rule $\mathbf{x}^* \in \Delta(AS_1)$ such that

$$u_{\text{defender}}(\mathbf{x}^*, \mathbf{y}^*) \leq u_{\text{defender}}(\mathbf{x}^*, \mathbf{y}) \quad \text{for all } \mathbf{y} \in \Delta(AS_2) \tag{9}$$

That is, the defender can, upon playing the optimal spot checking strategy $\mathbf{x}^*$, enforce the worst-case minimal likelihood for the attacker to reach v_0, for all choice rules $\mathbf{y} \in \Delta(AS_2)$, i.e., irrespectively of what the attacker actually does.

The security strategy is computable by solving a conventional matrix game, which is finite since there are only finitely many spot check locations, and likewise finitely many attack paths. The game matrix is thus computable by evaluating formula (7), for all locations $c \in AS_1$ and all paths $\pi \in AS_2$. The Nash equilibrium of this game is $(\mathbf{x}^*, \mathbf{y}^*)$, in which $\mathbf{x}^*$ is the sought security strategy, and $\mathbf{y}^*$ is the optimal choice rule for the attack paths towards v_0.

The latter information is, however, of limited use for the defender, since equilibria are generally not unique. Therefore, taking $\mathbf{y}^*$ as a guidance on where to find for the invisible intruder with highest probability can be misleading, since there may be (plenty of) other equilibria giving entirely different advice.

On the contrary, since the saddle point value giving the lower bound value in (9) is invariant w.r.t. different equilibria $(\mathbf{x}^*, \mathbf{y}^*)$ any alternative defense advice cannot accomplish any better lower bound for the defender. Hence, $\mathbf{x}^*$ is in fact useful as optimal advice.

Remark 1. The original solution concept proposed in Rass et al. (2019) has been a perfect Bayesian equilibrium, but this raises issues with the interpretation of the results. While the game's setting formally *fits* into the definition of a perfect Bayesian equilibrium as given by (Fudenberg and Tirole, 1991), it *does not fit* equally well into the interpretation thereof: the game is not about signaling, while the solution concept in Rass et al. (2019) took an equilibrium designed for signaling games. Also, there is no random conditioning on adversary types, which a Bayesian equilibrium would require. Rather, Cut-The-Rope is – from the defender's point – played with avatars, all of which concurrently move on their routes, without a particular type choice made by nature. Therefore, a security strategy (computed as a Nash equilibrium) is the more suitable solution concept.

3. Movement patterns

In lack of any particular knowledge about the difficulty of the attack path, a simple heuristic is to just use the *shortest path*, in a graph theoretic sense. This will later also be the intuitive benchmark (see Section 5.1) to compare the defense obtained from Cut-The-Rope to a defense based on the (plausible) assumption that the attacker takes the shortest/easiest route towards v_0.

3.1. Periodically active defender

In the simplest case, originally proposed in Rass et al. (2019), we assume that the defender becomes active in fixed time intervals that are known to the attacker. The *unit of time* (see the previous section) is herein the period in which the defender becomes active (each day, each week, or similar). Furthermore, we assume no particular cost for the attacker to penetrate (this case is covered in Section 3.2). This corresponds to the situation of having a "just conceptual" attack graph, displaying general strategies to penetrate, but without reference to concrete exploits, CVE numbers or similar.

During the defender's idle times, we assume an average number $N \sim \mathcal{P}ois(\lambda)$ of steps towards its target at "average speed λ". This analytical choice is common in related literature (see, e.g., the FlipIt Game (van Dijk et al., 2013) to describe APTs, calling this strategy "exponential"). Empirically estimating the rate parameter from data, for example, taken from intrusion detection or other monitoring systems is an interesting challenge of independent research.

The function f_N for a periodic defender and attacker with average speed λ is the Poisson distribution density

$$f_N(n) = f_{\mathcal{P}ois(\lambda)}(n) = \frac{\lambda^n}{n!} e^{-\lambda}, \tag{10}$$

which would be substituted into (2) and (3) to set up the game.

The value λ must be set relative to the frequency at which the defender takes actions. For example, if the attacker makes two attempts per day, and the defender does one spot check per week, then we have $\lambda = 2 \times 7 = 14$. If the defender checks twice per day, then the attack rate is $\lambda = 2 \times \frac{1}{2} = 1$. The actual choice of λ was, experimentally, found to mostly impact the likelihood to hit v_0. The defense advice, however, did not significantly change (see Appendix A), meaning that an inaccurate choice of λ in practice will deliver a respectively inaccurate estimate on how likely v_0 will fall, but can nonetheless deliver a valid best defense recommendation.

We assume that the defender has knowledge (or a reasonable assumption) about λ, so that s/he is able to adapt the defense to it accordingly, as the security resources permit. The choice of λ itself can be considered as a strategic decision for the attacker too, knowing the defender's behavior. However, we do not explore this variation here any further, as it leads to a different game, but point out this investigation as a separate research question. We refer to the work of Xiao et al. (2018) as being a game about computing the optimal check-intervals explicitly with help from prospect theory, and under some assumptions on the attacker's attitude (risk appetite, etc.), but not considering attack graphs.

3.2. Probabilistic success on exploits

The attacker may not necessarily succeed in every penetration that it attempts. As before, if we assign probabilities[5] $q(e)$ to express the chance of a successful exploit e on the respective attack path. Figure 2 shows some (not all) such exploits as annotations "e" in gray color. Formally, $q(e)$ could be equated to the likelihood of meeting some precondition to penetrate a node. Let us slightly change the view to think of an attack path π as a sequence of exploits $\pi = (e_1, e_2, \ldots)$ (instead of nodes). Then, the chances to progress forward by a lot of $n = 0, 1, 2, \ldots$ steps is no longer Poisson distributed; rather, assuming stochastic independence of ex-

[5] For example, using subjective probability, prospect theory and generally empirical studies on human risk perception and subjective assessments, CVSS ratings to derive probabilities from, and others. Helpful related work hereto was done by Hota and Sundaram (2018); König et al. (2018); Xiao et al. (2018).

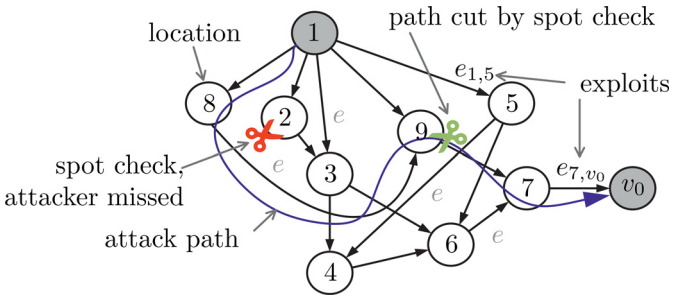


Fig. 2. Attack Graph from Singhal and Ou (2011) used in Rass et al. (2019) to play Cut-The-Rope, showing only the topology, but no technical details.

ploits, the chances to take $n = 0, 1, 2, \ldots$ steps are

$$f_N(n) = (1 - q(e_{n+1})) \cdot \prod_{k=1}^{n} q(e_k), \tag{11}$$

i.e., the probability to succeed with exactly n exploits, and to fail on the $(n+1)$st step on the attack path. This function then goes into (2) and (3) to instantiate the game under the setting described here. A *unit of time* is, again, the period between two appearances of the defender in the system, again taken as fixed and constant over time (e.g., one day, one week, etc.).

3.3. Checks with random intervals ("exponential Strategy")

If the defender becomes active at its own random (Poisson) rate λ_D, the attacker will be able to take a $\mathcal{P}ois(\lambda)$-distributed number of steps in an exponentially distributed pause time controlled by the defense intensity λ_D. This defense regime defines a *random unit of time*, whose long run average is exactly λ_D.

This change of the setting amounts to a humble change of the Poisson distribution into a geometric distribution, because: we now have two types of events to consider, which are activity of the attacker at rate λ and activity of the defender, at rate λ_D. Within a unit of time, we will thus have a number k_A of attack events, vs. a number k_D of defense actions. So, the likelihood of the defender to become active is (frequentistically) approximated as $p = k_D/(k_A + k_D) = \frac{1/n \cdot k_D}{1/n \cdot (k_A + k_D)}$ for all $n > 0$. The last term, however is the average number of events per n time units, which upon $n \to \infty$ converges to λ for k_A/n and to λ_D for k_D/n. Thus, the probability for an action to be taken by the defender is $p = \lambda_D/(\lambda_D + \lambda)$, and the number of trials that the attacker can take until the defender becomes active again is a geometric distribution with that parameter p. Conceptually, the model thus remains unchanged, except that the attacker's step number is now computed using the geometric distribution density with the given rate parameter. Consequently, we have

$$f_N(n) = p \cdot (1 - p)^n \quad \text{with} \quad p = \frac{\lambda_D}{\lambda_D + \lambda} \tag{12}$$

in (2) and (3).

3.4. Spot checks with random intervals and probabilistic success on exploit

Unlike before, we now consider a *fixed unit of time*, in which an exploit for a given vulnerability can be tried. The defender comes back in random intervals, measured in the this (fixed) unit of time, and has an average return time of λ_D. Consequently, the time window for the attacker to run exploits is an exponentially distributed random variable $W \sim \mathcal{E}xp(\lambda_D)$. Within this time window W, the attacker ought to accomplish n exploits, along an attack path $\pi = \theta \to w_1 \to w_2 \to \ldots v_0$, in the notation of Section 2.2. Like in Section 3.2, let us call e_k the edge into node v_k, which carries a known exploit complexity as the quantity $q(e_i) = \Pr(\text{exploit on } e_i \text{ is successful within a (fixed) unit of time})$. Then, an exploit on edge e_i takes an exponentially distributed time $T_i \sim \mathcal{E}xp(1/q(e_i))$. The total time for n exploits is thus $T_1 + T_2 + \ldots + T_n$, which, unfortunately, does not admit a closed analytical expression for its distribution, since the values can be assumed independent, but not identically distributed. To escape the issue, we simplify matters by assuming the avatar to move at a uniform velocity along the attack path, instead of being faster and slower depending on the attack complexities. We believe this assumption to be mild, since our main concern is the time it takes to reach the end v_0 anyway, and we are not as much interested in determining the avatar's location anywhere in the middle of the attack path.

This simplification comes to a geometric mean of the probabilities

$$\overline{q} = \text{geomean}\{q(e_i) \mid e_i \text{ is on the chosen attack path}\}.$$

The point is that the product of the actual probabilities, i.e., the chance to hit v_0, remains unchanged hereby, since $\prod_i q(e_i) = \overline{q}^{|V(\pi)|}$ where $|V(\pi)|$ is the length of the attack path. Let us put $\lambda_\pi := 1/\overline{q}$ to bring the notation closer to that of Section 3.3, since the result (to come later) will also be close to this previous finding. The subscript π to λ_π herein reminds about the attack rate now to depend on the chosen path.

Under this simplification, the time for n exploits is the sum of all identically $\mathcal{E}xp(1/\overline{q})$-distributed random variables $E_n := T_1 + T_2 + \ldots + T_n \sim \mathcal{E}rl(\lambda_\pi, n)$ that is Erlang distributed. We are interested in the probability of $T_1 + T_2 + \ldots + T_n \le W$, which is a matter of computing a convolution integral. We shift the algebraic details to the appendix, and directly give the result here:

$$f_N(n) = \begin{cases} \Pr(E_N \le W) = \left(\frac{\lambda_\pi}{\lambda_\pi + \lambda_D}\right)^n & n \ge 1; \\ \Pr(E_1 > W) = 1 - \Pr(E_1 \le W) & n = 0. \end{cases} \tag{13}$$

Observe that this is movement pattern is like in Section 3.3, which is yet another geometric distribution, only with the different parameterization.

The approach of geometric averaging over the entire attack path deserves a bit of discussion: we could equally well average only across the segment of length n that the attacker targets to overcome, and/or exclude all exploits with $q(e_k) = 1$ from the averaging. We refrained from both these options for two reasons: first, removing the 1es from the averaging would unrealistically shorten the attack path to less than its physical reality. Even if an exploit has a 100% chance to be used within short time, there is nonetheless a time step necessary to do it, so including it in the geometric mean seems plausible. An attack path that is longer will, despite the same product probability of accomplishing it, take a proportionally longer time to traverse. Second, concerning the focus on only a segment, this may miss the actual intention of the adversary, since it targets the end of the attack path, and not only a specific segment. In other words, geometrically averaging only over the first k exploits would be the assumption that the adversary would stop at the kth step, even if there is time left before the defender comes back. Since the target is getting to the end of π, it appears plausible to include all exploits towards this end.

4. Case studies

We dedicate the next couple of subsections to numeric results, starting with a brief correction to past calculations in the literature, and then moving onward to the new case studies and the

comparison of defense policies optimized with Cut-The-Rope, versus a heuristic common-sense defense policy.[6]

To assess the game w.r.t. a real-life application, we conducted two case studies on the industrial robots in Section 4.3. The game is similar to capture-the-flag competitions known in ethical hacking, since there and also here, the goal is to "capture" a target asset v_0. Our analysis, different from ethical hacking, is purely game-theoretic and optimization-based here.

4.1. Implementation remarks

We adapted the implementation from Rass et al. (2019) and thereby discovered a few bugs in this older code that we corrected in our version.[7] The original code used fictitious play on the full distribution $U = (u_1, u_2, \ldots, u_n = \Pr(\text{adversary reaches } v_0))$ obtained from Eq. (5). We compute an optimum U' w.r.t. a lexicographic order from right to left, first minimizing the last coordinate u_n, and breaking ties by continuing to minimize u_{n-1} while keeping u_n at minimum. The next tie is broken using u_{n-2}, while keeping the so-far optimized coordinates at their minima and so on. This introduces a dependence on the ordering of the coordinates, corresponding to a likewise ordering of locations in the attack graph. Therefore, the solution returned by the implementation from Rass et al. (2019) is ambiguous in the sense of depending on the node ordering.

The optimization, however, independently of the node order, always minimizes the chances to reach the target asset, and hence provides a valid defense policy w.r.t. the targets of the defender. Our implementation inherits this dependence on the node order, but since our sole interest is reaching or avoiding to reach v_0 anyway as (7) and (8) define, this ambiguity is not a limitation. In light of this, we chose the graph-topological sorting to order the probabilities in U' other than for v_0, which is the last element in this vector.

The attack graphs for our robot case studies have several entry points for the attacker and also several targets to reach. To handle them all in a single run of the analysis, we *added an artificial* (virtual) *entry node* from which all (real) entry nodes are trivially reachable (with probability 1). Since Cut-The-Rope in the original version, analyzed here, assumes only one target, we *contracted the multitude of target nodes* into a single "compound" target node. This corresponds to the target being to reach any of the possible target nodes, not distinguishing which in particular. A *target* node is, by default in our implementation, any node that does not have descendants (zero out-degree in the attack graph). Consequently, all inner nodes, except the virtual start, are possible defense spot-check locations. The technical simplification towards having one target (only) is to avoid multi-criteria optimization, which is theoretically possible (even supported by the packages to run the optimization practically), but is more involved to interpret for a defense policy.

4.2. The example from Rass et al. (2019)

To soundly align with past results, we first reproduce the examples from the original reference (Rass et al., 2019) proposing Cut-The-Rope. Figure 2 shows the attack graph from Rass et al. (2019) in a more compact form: the battlefield is an attack graph constructed for a network with one desktop computer, connected to a switch that also serves a file- and a database server. The connection from the desktop machine is protected by a firewall. The respective attack graph, whose full details are found in Singhal and Ou (2011), is here represented by a directed acyclic graph: its nodes correspond to possible physical or logical locations of the intruder, corresponding to nodes of the attack graph. Directed edges are – in our example – labeled by exploits, say, $e_{i,j}$, to mean that some exploit (e.g., a buffer overflow, remote shell execution, etc.) is necessary to get from location i to location j in the attack graph (others, not all of them, appearing simply as "e" to indicate their presence on all links in the graph without overloading the picture). Two designated nodes, shown in gray in Fig. 2 mark the entry point for the attacker (the desktop PC), and the adversary's target node (here, the database server), denoted as v_0.

For a deeper insight into how much a defense based on Cut-The-Rope can offer, we first tried to reproduce the results from the original work (Rass et al., 2019), thereby discovering a calculation error in how the paths were implemented in the code (in detail, three paths had nodes on them towards which there was no edge in the graph that Fig. 2 displays). Correcting these issues in the code, and running it again (under the current setting), we obtained slightly different results than (Rass et al., 2019): the optimal protection in case that the defender can spot everywhere is likewise to protect node ⑦ (thus confirming the defense computed in Rass et al. (2019)), but the most likely attack paths were different. Given that these are hypothetical anyway, the important finding lies in the defense recommendation, which, despite past code errors, was nonetheless correct in Rass et al. (2019). The results were, however, largely different if the defender's action set is restricted to guarding only some FTP connections (specifically guarding at locations ②, ⑤, ⑥, ⑧ and ⑨ only), recommending the optimal defense to be on nodes ⑥ and ⑨ with probabilities $\approx 54.91\%$ and $\approx 45.09\%$, leaving a residue chance of $\approx 12.4\%$ for the attacker to reach the target v_0.

4.3. Robot case studies

For both of the robot cases to follow, we give computational results and a discussion of their practicality. To avoid confusion between the attack graphs appearing here and those found in the cited literature, we use the original versions thereof to visualize the battlefield and results. The actual simulation was done on an attack graph with added virtual starting and a single compound target node (if more than one exists).

It is perhaps practically interesting to remark that both attack graphs have inner nodes that classify as attack targets, but have descendant nodes as subsequent attack targets. With the convention of taking nodes with zero out-degree in the graph as targets (see Section 4.1), the simulation will include all "inner" nodes as defense locations even though they may be attack targets too. This is not precluded by the game design, and may be interpreted as considering inner nodes as "intermediate targets" whose prevention may avoid subsequent final, perhaps more dangerous, attack targets. The game's defense policy would then advise to prevent a certain attack sub-target in the attack graph, with a certain level of effort (expressed as likelihood). From a simulation perspective, including or excluding any node from the defense policy is a simple matter of defining the action set for the defender accordingly.

4.3.1. Case #1: modular articulated robotic arm (MARA)

MARA is a collaborative robotic arm with ROS 2.0 in each actuator, sensor or any other representative module. Each module has native ROS 2.0 support, can be physically extended in a seamless manner and delivers industrial-grade features including synchronization, deterministic communication latencies, a ROS 2.0 software and hardware component life-cycle and more. Altogether,

[6] The full code is available for download at https://github.com/jku-lit-scsl/ComputersAndSecurity_RoboticsCaseStudies_Cut-The-Rope.git

[7] The full code is available for download at https://github.com/jku-lit-scsl/ComputersAndSecurity_RoboticsCaseStudies_Cut-The-Rope.git

MARA empowers new possibilities and applications in the professional landscape of robotics. The use case considered contemplates the MARA modular robot operating in an industrial environment while performing a pick & place activity. Details about MARA for this case study can be found in AcutronicRobotics (2021); Alias Robotics (2019).

4.3.2. Case #2: MiR100 - Mobile industrial robotics

The MiR100 autonomous mobile robot is advertised as a safe and cost-effective mobile robot that quickly automates your internal transportation and logistics. The robot claims to optimize workflows, freeing staff resources so you can increase productivity and reduce costs. A case study analyzing the cyber-resilience of MiR100 robots was conducted and documented at Alias Robotics (2020) and Alias Robotics (2021), which considered a single robot operating in a structured environment while connected to a local area network that gets compromised. Through the local area network, prior work demonstrated how an attacker could exploit vulnerabilities, pivoting across subsystems in the robot all the way into its safety system, disabling it fully in a remote manner.

For both robots, we took attack graphs out of industrial security assessments, which, in the particular case of MiR100, were also annotated with Common Vulnerabilities and Exposures (CVE) and CVSS information, which allows an assessment of the "hardness" of vulnerabilities along the attack path. Such annotations were not available for the MARA use case, which, in lack of such details, suggests an application of the Poissonian movement pattern of Section 3. The more detailed attack graph for the MiR100 robot enables the consideration of probabilistic success on exploits as Section 3.2 described.

5. Results and comparison

To evaluate how much a game-theoretic defense may add to the security, we do not only give the absolute results from the simulations, but also compare them to a heuristic best-effort defense policy, described in Section 5.1. Its simulation is run likewise with each of the four movement patterns from Sections 3, with the probability to reach v_0 given for each case as (i) optimized by Cut-The-Rope versus (ii) according to a best-effort defense.

5.1. Baseline comparison: a best-effort defense policy

For an assessment of the quality of the game-theoretic defense, let us use the following heuristic defense policy to compare:

- We assume that an adversarial avatar will always follow the shortest, or "easiest" attack path towards v_0. The distinction between shortest and easiest is made in dependence of how much is known about exploit complexities. In the MARA use case, the path choice will be for shortest, in terms of the number of exploits, since there is no further detail given about the exploit complexities. In the more detailed MiR100 use case, we have attack complexities and can likewise apply a shortest path algorithm to guide the attacker to the path whose success probability (as the product of all exploit success probabilities) is maximal.[8]
- The defender, unbeknownst of where the attacker is, and unable to actively detect it, applies a uniformly random defense strategy. That is, if the attacker is equally likely to be anywhere in the system, the defense policy would likewise be a uniformly random spot checking.

[8] The usual trick of assigning the negative logarithm of probabilities as edge weights and computing a shortest path in the well known way

Under these hypotheses, we apply the same mechanism as in Cut-The-Rope, i.e., we let the attacker follow its chosen (shortest/easiest) path, and be occasionally sent back by the defender upon a coincidental cut of the path equivalently, closure of any backdoor. If so, then the avatar will keep retrying, until it hits the final target v_0. Note that this regime also includes lateral movement, since we still have a multitude of avatars attacking in parallel, each on its individually optimal route from its starting location θ towards v_0.

We implemented this defense policy simulation by adapting the code from the implementations of Cut-The-Rope accordingly, to implement the heuristic defense and attack policy of above. Like for the game optimization, the heuristic defense implementation outputs the probability to reach v_0 by simulating this defender-attacker interaction. We remark that this heuristic defense may still be overly optimistic relative to real life situations, in which defense teams may have only an incomplete view on the attack graph $G = (V, E)$. The defender would thus only be active on a subset $D \subset V$, so that all nodes in $V \setminus D$ would be zero-day exploits.

5.2. Overview of experiments

In total, comparing the periodic/exponential defense strategy against a randomly moving adversary in two use cases, gives a total of 4 evaluation scenarios, each accompanied with its own comparison to the baseline heuristic of Section 5.1. Table 1 relates the sections and figures in the following to these four configurations.

5.3. MARA: results

The attack graph for the MARA robot is taken from AcutronicRobotics (2021) and shown in Figs. 3 and 4. This graph has 11 nodes and 10 edges in total, among them one entry point (node ①) for the attacker, and two targets (nodes ⑥, and ⑨).

We played Cut-The-Rope on this graph with a periodic defender versus an attacker that takes an average of 2 moves per time unit (i.e., in-between two appearances of the defender, e.g., per day). Figure 3 shows a table with the probabilities to spot-check each node on the attack graph. For the visualization, we have put bubbles on the attack graph, whose size corresponds to the probability of spot-checking there. That is, the larger the bubble, the more effort should be out on defending at this point.

Turning to the case of the defender coming back in random time intervals, we let the game run in three configurations, with the defender moving slower ($\lambda_D = 1 < \lambda$), at equal speed ($\lambda_D = \lambda = 2$) and faster than the attacker ($\lambda_D = 3 > \lambda$) in (12). The resulting spot checking probabilities are again displayed as bubbles located at the respective nodes in the attack graph, and put over one another in Fig. 4.

The numbers and bubbles are almost of the same size, showing that for the defense locations, the speed of spot checking has only a negligible impact, while the performance of the defense accordingly becomes better if the defender is "more active". The performances of the defense policy as displayed in the bottom table of Fig. 4 show that the optimized defense pays over the heuristic "blind" spot checking policy.

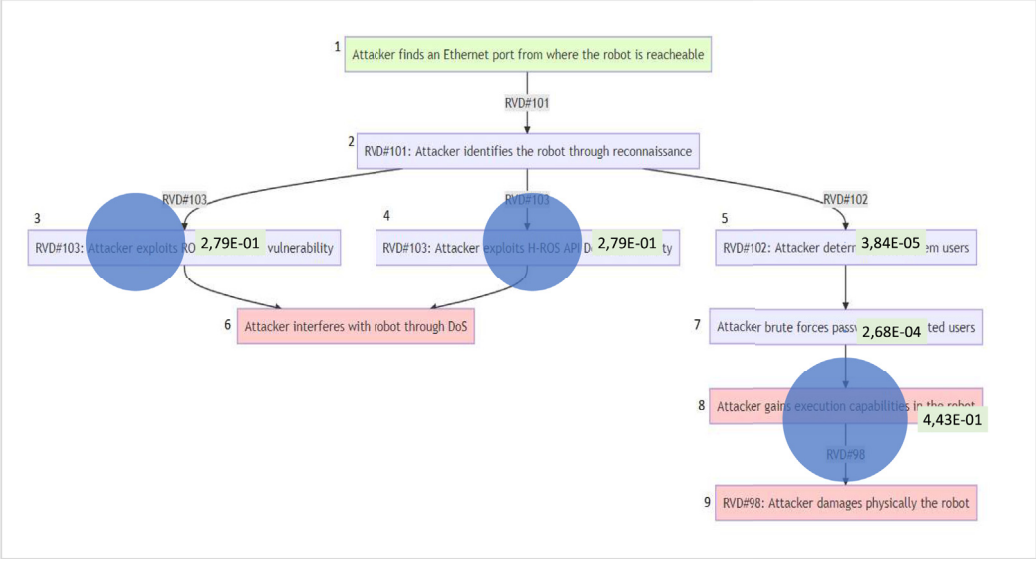
The takeaways from these findings is not that a more intense defense activity will reduce the chances of the attacker (this would be obviously the case), but rather giving the defender an indication of where to allocate its (limited) resources to gain the best possible effect. Without signaling and without additional information in the attack graphs, the results are necessarily a crude approximation of reality, and Cut-The-Rope has been designed to be workable in such a situation of limited information, as well as with cases when

Table 1
Overview of experiments.

Use case \ defender's policy	Periodic	Exponential strategy
MARA (no particular exploit hardness annotations), Section 5.3	attacker movement model: Section 3.1 results shown in: Fig. 3	attacker movement model: Section 3.3 results shown in: Fig. 4
MiR100 (known exploit complexities to consider), Section 5.4	attacker movement model: Section 3.2 results shown in: Fig. 5	attacker movement model: Section 3.4 results shown in: Fig. 6

Optimal defense policy:

Node	Probability
2	0
3	0.279
4	0.279
5	0.0000384
7	0.000268
8	0.443



Efficacy of the optimal vs. heuristic defense:

defense policy	chance to hit v_0
CUT-THE-ROPE	12.8%
heuristic	30.6%

Fig. 3. MARA use case results for periodic spot checks (Section 3.1).

more details are available, such as for the MiR100 robot following next. The results in the rather little detailed MARA use case are quite evident but therefore also plausible ("guard the closest graph cut between the asset and the defender"). The nontrivial indication here is the advice to let the attacker come "close" to the asset, while a defender would perhaps otherwise try to guard the outer perimeter of the system to keep the intruder out in first place. The optimum to be at the closest graph-cut towards the asset is here explainable by our assumption that the attacker is stealthy and can start from anywhere, and in a practical situation, the defender may indeed have no reliable information about infected parts (otherwise, it would be trivial to disconnect and repair/replace the malfunctioning component). The defense policy that CUT-THE-ROPE computes is for practitioners operating blue teams that need to protect a large attack surface with no monitoring or signaling. A game-theoretic defense can help prioritize resources.

5.4. MiR100: results

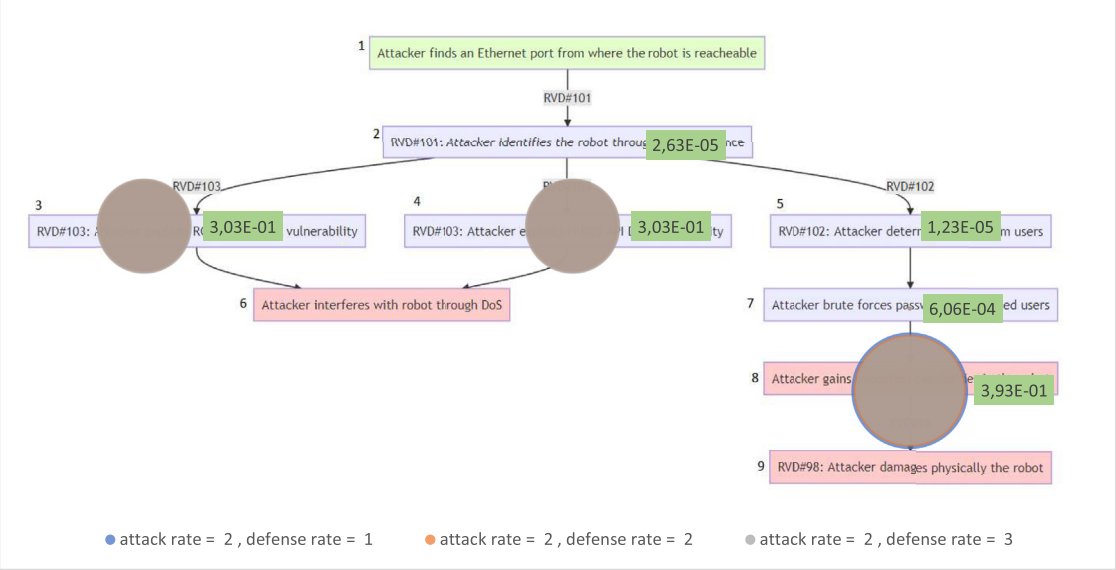
Similarly as for MARA, we used an attack graph for the MiR100 robot as shown in Figs. 5 and 6. The attack graph has 16 nodes and 24 edges. The attacker can enter at four points (nodes ①...④), and four targets (⑫, ⑬, ⑭ and ⑯).

We conducted the likewise experiments under the same configurations as for the MARA use case, but this time making use of the CVE annotations to give information on how hard it is for the attacker to mount an exploit. For the defender, we again assume this one to be periodically active (as in Section 3.1) and to randomly spot check (as in Section 3.4). Note that in this case we do not have an attack rate λ as for the MARA use case before, since the movement of the attacker is solely governed by the difficulty to mount exploits.

It is interesting to note that the optimal defense policy does not advise to guard node ⑦ or ⑨, which is a way towards reach-

Optimal defense policy:

Node	probability for $\lambda_A = 2$ and		
	$\lambda_D = 1$	$\lambda_D = 2$	$\lambda_D = 3$
2	0.000147	0.0000502	0.0000263
3	0.272	0.291	0.303
4	0.272	0.291	0.303
5	0	0	0.0000123
7	0.000473	0.000548	0.000606
8	0.455	0.417	0.393



Efficacy of the optimal vs. heuristic defense:

defense policy	chances to hit v_0		
	$\lambda_D = 1$	$\lambda_D = 2$	$\lambda_D = 3$
Cut-The-Rope	7.4%	5.7%	4.6%
heuristic	17.8%	13.8%	11.2%

Fig. 4. MARA use case results for spot checks at random intervals (Section 3.3).

ing goal node ⑯. This may be assumption of the game, of the attacker already being somewhere in the network. The defense policy accounts for this and hence does not put more weight on lower nodes with higher incidence index. This way, the model accounts for defense in depth rather than entry prevention.

Finally, let us turn to the case of the adversary working towards v_0 only in random time intervals between two appearances of the defender. This time, the defender's parameter λ_D is the average "window size" W (see Appendix B), measured in units of time, e.g., days. It is the time that we give the attacker to mount activities in the game. The results are shown in Fig. 6.

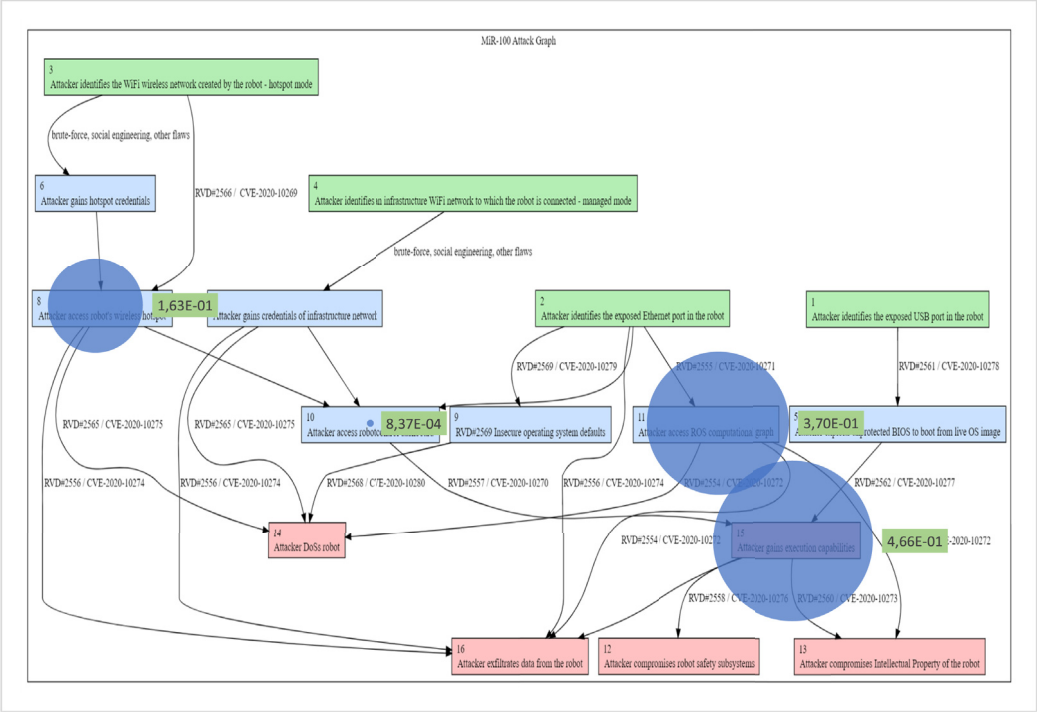
Similar as for the MARA use case, the defense locations are the same in all cases, with the defense efforts only slightly differing according to how large the window is for the attacker, respectively, how frequently the defender comes back. The performance of the defense is shown in the bottom table of Fig. 6. Consistent with the intuition, the attacker's chances to reach v_0 become larger if the defense window is made larger. In both, the experiments with the heuristic defense and optimized under Cut-The-Rope, the value λ_D gives the average number of time units before the defender comes back. That is, larger λ_D give the attacker more time to exploit (conversely to the interpretation of λ_D in the other experiments, where it was the frequency of the defender's return). Again, the experiments show that Cut-The-Rope outperforms the heuristic defense considerably.

6. Related work

APTs, like most targeted attacks conducted by cybercriminals, due to their diverse combination of attacks, hardly admit a single model to capture them; rather, they call for a combination of models designed for different aspects or characteristics of the attack. Game-theoretic defense models may be distinguished according to the nature of APT (Rass et al., 2020) that they cover: there is the parasitic type, in which the attacker tries to steal resources for as long and much as possible, but does not aim to kill its victim. Related models are FlipIt (van Dijk et al., 2013; Zhang and Zhu, 2019) and its descendants. Minimizing the total time that the

Optimal defense policy:

Node	Probability
5	0
6	0
7	0
8	0.163
9	0
10	0.000837
11	0.37
15	0.466



Efficacy of the optimal vs. heuristic defense:

defense policy	chance to hit v_0
Cut-The-Rope	7.7%
heuristic	29.8%

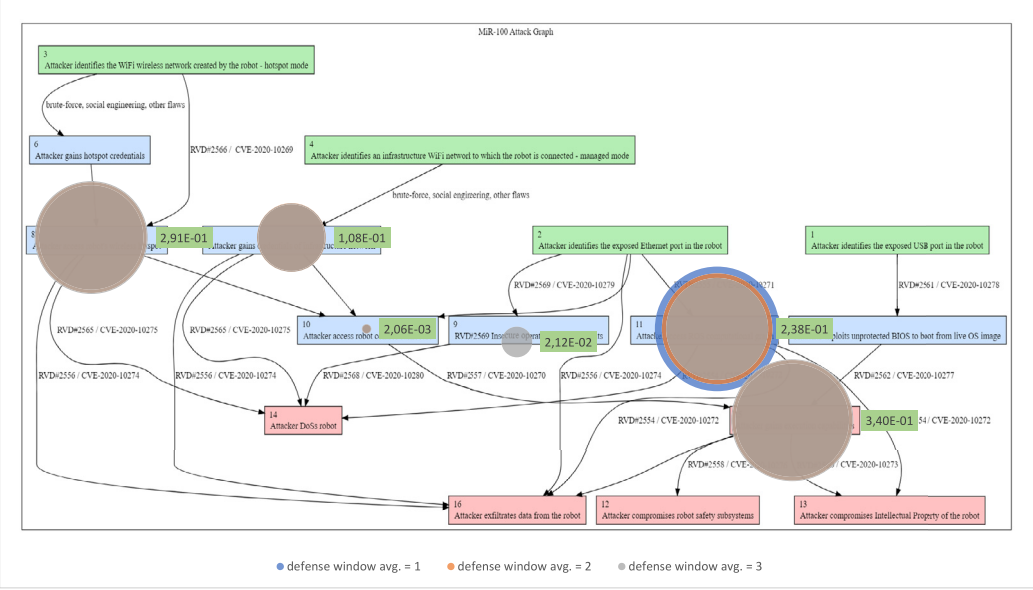
Fig. 5. MiR100 use case results for periodic spot checks (Section 3.1).

attacker spends in the system may not necessarily minimize damage too, since the attacker may entirely destroy the asset v_0 even within a very short period of time. The defender may nonetheless suffer a permanent defeat (upon loss of v_0). For example, if the attacker can gain access to the security controls of a nuclear power plant even for a very short time, this may be sufficient to cause an unstoppable meltdown. Conversely, the attacker may spend a considerably larger amount of time in other areas of the nuclear power plant's system; as long as there is no vital subsystem to fiddle with, the damage to the infrastructure may be bearable. This motivates the consideration of the second type of APT, for which the game model Cut-The-Rope is tailored to: there, the attacker aims to kill the victim and silently prepare the final blow. A documented case of this is Stuxnet (Kushner, 2013), and Cut-The-Rope is a game model designed for this latter type.

Many other game models are aligned with the phases in the kill chain, and most related work (Etesami and Basar, 2019) is specific for at least one of them. We note that the ADAPT project (ADAPT, 2018) covers a wide spectrum of aspects and phases here. Specific defense models include the detection of spying activities (Qing et al., 2017), tracing information flows (Moothedath et al., 2018), detection of malware (Khouzani et al., 2012), deception (Carroll and Grosu, 2009) also via honeypots (La et al., 2016), attack path prediction (Fang et al., 2014), path selection to support malware detection in distributed networks (Panaousis et al., 2017), and general network defense (Alpcan and Basar, 2010) to name only a few. Our game is in a way similar to that of the seminal work (Lye and Wing, 2005), yet differs from this previous model in not being stochastic, and in using payoffs that are not real-valued. The stochastic element is included in a much simpler way in our model, yet preserving information about uncertainty in a full distribution, to avoid losing information by averaging out randomness (for example, replacing a random payoff by a real-valued expected payoff).

Optimal defense policy:

Node	probability for		
	$\lambda_D = 1$	$\lambda_D = 2$	$\lambda_D = 3$
5	0	0	0
6	0	0	0
7	0.101	0.108	0.108
8	0.249	0.281	0.291
9	0	0	0.0212
10	0.00194	0.00161	0.00206
11	0.354	0.28	0.238
15	0.294	0.33	0.34



Efficacy of the optimal vs. heuristic defense:

defense policy	chances to hit v_0		
	$\lambda_D = 1$	$\lambda_D = 2$	$\lambda_D = 3$
Cut-The-Rope	2.4%	3.5%	4.1%
heuristic	11%	16.1%	19%

Fig. 6. MiR100 use case results for spot checks at random intervals.

Since the methods applied here come from the risk management field, this relates our work to that of Yang et al. (2018), who presents a framework to optimally respond to a detected APT. Their work is thus an *a posteriori* treatment after the APT succeeded, while ours complements the risk management here by an *a priori* treatment to prevent the APT from success. Likewise notable is also the work of (Hota et al., 2018; 2016), who consider interdependency graphs in relation to attack graphs in a game-theoretic analysis of targeted attacks. Their work adds constraints to budgets or desirable risk levels, and is specifically about investments in defenses of nodes and edges, but also works with crisp payoff measures (such as, e.g., paths of maximal attack probability or similar).

A different classification of related work is based on the protection targets. defenses can be optimized for confidentiality (Lin et al., 2012), the monetary value of some asset upon theft or damage (Zhu and Rass, 2018), or the time that an adversary has parts of the system under control (van Dijk et al., 2013). This distinction can be important depending on the context, as industrial production typically puts priority on availability and integrity, with confidentiality as a secondary or tertiary interest. Conversely, whenever personal data is processed, confidentiality becomes the top priority, putting availability further down on the list.

The techniques applied to capture and defend against APTs are manifold, but in most of these (like in our work), the network graph is in the center of attention: it may define how an attack evolves as a dynamical system (Senejohnny et al., 2018; Yang et al., 2019) inside the graph topology, with the challenge of optimized orchestrated defense. A good defense design that needs to account for new vulnerabilities potentially being opened up when closing known security holes. The work of Touhiduzzaman et al. (2019), in this regard, utilizes a game model for graph coloring for a systematic and optimized defense, applying these results to industrial bus systems. Another dynamic yet queuing-based model is that of Li et al. (2019), which like our model computes optimal

resource allocations by the defender and attacker, as an aid for decision making. Tailoring the attack model more closely to the application domain for the sake of a more accurate description, the work of Soltan et al. (2019) provides insightful connections of graph topological properties of a power grid, and how areas in danger of becoming attacked are identifiable from analyzing the graph.

The work of Pawlick et al. (2019); Pawlick and Zhu (2017) takes a more birds eye perspective on the domain of the internet of things (IoT), and applies it directly to or varies the FlipIt game (see the references above and Zhang and Zhu, 2019) to model individual parts of a cloud-based IoT infrastructure, combining these submodels into a larger hybrid game model that allows certain equilibria to play optimally against the adversary. Another cloud-related and -specific APT defense model is Yuan et al. (2019). Like us, they adopt a leader-follower model, but different to our work, they use a Stackelberg equilibrium concept.

Taking the APT as a long term yet one-shot event, an attack graph can be treated as a (big) game in extensive form. From this point of view, it is possible to think of the APT as an instance of the induced gameplay, to which Bayesian or subgame perfect equilibria can be sought (Huang and Zhu, 2018). More similar to this work, we can treat the APT as a game of inspections, to discover optimal strategies of inspection in different depths of a shell-structured defense (Rass and Zhu, 2016; Zhu and Rass, 2018). An aspect of strong relevance concerns the use of probabilities: the work of Hota and Sundaram (2018); Xiao et al. (2018) are most interesting in its account for subjective probability and prospect theory, since this includes the way of how humans bring in their individual risk attitudes in decision making under uncertainty (especially about defenses). We avoid this conceptual and practical difficulty in the modeling by designing our game with as few probabilistic parameters as possible.

Cut-The-Rope is, in two ways, different from most other game-theoretic models: first, it can let the players act in different time axes, meaning that the defender can be active in discrete or continuous time, while the attacker is (here always) acting in continuous time. This is in contrast to most other models in which both players act in fixed schedules (such as in extensive form games), or both can take actions continuously (such as in differential games). The second aspect is the added suggestion of tie-breaking if there are several equilibria. Cut-The-Rope implicitly addresses the equilibrium selection problem by refining the set of possibly many defense actions based on the probabilities to reach not only v_0, but also to get nearby it. Formally, the optimization, after having minimized the chance to conquer v_0, continues by minimizing the chances to reach a node close to v_0. As mentioned in Section 4.1 this induces a dependency on the ordering of nodes, but this ordering is up to the choice of the defender setting up the model. In any case, the defender is not left with a choice among possibly many equilibria, but can have the calculation automatically refine it in an interpretable sense. This equilibrium selection problem is not usually intrinsically addressed in other security game models.

Among the related work on attack graphs, two major types are state-enumeration and dependency attack graphs. The difference is, weather full states or attributes are used. There are also formalisms based on the network assets, such as host-based or host-centered attack graphs, whenever the focus is on the assets. Another frequently accounted high level categorisation is according to exploit- or condition-orientation of the nodes (involving logical connectives, such as AND/OR nodes). A well-written review is Lallie et al. (2020).

Various attack graph formalisms involve multiple parameters as intrinsic components of the graph model, such as (success) probabilities. These can be used for quantitative security analysis, to identify minimal sets of mitigations by hitting sets, graph cuts and minimal exploit sets (Jha et al., 2002). Along these lines, logical analyses towards enumerations of security violations, and combinatorial optimizations for finding minimal sets of components to "harden" are enable the generation of automated recommendations and rankings of vulnerabilities and assets (Shandilya et al., 2014) (e.g., using Google's PageRank algorithm Mehta et al., 2006). A natural extension to this is the inclusion of uncertainty by leveraging Bayesian reasoning and optimizing cost-benefit tradeoffs between exploit mitigations and resulting risk reductions (Zeng et al., 2019). This is not only to recommend mitigations, but to do so under economic aspects of efficiency or cost for the defender. Using a combination of logical conditions and combinatorial optimization, recent work (Wang et al., 2014) also proposed extending attack graphs with zero-day exploit edges. Besides ranks of assets, single vulnerabilities or criticality of attack paths, a variety of further security metrics is computable from attack graphs (Zenitani, 2023), for example, exploitability (e.g., average of CVSS scores), risks of exploits depending on the graph size, number of exploits relative to the total possibilities for the network to become compromised, number of unique vulnerability types referred to by explicit exploits, or various graph-theoretic scores, such as minimum, maximum and average of path lengths, number of length of cycles (or (a)cyclicity of the graph itself, thereby computing recommendations to deal with cyclic structures inside the graph), as well as the statistical distributions of these values. For models considering the graph as a Markov (decision) process, values like the conductance may be interesting in determining how quickly a stationary distribution will be reached, and to subsequently optimize the chances for the defender or attacker to succeed. The game theoretic model in this work follows the same goals, but using different methods of optimization.

7. Discussion

The experimental findings suggest that the apparent optimal defense strategy delivered by Cut-The-Rope is to guard the immediate neighborhood of the target asset, so as to cover cases where the attacker has already deeply penetrated the system when the game begins. Indeed, an analytic characterization of the optimal defense under Cut-The-Rope is obtained in Appendix A as Proposition 5. It confirms a certain graph cut to be optimal under certain assumptions, but not in all of our test cases. For this reason, we leave the discussion of analytic results as an appended remark here, and continue the discussion with more practical aspects.

7.1. Incomplete attack graphs and zero-day exploits

The heuristic defense of Section 5.1 may in reality be still over-optimistic, in its assumption of complete knowledge about the attack graph. Practical defense teams may only have a limited knowledge or possibility to construct the entire attack graph, and it is generally unaccomplishable for the defender to get exactly the same attack graph as the adversary has. The simulations implemented in this work have been made with the possibility to include only a randomly chosen subset of nodes in the defender's possibilities to spot check, so include such incomplete knowledge in the analysis. Concretely, the code was made to randomly reduce the defender's spot check locations to, for example, only 75% of the nodes in the attack graph. Under such reduced possibilities, the game runs against an attacker with more, i.e., full, knowledge about the attack graph. We confine ourselves here to reporting that the defense policies performed worse than under full

knowledge (not surprisingly), but both policies (Cut-The-Rope and the heuristic) lost performance at approximately equal magnitudes, leaving their relative quality over one another without substantial changes.

We emphasize that a simulation under such reduced knowledge for the defender, whereas giving the attacker full knowledge, can be viewed as a study of the impact of zero-day exploits used by the attacker. That is, any node excluded from the defense, but used by the attacker is nothing else than a zero-day vulnerability. Since a systematic account for this would be beyond the scope (and space limits) of this work, we will explore this route along future work. A promising possibility, besides adding zero-day edges implied by logical conditions (Wang et al., 2014), we may change the perspective towards services rather than exploits. Given a table of mutual dependencies of distinct services or applications (which is a common step in a risk management processes to identify dependencies and assess the criticality levels of services and applications), an attack may – abstractly – be considered as a disruption of a service that subsequently causes disruptions of dependent services. In turn, this also means that there would be ways in which a dependent service has access to a parent service, so that there would be an edge between these two. So, we may also obtain an attack graph by letting the nodes be services or applications, and drawing edges whenever a node depends on another node, following the intuition that "some exploit" on a node may, via their interaction, lead to access or some control over the neighboring node in the dependency graph (Cao et al., 2018; Fang et al., 2022). Playing the game on so-constructed graphs appears worth exploring in future research.

Constructing attack graphs in an automated fashion is considered via multiple methods, among them logic-based model checkers to enumerate conditions of security violations (Jha et al., 2002) or treating the graph as an automaton and analyzing it for paths that yield into conditions that violate security (Wang et al., 2012). Further techniques leverage machine learning to craft attack patterns or online-learn the attack graph (Zeng et al., 2020; Zenitani, 2023). A practical difficulty of building attack graphs especially for highly distributed systems such as (many) robotic systems are, makes applications of machine learning and automated attack graph algorithms particularly attractive in our context. Combined with automated network scanning methods, the problem is one of system identification and repeated updating of the model. The game theoretic analysis assumes a "static" battlefield to analyze the best defense at the current moment, and we can re-run the analysis after every change of the attack graph. Such updates are naturally implied by any (cyclic) risk management process, and Appendix C discusses the embedding of our analysis inside a cycle of risk assessment, -analysis, -mitigation (based on the analysis), and re-assessment of risks after a change of the threat landscape (i.e., the attack graph).

7.2. Cutting the rope vs. changing the attack graph

In our experimental instance of the game, we let the attack graph remain unchanged over time. In particular, we assume that none of the defender's actions causes a permanent removal of a certain backdoor. This is practically motivated by the fact that spot checking may remove some, but not all vulnerabilities, so that, for example, one buffer overflow vulnerability in a secure shell implementation might get fixed, but other exploits of the same kind remain open, making the respective nodes remain unchanged in the graph after an inspection. Likewise, remote shell access may be required for the business workflow and hence cannot be deactivated, but only the access credentials might be updated. In that case, the remote shell access exists before and after an inspection. Even though the game model itself uses a static attack graph, this one may itself require an update from time to time upon changes in the infrastructure. This is part of the business continuity management related to security, and accordingly changes the action sets for the defender and attacker. The implementation of the game, however, remains unaffected, except for the specification/input of the attack graph. We close the discussion at this point, referring to Appendix C for a continuation of this discussion.

7.3. Further generalizations

The movement patterns as studied in this work admit further modifications and generalizations, yet to be explored, such as:

7.3.1. Probabilistic success on spot checks

First, to the advantage of the attacker, suppose that the defender is not necessarily successful on wiping out the adversary inside a node c, which may the more "probable" case in an enterprise or embedded network. It is not difficult to generalize the model towards this: If we write p_c for the likelihood to actually cut the rope at $c \in V$ upon trying so, (5) becomes a mix of cut and uncut paths,

$$\begin{aligned}\Pr(\text{adversary's location} = v) = \\ p_c \cdot \Pr(\text{adversary's location} = v | V(\pi|_c)) \\ + (1-p_c) \cdot \Pr(\text{adversary's location} = v | V(\pi)),\end{aligned}$$

and (5) is defined alike by the entirety of all these values for all $v \in V$. Thus, the computation as such does not change, only the code needs to use the above formulas to compute the payoffs. If the probabilities are made conditional on the system state, the analysis can be made to account for changing system conditions too. We leave this route unexplored due to space limitations here.

7.3.2. Multiple adversarial targets

Cut-The-Rope may be modified towards a multi-criteria game, treating all target nodes as individual targets in the game. The concept of a security strategy has a multi-goal counterpart, which the software used for the experimental implementation already supports. The experiments reported here could, possibly, be reconducted without the merge of targets, i.e., the graph-theoretic contraction.

7.4. Complexity and scalability

The complexity of the analysis is governed by the time to solve a sequence of linear optimizations. The dimensions of these problems depend on the number of strategies for both players. Using interior point methods, the computational complexity is $O(|AS_1| \cdot p(|AS_1| \cdot |AS_2|))$ for a polynomial p that depends on the chosen optimization algorithm. The need for an exhaustive enumeration of attack paths can raise scalability issues, since the number of attack paths is worst-case exponential. However, the number of paths in an attack *graph* may become large only because many paths overlap in large portions, and the defender may consider using only a subset of paths that cover all edges in the attack graph, so as to cover all known exploits (which is a polynomial number), rather than all possible paths (whose number is exponential). Since this work was concerned with a study of the original model, which does not implement such a dimensionality reduction, this modification of the model is a possible aisle of future studies. For the use cases in this work, the number of paths was sufficiently small to admit an exhaustive enumeration. Likewise is the number of paths feasibly small if the battlefield is an attack tree, rather than an attack graph. In any case, Cut-The-Rope itself does not conceptually

change if the restriction is imposed only on the cardinality of the strategy sets to be polynomial in the number of nodes in the attack graph.

7.5. Including signals about adversarial activities

The model assumes zero information for the defender about where the adversary is located. Many real-life systems use intrusion detection, and other signaling means (here explicitly not to be understood in the game-theoretic sense of signaling games). We can compile the entirety of indications about the adversary's activity into a weight $\Pr(\theta) \neq 1/|AS_2|$ for location θ to possibly start from. Higher values may be assigned where we have stronger indication of recent adversarial activity at location θ.

8. Conclusions and outlook

Cut-The-Rope has been designed for ease of use in applications with little information (such as exemplified with the MARA robot use case), but also situations where there is detailed information encoded in the attack graph (such as for the MiR100 use case), or even when adversarial indications are available from auxiliary security systems, such as intrusion detection or others (including is possible as outlined in Section 7.5). The accuracy hence depends on how much information we can bring into the game, being a rather crude approximation for MARA, but much more fine-grained for the MiR100 use case. In both cases, however, the defender gets nontrivial advice on where to allocate its typically scarce resources for a best defense, beyond just guarding a graph-cut or choke point towards the critical asset v_0. When there are several such (evident) critical regions in the network to defend, different choke points may be of different criticality, depending on how many attack scenarios (each executed by another avatar in Cut-The-Rope) actually make use of this area in the attack graph.

The results obtained show some limitations for the practical use. First, and most substantially, the results depend on the ordering of the nodes, and – in addition to the general non-uniqueness of equilibria – hence may be ambiguous for the defender, leaving a residual chance of there being other defense possibilities. The algorithms applied in this work give only one solution, among perhaps many others. Second, the setting of probabilities from CVE, CVSS or likewise annotations is a nontrivial matter on its own, with only few first steps towards a systematic and sound derivation available in the literature (König et al., 2018). The assumption of invisibility of the intruder can perhaps be weakened by including signals from intrusion detection or other side-information in the defender's policy. The model simulated here does not include this possibility. Finally, the condensation of several attack goals into a single target node comes with the price of losing accuracy and some information about which attack goal may be more likely to be reached, thus making multi criteria optimization an interesting generalization to study.

Generally, Cut-The-Rope opens up an interesting class of games of mixed timing of moves between the actors, unlike as in extensive or normal form games, where players usually take actions in a fixed order. Likewise, and also different to many other game models, Cut-The-Rope has no defined start or finish for the defender ("security is never done"), while only one of the players knows when the game starts and ends, and the attacker can send its avatars from all possible locations in the network. The model is thus complementary to FlipIt, while it allows the attacker to spend any amount of time in the system, as long as the vital asset remains out of reach. This is actually to reflect the reality of security management: we cannot keep the adversary out, we can only try keeping him as far away as possible.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Jasmin Wachter reports financial support was provided by Alpen-Adria University. Stefan Rass reports article publishing charges was provided by Johannes Kepler University Linz. Victor Mayoral-Vilches reports a relationship with Alias Robotics that includes: employment.

CRediT authorship contribution statement

Stefan Rass: Conceptualization, Methodology, Writing – original draft. **Sandra König:** Conceptualization, Methodology, Software. **Jasmin Wachter:** Investigation, Writing – review & editing. **Víctor Mayoral-Vilches:** Data curation, Resources, Validation. **Emmanouil Panaousis:** Validation, Writing – review & editing.

Data availability

Data will be made available on request.

Acknowledgment

We thank the anonymous reviewers for their very useful comments and pointers to additional thoughts on future work and various pointers that helped to improve this manuscript. This work was supported by the Karl-Popper-Kolleg SEEROSE (Responsible, Safe, and Secure Robotic Systems Engineering) at the University of Klagenfurt.

Appendix A. Analytic results

For a plausibility assessment of the game's results, we analytically study the results on attack graphs with no information at all, so that we can intuitively (and without any model) determine a best defense. The analytic results to follow are consistent with the numeric findings for MARA, and, more importantly, are independent of the attack rate λ (see Proposition 5).

Given a set AS_2 of attack paths, the defender's best strategy in Cut-The-Rope is to find and guard a minimal (in a sense to be defined later) graph-theoretic cut $C \subseteq V$ between the starting node (set) of all attack paths, and the target asset v_0. Towards proving this claim, suppose that the defender would focus on a set $S \subset V$ that is not a cut. Then, there is a path π that bypasses S, i.e., $S \cap V(\pi) = \emptyset$, but this makes π a winning strategy for the attacker (since the defender can never catch the attacker on that route). Suppose that the defender's guarded set C were not minimal, i.e., the defender spot-checks on a superset $S' \supset C$. Then, we can distinguish two cases:

1. either no attack path passes through the nodes $S' \setminus C$, in which case defending them is useless, and hence defending S' is a sub-optimal strategy (as it consumes too many resources), or
2. there is at least one attack path through a node in $S' \setminus C$ and another node in C. In that case, we can safely remove either of the two, since both would cut the rope in the sense we desire. This strictly shrinks S', and we can repeat this reduction until the resulting set has become minimal (in terms of cardinality).

Compiling the thoughts above concludes the proof of the next result:

Lemma 2. *Let s be the root of the attack graph, and let v_0 be the target asset. Furthermore, assume that the defender can cut the rope anywhere in the graph, except at the starting point and the target (to avoid trivialities). The optimal strategy of defense in Cut-The-Rope is*

guarding an $s - v_0$-cut of minimal cardinality. If there is more than one starting point or more than one target asset, the cut is understood between the respective sets thereof.

Lemma 2 makes no assertion about what cut to choose if there are several. For example, if we have only one attack path overall, then every node on it would be a valid cut. Intuitively, the best option is cutting the (single) rope as close as possible near v_0, in order to get the most likely locations covered from which an attacker's avatar could start. The proof of Lemma 3 makes this rigorous:

Lemma 3. *Assume that a defender's (mixed) strategy prescribes to spot-check on the attack path π. The best point to cut the rope is the location v whose distance to v_0 along the path π is minimal.*

Proof. Consider the attack path π as a sequence of consecutive vertices $(u_0, u_1, u_2, \ldots, u_l = v_0)$, and write $V(\pi)$ to mean the set of all vertices on π. Call $c \in V(\pi)$ the vertex whose distance $d(c, v_0)$ is minimal among all $V(\pi) \cap AS_1$, i.e., all nodes on π that the defender has in its action set AS_1 and can hence spot-check. Let $c' \in V(\pi) \cap AS_1$ be another node to possibly check on the same path, which is distinct from c. It follows that either there is a connection $c' \to c$ (if the two are consecutive) or there is at least one node in between $c' \to \cdots \to c$. In either case, we have distinct avatars $\theta_{c'}$ and θ_c, corresponding to these two nodes as starting points. Both use the same distribution F_N with probability mass function f_N, for the number N of steps taken forward on π, only starting at different locations (c or c' hereafter). To ease notation in the following, let us associate the avatar θ directly with a node on π (this creates no ambiguities).

The probability mass that an avatar θ_i puts on v_0 when starting from location i is given by the chances to take at least the residual distance $d_\pi(\theta, v_0)$ from the starting point (θ) until v_0. Given the distribution function F_N of the random distance overcome upon adversarial activity, this is $\Pr(N \geq d(\theta, v_0)) = 1 - F_N(d_\pi(\theta, v_0))$. Throughout the rest, π and v_0 will both be fixed, so we can safely omit them from our notation, so let us write $\Delta_\theta := d_\pi(\theta, v_0)$, for the residual distance on the path π between the avatar starting from θ, and the target v_0. Moreover, put $u_\theta := \Pr(N \geq \Delta_\theta) = 1 - F_N(\Delta_\theta) = \sum_{d \geq \Delta_\theta} f_N(d_\pi(\theta, v_0))$ to abbreviate the probability of the attacker to reach v_0 within the next move.

The utility over all attacker avatars is then

$$\Pr(\text{asset } v_0 \text{ is lost to the attacker}) = \sum_{\theta \in \Theta} \Pr(\theta) \cdot u_\theta \tag{A.1}$$

which is the total probability mass assigned to v_0 by all adversary avatars.

Now, let us compare the effects of spot-checking c vs. spot-checking c' that is farther away from v_0. Since we have only the attack path π on which c' comes before c, let us break up the path into three corresponding parts $\pi = (u_0, \ldots, c' = u_i, \ldots, c = u_j, \ldots, u_l = v_0)$, and expand (A.1) accordingly

$$\sum_{\theta \in \Theta} \Pr(\theta) \cdot u_\theta = \sum_{\theta \in (u_0, \ldots, u_i = c')} \Pr(\theta) \cdot u_\theta \tag{A.2}$$

$$+ \sum_{\theta \in (u_{i+1}, \ldots, u_j = c)} \Pr(\theta) \cdot u_\theta \tag{A.3}$$

$$+ \sum_{\theta \in (u_{j+1}, \ldots, u_\ell = v_0)} \Pr(\theta) \cdot u_\theta . \tag{A.4}$$

It will be helpful to remember the effect of truncating a distribution at t, which is switching from $F_N(d)$ to the conditional distribution on $F_N(d | d \leq t)$, whose density is

$$f_N(d | d \leq t) = \begin{cases} \frac{f_N(d)}{F_N(t)}, & \text{if } d \leq t; \\ 0, & \text{otherwise.} \end{cases} \tag{A.5}$$

The important fact is that cutting at some point on the path affects all avatars on the segment from the beginning node until the cut node c or c'. If we cut at c', we take out the whole expression (A.2), leaving (A.1) = (A.3) + (A.4), in a slight abuse of formalism here. However, if we cut at c, term (A.3) also drops out of (A.1), leaving this to be the better option for the defender. □

Now, we can compile the findings so far into a generic characterization of the defender's best choice:

Proposition 5. *Let an acyclic attack graph G be with root node u_0, and let v_0 be the target node (likewise, for sets thereof if there are multiple). Furthermore, let d be a distance measure in G. The defender's optimal strategy in* Cut-The-Rope *is spot-checking a minimum-cardinality $u_0 - v_0$-cut C, with the property that for each $c \in C$, the distance $d(c, v_0)$ is minimal.*

A1. Consistency of numeric and analytic results

The numeric findings for the MARA use case agree with the analytic predictions to defend the graph cut that is closest to the target nodes. The formal arguments above assume the same distribution for all possible paths, which does not hold for the MiR100 use case. Thus, the optimal defense no longer needs to be a graph cut, and the numeric results about the MiR100 use case confirm this possibility. Since in the MiR100 case, the attack paths have different efficacies, strategic dominance among the attack paths may affect the results accordingly. Since the results, in this more general case, depend on the distribution conditional on the attack path, it appears unlikely that comparable analytic predictions can be made for the movement pattern of Section 3.4, and we leave this as an open problem.

Regarding the heuristic defense, its bad performance in comparison to Cut-The-Rope can be attributed to the defender blindly checking everywhere on the attack graph, while the intuition (also behind the formal arguments here) would rather advise to defend closer to the goal. This suggests that the optimization that Cut-The-Rope may be reasonably replaced by a heuristic defense, only focused on a graph cut subset of nodes, and indeed, the numbers for the MARA use case show an approximately uniform defense of nodes on such a cut to be optimal. Overall, however, it is advisable to run an optimization, since just adding the analytic prediction of where to defend to the heuristic is incorrect in the case where the traversal of an attack path depends on the path's properties, such as distinct difficulties to exploit, as in the MiR100 use case. Here, the performance of the defense is substantially better than for the heuristic, but the apparent focus on a graph cut is not found in the results.

Appendix B. Derivation of the probability (13)

The density of the $\mathcal{E}rl(n, \lambda)$ distribution family is for $x \geq 0$ given by $f_{\mathcal{E}rl(n,\lambda)}(x) = \frac{\lambda^n x^{n-1}}{(n-1)!} e^{-\lambda x}$ and $f(x) = 0$ for $x < 0$. The density of the exponential distribution is a special case thereof, $f_{\mathcal{E}xp(\lambda)}(x) = f_{\mathcal{E}rl(1,\lambda)}(x)$. Abbreviating the total time as $T = T_1 + T_2 + \ldots + T_n$, with all i.i.d. summands $T_i \sim \mathcal{E}xp(\lambda_\pi)$, we are interested in whether $T \leq W \iff Z := W - T \geq 0$. The case $Z = z$ for $z \in \mathbb{R}$ occurs if and only if $W = t + z$ and $T = t$ for any $t \in \mathbb{R}$, and we get the convolution-like integral for the density of $T - W$ as

$$f_{T-W}(z) = \int_{-\infty}^{\infty} f_{\mathcal{E}rl(\lambda_\pi, n)}(t) f_{\mathcal{E}rl(1,\lambda_D)}(t + z) dt .$$

We are, however, only interested in the probability $p = \Pr(T - W \geq 0)$, which adds a second integral to get the quantity of interest

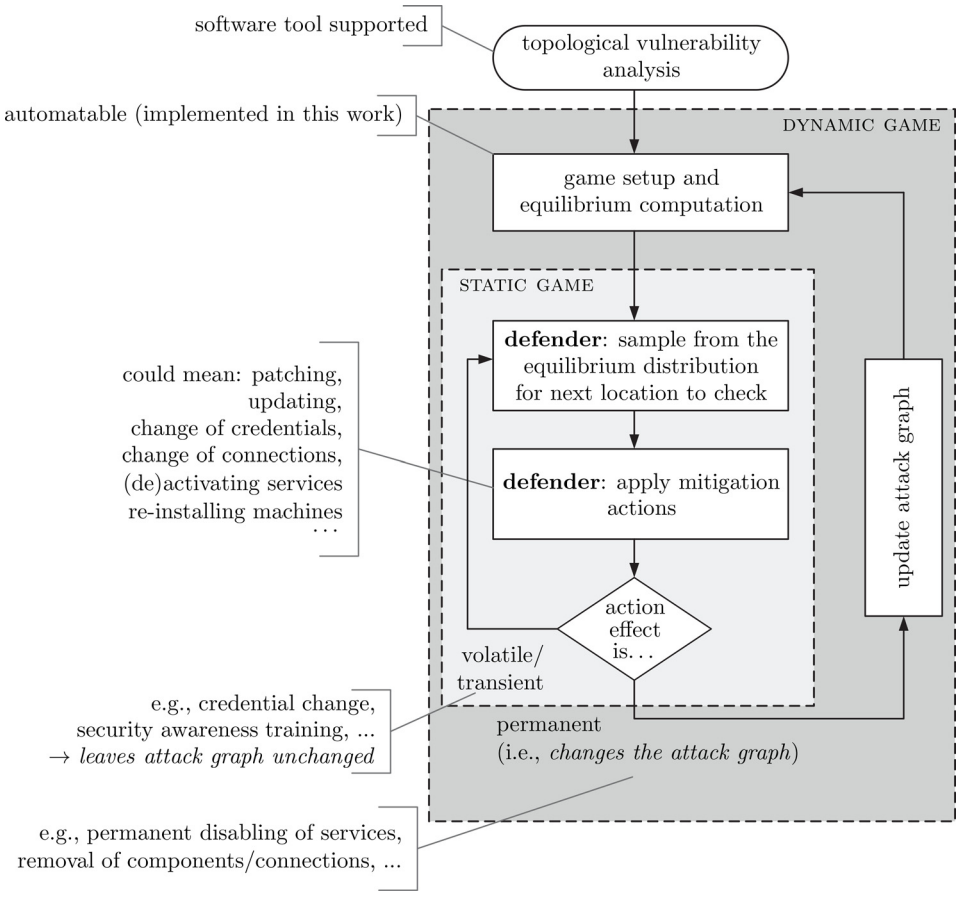


Fig. C.7. Cut-The-Rope (static game) inside the continuous process of permanent system hardening (dynamic game).

$$p = \int_0^\infty f_{T-W}(z)dz$$
$$= \int_0^\infty \int_{-\infty}^\infty f_{\mathcal{E}rl(\lambda_\pi,n)}(t) f_{\mathcal{E}rl(\lambda_D,1)}(t+z)dtdz.$$

A bit unexpectedly, the double integral makes things easier to evaluate here, since we can swap the order of integration (by the Fubini-Tonello theorem), to get

$$p = \int_{-\infty}^\infty \int_0^\infty \underbrace{f_{\mathcal{E}rl(\lambda_\pi,n)}(t)}_{\text{const. w.r.t. } z} f_{\mathcal{E}rl(\lambda_D,1)}(t+z)dzdt$$
$$= \int_{-\infty}^\infty f_{\mathcal{E}rl(\lambda_\pi,n)}(t) \underbrace{\int_0^\infty f_{\mathcal{E}rl(\lambda_D,1)}(t+z)dz}_{=e^{-\lambda t}} dt$$
$$= \int_{-\infty}^\infty \underbrace{f_{\mathcal{E}rl(\lambda_\pi,n)}(t)}_{=0 \text{ for } t<0} e^{-\lambda t} dt = \int_0^\infty f_{\mathcal{E}rl(\lambda_\pi,n)}(t) e^{-\lambda t} dt$$
$$= \left(\frac{\lambda_\pi}{\lambda_\pi + \lambda_D}\right)^n$$

Appendix C. Application for risk control

Actions with a *permanent effect* change the attack surface by blocking certain paths, increasing the attack detection capabilities, or similar. Examples include the installation of a firewall, malware scanners, deactivation of services or accounts, and many more.

If the defender's action space includes at least one with potentially permanent effect, the attack graph, and hence the overall game, *changes* with the defender's activity, and the game must be re-instantiated before the next round after *pruning the attack graph*. This turns Cut-The-Rope into a *dynamic* game, but it is still repeated with infinite time horizon. It is fair to remark that the tree may not only become pruned, but introduce new attack paths upon inserting new components, installing new software or similar.

In both cases, the setup of the game may (but does not need to) start from the results of a topological vulnerability analysis, with repetitions being either from the existing defense equilibrium strategy (static instance) or including the re-instantiation and equilibrium computation (dynamic instance); see Fig. C.7 for a flowchart-like presentation.

References

AcutronicRobotics. Threat Model analysis for MARA robot. 2021. Original-date: 2019-04-07T15:53:00Z.

ADAPT: Analytical Framework for Actionable Defense against Advanced Persistent Threats | UW Department of Electrical & Computer Engineering. 2018. https://www.ece.uw.edu/projects/adapt-analytical-framework-for-actionable-defense-against-advanced-persistent-threats/.

Alpcan, T., Basar, T., 2010. Network Security: A Decision and Game Theoretic Approach. Cambridge University Press.

alias Robotics. Case Study - threat modeling a ROS2 robot. 2019. https://aliasrobotics.com/case-study-threat-model-mara.php.

alias Robotics. The Week of Mobile Industrial Robots's bugs. 2020. https://news.aliasrobotics.com/the-week-of-mobile-industrial-robots-bugs/.

Alias Robotics. Case Study - penetration testing Mobile Industrial Robots. 2021. https://aliasrobotics.com/case-study-pentesting-mir.php.

Cao, C., Yuan, L.P., Singhal, A., Liu, P., Sun, X., Zhu, S., 2018. Assessing attack impact on business processes by interconnecting attack graphs and entity dependency graphs. In: Kerschbaum, F., Paraboschi, S. (Eds.), Data and Applications Security and Privacy XXXII, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 330–348. doi:10.1007/978-3-319-95729-6_21.

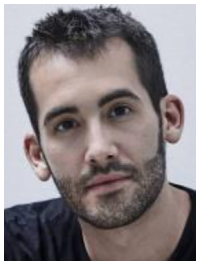
Carroll, T.E., Grosu, D., 2009. A game theoretic investigation of deception in network security. In: 2009 Proceedings of 18th International Conference on Computer

Communications and Networks. IEEE, San Francisco, CA, USA, pp. 1–6. doi:10.1109/ICCCN.2009.5235344.
Etesami, S.R., Basar, T., 2019. Dynamic games in cyber-physical security: an overview. Dyn. Games Appl. doi:10.1007/s13235-018-00291-y.
Fang P., Gao P., Liu C., Ayday E., Jee K., Wang T., Ye Y.F., Liu Z., Xiao X.. {Back-Propagating} System Dependency Impact for Attack Investigation. 2022. 2461–2478. https://www.usenix.org/conference/usenixsecurity22/presentation/fang.
Fang, X., Zhai, L., Jia, Z., Bai, W., 2014. A game model for predicting the attack path of APT. In: 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing. IEEE, Dalian, China, pp. 491–495. doi:10.1109/DASC.2014.94.
Fudenberg, D., Tirole, J., 1991. Game Theory. MIT Press.
Hota, A.R., Clements, A.A., Bagchi, S., Sundaram, S., 2018. A game-theoretic framework for securing interdependent assets in networks. In: Rass, S., Schauer, S. (Eds.), Game Theory for Security and Risk Management. In: Static & Dynamic Game Theory: Foundations & Applications. Springer, pp. 157–184.
Hota, A.R., Clements, A.A., Sundaram, S., Bagchi, S., 2016. Optimal and game-theoretic deployment of security investments in interdependent assets. In: Decision and Game Theory for Security. In: LNCS, vol. 9996. Springer, pp. 101–113.
Hota, A.R., Sundaram, S., 2018. Interdependent security games on networks under behavioral probability weighting. IEEE Trans. Control Netw. Syst. 5, 262–273. doi:10.1109/TCNS.2016.2600484.
Huang L., Zhu Q.. Adaptive Strategic Cyber Defense for Advanced Persistent Threats in Critical Infrastructure Networks. arXiv:180902227 [cs]2018; arXiv:1809.02227
Jajodia, S., Noel, S., Kalapa, P., Albanese, M., Williams, J., 2011. Cauldron mission-centric cyber situational awareness with defense in depth. In: 2011 - MILCOM 2011 Military Communications Conference. IEEE, pp. 1339–1344. doi:10.1109/MILCOM.2011.6127490.
Jha, S., Sheyner, O., Wing, J., 2002. Two formal analyses of attack graphs. In: Proceedings 15th IEEE Computer Security Foundations Workshop. CSFW-15, pp. 49–63. doi:10.1109/CSFW.2002.1021806. iSSN: 1063-6900
Kamhoua, C.A., Leslie, N.O., Weisman, M.J., 2018. Game theoretic modeling of advanced persistent threat in internet of things. J. Cyber Secur. Inf. Syst. 6, 40–46.
Khouzani, M., Sarkar, S., Altman, E., 2012. Saddle-point strategies in malware attack. IEEE J. Sel. Areas Commun. 30, 31–43. doi:10.1109/JSAC.2012.120104.
König, S., Gouglidis, A., Green, B., Solar, A., 2018. Assessing the Impact of Malware Attacks in Utility Networks. Springer, pp. 335–351. doi:10.1007/978-3-319-75268-6_14.
Kushner, D., 2013. The real story of stuxnet. IEEE Spectr. 50, 48–53. doi:10.1109/MSPEC.2013.6471059.
La, Q.D., Quek, T.Q.S., Lee, J., 2016. A game theoretic model for enabling honeypots in IoT networks. In: 2016 IEEE International Conference on Communications (ICC). IEEE, Kuala Lumpur, Malaysia, pp. 1–6. doi:10.1109/ICC.2016.7510833.
Lallie, H.S., Debattista, K., Bal, J., 2020. A review of attack graph and attack tree visual syntax in cyber security. Comput. Sci. Rev. 35, 100219. doi:10.1016/j.cosrev.2019.100219.
Li, Y., Dai, W., Bai, J., Gan, X., Wang, J., Wang, X., 2019. An intelligence-driven security-aware defense mechanism for advanced persistent threats. IEEE Trans. Inf. Forensics Secur. 14, 646–661. doi:10.1109/TIFS.2018.2847671.
Lin, J., Liu, P., Jing, J., 2012. Using signaling games to model the multi-step attack-defense scenarios on confidentiality. In: Grossklags, J., Walrand, J. (Eds.), Decision and Game Theory for Security. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 118–137.
Lye, K.w., Wing, J.M., 2005. Game strategies in network security. Int. J. Inf. Secur. 4, 71–86. doi:10.1007/s10207-004-0060-x.
Mehta, V., Bartzis, C., Zhu, H., Clarke, E., Wing, J., 2006. Ranking attack graphs. In: Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Rangan, C.P., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M.Y., Weikum, G., Zamboni, D., Kruegel, C. (Eds.), Recent Advances in Intrusion Detection. In: Lecture Notes in Computer Science, vol. 4219. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 127–144. doi:10.1007/11856214_7.
Moothedath S., Sahabandu D., Allen J., Clark A., Bushnell L., Lee W., Poovendran R.. A Game Theoretic Approach for Dynamic Information Flow Tracking to Detect Multi-Stage Advanced Persistent Threats. arXiv:181105622 [cs]2018; arXiv:1811.05622
Panaousis, E., Karapistoli, E., Elsemary, H., Alpcan, T., Khuzani, M., Economides, A.A., 2017. Game theoretic path selection to support security in device-to-device communications. Ad Hoc Netw. 56, 28–42. doi:10.1016/j.adhoc.2016.11.008.
Pawlick, J., Chen, J., Zhu, Q., 2019. iSTRICT: an interdependent strategic trust mechanism for the cloud-enabled internet of controlled things. IEEE Trans. Inf. Forensics Secur. 14, 1654–1669. doi:10.1109/TIFS.2018.2883272.
Pawlick, J., Zhu, Q., 2017. Strategic trust in cloud-enabled cyber-physical systems with an application to glucose control. IEEE Trans. Inf. Forensics Secur. 12, 2906–2919. doi:10.1109/TIFS.2017.2725224.
Qing, H., Shichao, L., Zhiqiang, S., Limin, S., Liang, X., 2017. Advanced persistent threats detection game with expert system for cloud. J. Comput. Res. Dev. 54, 2344. doi:10.7544/issn1000-1239.2017.20170433.
Rass, S., König, S., Panaousis, E., 2019. Cut-the-rope: a game of stealthy intrusion. In: Decision and Game Theory for Security. Springer LNCS 11836, pp. 404–416.
Rass, S., Schauer, S., König, S., Zhu, Q., 2020. Cyber-Security in Critical Infrastructures: A Game-Theoretic Approach. SpringerNature.
Rass, S., Zhu, Q., 2016. GADAPT: a sequential game-theoretic framework for designing defense-in-depth strategies against advanced persistent threats. In: Decision and Game Theory for Security. In: LNCS, vol. 9996. Springer, pp. 314–326. doi:10.1007/978-3-319-47413-7_18.
Senejohnny, D., Tesi, P., De Persis, C., 2018. A jamming-resilient algorithm for self-triggered network coordination. IEEE Trans Control Netw. Syst. 5, 981–990. doi:10.1109/TCNS.2017.2668901.
Shandilya, V., Simmons, C.B., Shiva, S., 2014. Use of attack graphs in security systems. J. Comput. Netw. Commun. 2014, 1–13. doi:10.1155/2014/818957.
Singhal, A., Ou, X., 2011. Security Risk Analysis of Enterprise Networks Using Probabilistic Attack Graphs. Technical Report NIST IR 7788. National Institute of Standards and Technology doi:10.6028/NIST.IR.7788.
Soltan, S., Yannakakis, M., Zussman, G., 2019. REACT to cyber attacks on power grids. IEEE Trans. Netw. Sci. Eng. 6, 459–473. doi:10.1109/TNSE.2018.2837894.
Touhiduzzaman, M., Hahn, A., Srivastava, A.K., 2019. A diversity-based substation cyber defense strategy utilizing coloring games. IEEE Trans. Smart Grid 10, 5405–5415. doi:10.1109/TSG.2018.2881672.
van Dijk, M., Juels, A., Oprea, A., Rivest, R.L., 2013. FlipIt: the game of "stealthy takeover. J. Cryptol. 26, 655–713. doi:10.1007/s00145-012-9134-5.
Wang, C., Du, N., Yang, H., 2012. Generation and analysis of attack graphs. Procedia Eng. 29, 4053–4057. doi:10.1016/j.proeng.2012.01.618.
Wang, L., Jajodia, S., Singhal, A., Cheng, P., Noel, S., 2014. k-Zero day safety: a network security metric for measuring the risk of unknown vulnerabilities. IEEE Trans. Dependable Secure Comput. 11, 30–44. doi:10.1109/TDSC.2013.24.
Xiao, L., Xu, D., Mandayam, N.B., Poor, H.V., 2018. Attacker-centric view of a detection game against advanced persistent threats. IEEE Trans. Mob. Comput. 17, 2512–2523. doi:10.1109/TMC.2018.2814052.
Yang, L.X., Li, P., Yang, X., Tang, Y., 2018. A risk management approach to defending against the advanced persistent threat. IEEE Trans. Dependable Secure Comput. 1. doi:10.1109/TDSC.2018.2858786.
Yang, L.X., Li, P., Zhang, Y., Yang, X., Xiang, Y., Zhou, W., 2019. Effective repair strategy against advanced persistent threat: a differential game approach. IEEE Trans. Inf. Forensics Secur. 14, 1713–1728. doi:10.1109/TIFS.2018.2885251.
Yuan, H., Xia, Y., Zhang, J., Yang, H., Mahmoud, M., 2019. Stackelberg-game-based defense analysis against advanced persistent threats on cloud control system. IEEE Trans. Ind. Inform. 1. doi:10.1109/TII.2019.2925035.
Zeng, J., Wu, S., Chen, Y., Zeng, R., Wu, C., 2019. Survey of attack graph analysis methods from the perspective of data and knowledge processing. Secur. Commun. Netw. 2019, e2031063. doi:10.1155/2019/2031063. Publisher: Hindawi
Zeng, P., Lin, G., Pan, L., Tai, Y., Zhang, J., 2020. Software vulnerability analysis and discovery using deep learning techniques: a survey. IEEE Access 8, 197158–197172. doi:10.1109/ACCESS.2020.3034766.
Zenitani, K., 2023. Attack graph analysis: an explanatory guide. Comput. Secur. 126, 103081. doi:10.1016/j.cose.2022.103081.
Zhang, R., Zhu, Q., 2019. FlipIn: a game-theoretic cyber insurance framework for incentive-compatible cyber risk management of internet of things. IEEE Trans. Inf. Forensics Secur. 1. doi:10.1109/TIFS.2019.2955891.
Zhu, Q., Rass, S., 2018. On multi-phase and multi-stage game-theoretic modeling of advanced persistent threats. IEEE Access 6, 13958–13971. doi:10.1109/access.2018.2814481.

Stefan Rass holds degrees in mathematics and computer science from the Universitaet Klagenfurt (AAU). His research interests cover decision theory and gametheory with applications in system security, especially robotics security, as well as complexity theory, statistics, and information-theoretic security. He authored numerous papers related to practical security, security infrastructures, robot security, and applied statistics and decision theory in security. He participated in various nationally and internationally funded research projects, as well as being a contributing researcher in many EU projects and offering consultancy services to the industry. Currently, he is a full professor at the Johannes Kepler University Linz, Austria, as a member of the Secure and Correct Systems Lab.

Sandra König is a researcher at the Austrian Institute of Technology in the Center for Digital Safety & Security. She received her bachelor's and master's degree in mathematics at ETH Zurich and her Ph.D. with distinction in Mathematics at Alpen Adria University Klagenfurt. Her core focus lies on probabilistic risk models and game theoretic risk mitigation methods. She has been involved in several Austrian and EU projects dealing with protection of critical infrastructures as well as safety and security of autonomous systems. Beyond this, she is interested in application of machine learning techniques and domains such as logistics. She is a lecturer at the University of Zurich.

Jasmin Wachter is a Senior Scientist at the Alpen-Adria University Klagenfurt (AAU). She graduated with a master's degree in mathematics and worked as a researcher in the robotics as well as security domain in various security related projects. Currently Ms. Wachter is pursuing her Ph.D. degree in Computer Science investigating incentive-based security engineering using game-theory. Her main research interests cover game-theory with applications in system security, safety and security of autonomous systems, as well as statistics and data science.



Víctor Mayoral-Vilches is a roboticist and has a strong technical background and is one of the top experts globally on ROS 2. He spent the last 10 years building robots. Founded, funded and led 4 robotics startups. Experience leading research initiatives and projects in the fields of robotics, cybersecurity and artificial intelligence. Víctor is also a Ph.D. candidate at the System Security Group, Universität Klagenfurt, researching in the area of robot cybersecurity.

Emmanouil (Manos) Panaousis is a Professor of Cyber Security and Head of Cyber Risk at the Internet of Things and Security Centre (ISEC) at the University of Greenwich. He is also a Senior Member of IEEE and author of more than 100 peerreviewed publications including publications in top scientific journals. He is most known for his contributions in the field of decision support for cyber security.