# Cut-The-Rope: A Game of Stealthy Intrusion

Stefan Rass[1][0000−0003−2821−2489], Sandra König[2][0000−0000−0000−0000], and Emmanouil Panaousis[3][0000−0001−7306−4062]

[1] Universitaet Klagenfurt, Institute of Applied Informatics, Austria
[2] Austrian Institute of Technology, Center for Digital Safety & Security, Austria
[3] Department of Computer Science, University of Surrey, UK
stefan.rass@aau.at, sandra.koenig@ait.ac.at, e.panaousis@surrey.ac.uk

**Abstract.** A major characteristic of Advanced Persistent Threats (APTs) is their stealthiness over a possibly long period, during which the victim system is being penetrated and prepared for the finishing blow. We model an APT as a game played on an attack graph $G$, and consider the following interaction pattern: the attacker chooses a path in $G$, and step-by-step works its way towards the goal by repeated penetrations. In each step, it leaves a backdoor for an easy return to learn how to accomplish the next step. We call this return path the "rope". The defender's duty is "cutting" this rope by cleaning the system from (even unknown) backdoors, e.g., by patching or changing configurations. While the defender is doing so in fixed intervals governed by working hours/shifts, the attacker is allowed to take any number of moves at any point in time. The game is thus repeated, i.e., in *discrete time*, only for the defender, while the second player (adversary) moves in *continuous time*. It also has asymmetric information, since the adversary is stealthy at all times, until the damage causing phase of the APT. The payoff in the game is the attacker's chance to reach this final stage, while the defender's goal is minimizing this likelihood (risk). We illustrate the model by a numerical example and open access implementation in `R`.

**Keywords:** Advanced persistent threats · Security · Cyber defense · Cyber physical system · Attack graph · Attack tree.

## 1 Introduction

Contemporary APTs exhibit some similarities to human diseases: there is a phase of infection (where the attacker makes the initial contact, e.g., by sending a successful spam or phishing email), a phase of incubation (where the attacker penetrates the system as deep as it can; often slowly and stealthy to avoid detection), and a phase of outbreak (where the attacker causes the actual damage). The game proposed in this work covers the incubation phase, letting the defender, similar to the human body's immune system, taking actions to keep the adversary away from vital assets, even without knowing explicitly about its moves, location or even presence.

The playground of our penetration game is an attack graph, such as obtained from a topological vulnerability analysis (see, e.g., [10]). We adopt an example

from the literature to illustrate our game thereon. The game is hereby designed for ease of application, to account for the expected large diversity of infrastructures on which CutTheRope is playable; nonetheless, the treatment is novel in two aspects:

- there is no natural synchronicity in the players taking actions; Particularly, we have a defender that acts in rounds, as working days/hours, time shifts or other organisational regulations prescribe, facing an opponent that can act in continuous time, at any time, as often as s/he likes, and in any pattern (independent or adaptive to the defender's actions to learn about the current system configuration (leader-follower style), etc.)
- the goal is *not* minimizing the *time* that an attacker spends in the system, but rather the *chances* for the attacker *to hit vital assets* (no matter how long it takes, or attempting to keep it completely outside the system).

The second point distinguishes CutTheRope from related games like FlipIt [5], based on the recognition that even a very short access time to a critical asset may suffice to cause huge damage. For example, the cooling system of a nuclear power plant could be shut down within a short period of time, causing an unstoppable chain reaction. On the contrary, the attacker may spend an arbitrary lot of time with a honeypot, where no damage is possible. Thus, the average or total time spent in a system is not necessarily what counts; what is important is the adversary's chance to use its time (no matter how short) to cause damage. Consequently, CutTheRope is about minimizing the adversary's odds to reach a critical area, rather than to keep it out completely or to minimize its time of having parts of the system under control.

**Related Work**. APTs, due to their diverse combination of attacks, hardly admit a single model to capture them; rather, they call for a combination of models tailored to different aspects or characteristics of the attack. The common skeleton identified for "the general" APT incurs the three above mentioned phases, but can be refined into what is called the *kill chain* [11], consisting of *reconnaissance*, *exploit*, *command & control*, *priviledge escalation*, *lateral movement* and *objective/target*, in the sequential order just given. A proper defense aligns with these phases, and most related work [6] is specific for at least one of them. Notable is the ADAPT project [1], covering a wide spectrum of aspects and phases. Specific defense models include the detection of spying activities [17], tracing information flows [16], detection of malware [12], deception [4] (also via honeypots [13]), attack path prediction [7], and general network defense [2] to name only a few. Our game is in a way similar to that of [15], 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 simpler way in our model.

Taking the APT as a long term yet one-shot event, an attack tree can be treated as a (big) game in extensive form. In this 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 [9]. 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 [24,27].

A different classification of related work is based on the protection goals. Defenses can be optimized for confidentiality [14], the monetary value of some asset upon theft or damage [27], or the time that an adversary has parts of the system under control [5]. 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 more down on the list.

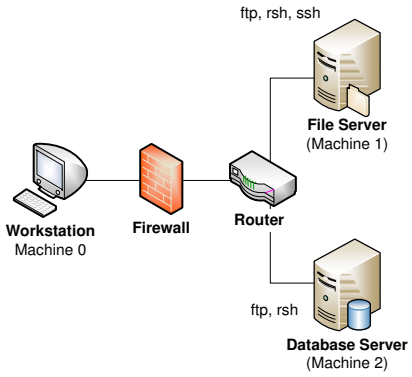
## 2 The Model

The story evolves around a defender seeking to protect some asset from a stealthy intruder. To this end, the defender maintains an attack graph on which it engages in a game to keep the attacker away from the asset. It does so by taking turns periodically, doing spot-checks on randomly chosen nodes in the graph. A spot check hereby can mean various things, such as a mere change of credentials, a malware check, but also more complex operations such as the deactivation of services or a complete reset or reinstallation of the respective computer from a clean (trusted) reference image. Not all these options may be open for all nodes, e.g., the defender may not be allowed to deactivate certain services (like a secure shell), or a reinstallation may cause undesirably high costs due to the temporal outage of the node (see [23] for a game model including this aspect).

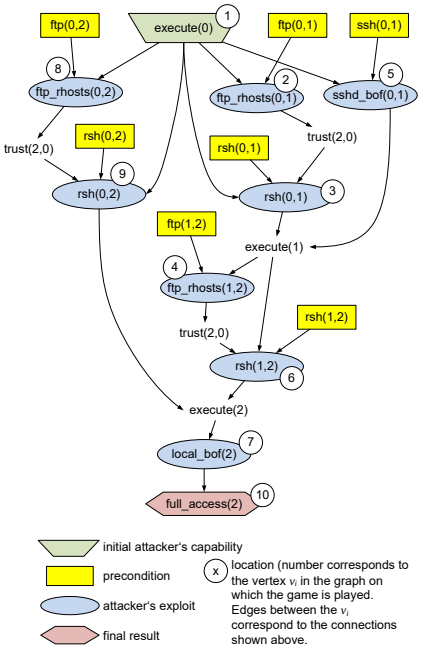
### 2.1 A Running Example

Let us illustrate CutTheRope using an example attack graph shown in Figure 1b, computed from a topological vulnerability analysis in the infrastructure as shown in Figure 1a. In the (simplified) instance of CutTheRope described next, the devices have no distinct resilience against penetration. That is, the attacker has equal chances to take any step along the attack path. We can later drop this assumption easily for the price of an only slightly modified implementation of CutTheRope, as we outline in the conclusions Section 4. The attacker works its way from a starting point (not necessarily a fixed position; multiple possibilities for a start are permitted), stepwise towards the asset. It does so by exploiting individual vulnerabilities found at each node along the way. The attack graph (Figure 1b) gives rise to a set of attack paths as listed in Figure 1c. Note that this list is in first place made to be exhaustive, and practically may undergo a cleanup to remove attack paths that are not meaningful.

In our example, each path describes a penetration by a sequence of unary or binary predicates `access(x)` or `protocol(x,y)`, expressing the sort of `access` gained on machine `x` or access to machine `y` by some `protocol` (see Figure 1b for examples, e.g., `rsh(0,1)` means a gain of access to machine 1 from machine 0, via a remote shell exploit). Preconditions (appearing as rectangular boxes in Figure 1b shows) are omitted for simplicity. For each respective next stage along the overall attack, the adversary will come back over its so-far prepared route, to




(a) Infrastructure from [26]




(b) Attack Graph [26]

| No. | Attack path |
|---|---|
| 1 | execute(0) → ftp_rhosts(0,1) → rsh(0,1) → ftp_rhosts(1,2) → rsh(1,2) → local_bof(2) → full_access(2) |
| 2 | execute(0) → ftp_rhosts(0,1) → rsh(0,1) → rsh(1,2) → local_bof(2) → full_access(2) |
| 3 | execute(0) → ftp_rhosts(0,2) → rsh(0,2) → local_bof(2) → full_access(2) |
| 4 | execute(0) → rsh(0,1) → ftp_rhosts(1,2) → sshd_bof(0,1) → rsh(1,2) → local_bof(2) → full_access(2) |
| 5 | execute(0) → rsh(0,1) → rsh(1,2) → local_bof(2) → full_access(2) |
| 6 | execute(0) → rsh(0,2) → local_bof(2) → full_access(2) |
| 7 | execute(0) → sshd_bof(0,1) → ftp_rhosts(1,2) → rsh(0,1) → rsh(1,2) → local_bof(2) → full_access(2) |
| 8 | execute(0) → sshd_bof(0,1) → rsh(1,2) → local_bof(2) → full_access(2) |

(c) Attack paths in the graph shown in Fig 1b

Fig. 1: Example Playground for CutTheRope

inspect the next node for vulnerabilities that can be exploited. We can imagine this path as a "rope" along which the adversary "climbs up to the top", i.e., the target asset. This may even happen unknowingly to the defender, since the attacker is stealthy.

The defender, repeatedly inspecting nodes, may successfully clean the respective node from any adversarial traces, and thus *cuts the adversary's rope*, sending it effectively back to the point immediately before the cut (i.e., the inspected point); hence the game's name CutTheRope. Cutting at a point that the adversary has not yet reached has no impact, since the adversary will always learn the current configuration before mounting a penetration (it will thus never get stuck). For the same reason, the adversary may move laterally towards a different attack path, if there is an easier alternative to reach the target.

### 2.2 Game Definition

An instance of CutTheRope is a tuple $(G, v_0, AS_1, AS_2, \lambda)$ with the following ingredients: $G = (V, E)$ is an attack graph, containing a designated node $v_0 \in V$ as the target for the attack(er). The action set $AS_1 \subseteq V \setminus \{v_0\}$ contains all nodes admissible for spot checking by the defender (excluding $v_0$ to avoid trivialities). The action set $AS_2$ for the attacker contains all attack paths towards $v_0$ in $G$, from one (or several) "entry nodes" in $G$. For a set $\Omega$, we write $\Delta(\Omega)$ to mean the set of all distributions supported on $\Omega$, and we write $X \sim F$ or $X \sim \boldsymbol{x} \in \Delta(\Omega)$ to tell that the random variable has the general distribution $F$ or categorical distribution $\boldsymbol{x}$. We assume $|AS_2|$ to be of manageable size, practically achievable by tool aids to construct the attack graph (often grouping of nodes with similar characteristics regarding vulnerabilities [10,3]). The value $\lambda \in \mathbb{R}^{>0}$ is the *attack rate*: it specifies an average over how many steps the attacker takes in times when the defender is inactive. Note that we assume no particular cost for the attacker to penetrate here, and for simplicity, we further assume that the attacker is always successful in the penetration (i.e., the idle times of the attacker are used for learning about configurations and exploits, and the learning is always successful; in reality, this assumption may be overly pessimistic for the defender, but can be relaxed almost trivially as we discuss in the conclusions section). Also for simplicity, we assume $\lambda$ to be constant over time (the generalization towards a nonhomogeneous attack process is beyond the scope of this work, but an interesting open question).

This implies the basic assumption that a move of the attacker at any time takes it a random (Poissonian) number of $N \sim Pois(\lambda)$ steps further down on the attack path, until the target asset $v_0 \in G$. The *payoff* in the game is the adversary's random location $L \sim U(\boldsymbol{x}, \theta, \lambda)(V)$ that depends on (i) the defender's probability vector $\boldsymbol{x} \in \Delta(AS_1)$ of doing spot checks on the node set $AS_1$, (ii) the attacker's starting point $\theta \in V \setminus \{v_0\}$, and (iii) the attack rate $\lambda$. Since the attacker is stealthy, the defender has no means of knowing where the attacker currently is, i.e., from where it has started to take the next $N$ steps towards $v_0$. Thus, the defender cannot work out the distribution of $L$, and can only choose its own randomized spot checking rule $\boldsymbol{x} \in \Delta(AS_1)$ with knowledge

of $\lambda$. We model the defender's uncertainty about the adversary's location by considering each starting point as inducing a distinct adversary *type* $\theta \in \Theta$, where $\Theta \subset V \setminus \{v_0\}$ is the set of possible starting locations ($\theta \neq v_0$ is assumed to avoid trivialities). This turns the competition into a *Bayesian game* where the defender faces an adversary of unknown type (location) from the finite set $\Theta$. The game, however, does not qualify as a signalling game, since the attacker remains invisible at all times (until it reaches $v_0$). Still, a perfect Bayesian equilibrium (Definition 1) will turn out as a suitable solution concept.

*Remark 1.* The value of $\lambda$ is common knowledge of both parties, but realistically an assumption made by the defender. As such, it may be subject to estimation errors, and may (in reality) be a presumed range of possible values rather than a fixed value. The implied change to the model, however, amounts to a mere change of the resulting distribution from plain $Pois(\lambda)$ into something more complex (e.g., a mix of Poisson distributions or other), but nonetheless a distribution over the number of steps being taken. Likewise, assuming a fixed number of attacker steps at any point in (continuous) time yet is describable by yet another distribution over the step number within a fixed time interval. Our model does not anyhow hinge on the shape of the step's distribution, so both generalizations are left for reports on future (practical) instances of CutTheRope.

For each attack path $\pi \in AS_2$, starting from the location $\theta$, the attack step number distribution, here $Pois(\lambda)$, assigns probability masses to the nodes on $\pi$. The totality of all attack paths then defines a total probability mass on each node of $G = (V, E)$, which is the distribution $U$ (later made more explicit in the derivation of expressions (2) and (3)). We put the nodes in $V$ in *ascending order* of (graph-theoretic) distance to $v_0$ (with any order on nodes of equal distance). Then, the mass assigned in the proximity of $v_0$ is the *tail mass* of $U$.

We stress that replacing the uncertain payoffs by numbers, i.e., taking the mere expected payoff is *not meaningful*, since we are interested in the *attacker not hitting the target*, but do not care about its average penetration depth. The latter is uninteresting, since it causes no damage for the defender; only loosing $v_0$ does that!

The optimization of both players in the game is then doable by a stochastic tail ordering on the random variable $U$ for the attacker (as given by (2)) and $U'$ for the defender (as given by (3)), where $U'$ differs from $U$ only in the fact that $U$ relates to an attacker of type $\theta$, while $U'$ is the weighted mix of attackers of all types, since the defender does not know which type it is facing (and hence has to adopt a hypothesis on $\theta$; this is where the Bayesian flavour of the game comes in). Our chosen stochastic order is the $\preceq$-relation introduced in [22], which, in the special case of a categorical distribution, is equivalent to a lexicographic ordering on the probability masses. That is, if two distributions $U_1 = (p_1, \ldots, p_n)$ and $U_2 = (q_1, \ldots, q_n)$ are given, then $U_1 \preceq U_2$ if and only if $p_n < q_n$ or $[p_n = q_n$ and $U_1' = (p_1, \ldots, p_{n-1}) \preceq U_2' = (q_1, \ldots, q_{n-1})]$, with (final) equality and $\succeq$ following in the canonic way. Applying this ordering on the masses that our payoff distributions put on the nodes in $V$ in order of distance to $v_0$, amounts

to the game being about minimizing the attacker's chances to reach $v_0$ for the defender, while the attacker at the same time pushes towards $v_0$ by $\preceq$-maximizing the tail mass. Formally, the optimization problems are:

**Defender:** $\preceq$-minimize $U'$ over $\boldsymbol{x} \in \Delta(AS_1)$, given $\lambda$ and a hypothesis $F$ (distribution) on the adversary type $\theta \sim F(\Theta)$.

**Attacker:** $\preceq$-maximize $U$, given the defender's strategy $\boldsymbol{x}$, from the starting point (type) $\theta \in V$.

To make these rigorous, let us now work out how $U$ and $U'$ are computed for the players: The formal definition of the adversary's utility is a simple matter of conditioning the attack steps distribution on the current situation in the attack graph. With $\Theta$ determining the random starting point $\theta \in V$, the adversary may take one out of several routes $\pi_{\theta,1}, \pi_{\theta,2}, \ldots, \pi_{\theta,m_\theta}$, all going from $\theta$ to $v_0$ (their count being denoted as $m_\theta$). A general such path is represented as $\pi = (\theta, w_1, w_2, \ldots, v_0)$ with all $w_i \in \{v_1, v_2, \ldots\} = V$. The set of nodes constituting $\pi$ is $V(\pi)$. Also, let $d_\pi(u, v)$ be the graph-theoretic distance counting the edges on the subsection from $u$ to $v$ on the chosen path $\pi$. Then, the utility distribution for the attacker assigns to each node $v \in V$ the mass

$$\Pr(\text{adversary's location} = v | V(\pi)) = \frac{f_{Pois(\lambda)}(d_\pi(\theta, v))}{\Pr_{Pois(\lambda)}(V(\pi))}, \tag{1}$$

in which $f_{Pois(\lambda)}(x) = \frac{\lambda^x}{x!} e^{-\lambda}$ is the density of the Poisson distribution, and $\Pr_{Pois(\lambda)}(V(\pi)) = \sum_{x \in V(\pi)} \Pr_{Pois(\lambda)}(d_\pi(\theta, x)) = \sum_{x \in V(\pi)} f_{Pois(\lambda)}(d_\pi(\theta, x))$ (in a slight abuse of notation).

Now, the defender's action comes in, who hopes to cut the rope behind the attacker. Let $c \in V$ be the checked node, then the possibly truncated path is

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

Cutting the rope then means conditioning the distribution of the adversary's location on the shorter (cut) path $\pi|_c$. The formula is the same as (1), only with $\pi$ replaced by $\pi|_c$ now. Since $c \sim \boldsymbol{x}$ follows the defender's mixed spot checking strategy (possibly degenerate), and the set of paths $\pi$ along which the attacker steps forward (at rate $\lambda$) is determined by the random starting position $\theta \sim \Theta$, the utility distribution for the attacker is given as

$$U(\boldsymbol{x}, \theta, \lambda) = (\Pr(\text{adversary's location} = v | V(\pi|_c)))_{v \in V} \tag{2}$$

### 2.3 Equilibrium Definition and -Computation

The defender may associate each possible attack starting point in $G$ with a distinct type of adversary. The belief about the adversary's type is then again $\Theta$, and the ex ante payoff is then

$$U'(\boldsymbol{x}, \lambda) = \sum_{\theta \in \Theta} \Pr_\Theta(\theta) \cdot U(\boldsymbol{x}, \theta, \lambda) \tag{3}$$

Observe that we can equivalently write $U(\boldsymbol{x}, \theta, \lambda)(v) = \Pr_{\boldsymbol{x}}(\text{adversary's location} = v|\theta, \lambda)$, since this is by construction the distribution of the attacker's location, *conditional* on the starting point $\theta$. In this view, however, (3) is just the law of total probability, turning $U'$ into the distribution of the attacker's location.

It turns out that a perfect Bayesian equilibrium fits nicely for our setting. We instantiate the definition from [8, Def.8.1] to our setting.

**Definition 1.** *A* perfect Bayesian equilibrium *(PBE) for a two-player signalling game is a strategy profile such that:*

- **($P_1$) Sequential rationality of the attacker:** *$\forall\theta$, the type-$\theta$-adversary maximizes $U(\boldsymbol{x}, \theta, \lambda)$ over its action space $AS_2$, for the (fixed, randomized) action $\boldsymbol{x}$ chosen by the defender*[4]*.*
- **($P_2$) Sequential rationality for the defender:** *$\forall a_2 \in AS_2$, the defender chooses its action conditional on the observed action $a_2$ of the attacker (signal) as $\boldsymbol{x}^* \in argmin_{a_1 \in AS_1} \sum_\theta \Pr(\theta|a_2) U(a_1, \theta, \lambda)$*
- **($B$) Bayesian updating of beliefs:** *the conditional belief is $\Pr(\theta|a_2) = \Pr(\theta) \cdot \Pr(a_2|\theta) \big/ \sum_{\theta' \in \Theta} \Pr(\theta') \Pr(a_2|\theta')$, whenever the denominator is $> 0$. Otherwise, any distribution $\Pr(\cdot|a_2)$ is admissible.*

How does this fit for us? The assumption of a stealthy attacker means that there are no signals, so our game is not truly a signalling game in the strict sense. However, the notion of a perfect equilibrium nonetheless is meaningful, if the defender has other means of updating a belief about which part of the system is infected. Condition ($P_1$) refers to the attacker taking, knowing the random spot checking pattern $\boldsymbol{x}^*$ of the defender, it will take the "best" path $a_2 \in \{\pi_{\theta,1}, \ldots, \pi_{\theta,m_\theta}\} \subseteq AS_2$ so as to maximize the probability of hitting $v_0$ (after a Poissonian number of steps). Regarding the defender's Bayesian updating prescription ($B$), recall that the attacker is stealthy and hence avoids sending signals. This makes the conditioning in ($B$) be on an empty set, since we receive no signal, formally meaning $a_2 = \emptyset$ and implying a zero denominator in ($B$) because $\Pr(a_2|\theta') = \Pr(\emptyset|\theta') = 0$. This allows the defender to just carry over its a priori belief $\Theta$ into the posteriori belief $\Pr(\theta|\cdot) := \Pr_\Theta(\theta)$, and ($B$) is automatically satisfied. Intuitively, unbeknownst of the attacker's location, the defender faces one big information set, on which it can impose the hypothesis $\Theta$ that will remain unchanged in absence of any signals (for the same reason, we do not have any separating equilibria; they are all necessarily pooling). Finally, plugging $\Pr(\theta|a_2) = \Pr_\Theta(\theta)$ into condition ($P_2$), we end up finding that the defender in fact $\preceq$-minimizes $U'$ as given by (3).

Taking yet another angle of view, we can also arrive at condition (3) by considering the competition as one-against-all [25], where the defender simultaneously faces exactly one adversary of type $\theta$ for all types $\theta \in \Theta$. Then, condition (3) is just a scalarization of the resulting multi-criteria security game (cf. [20]). It follows that each PBE in the sense just explained is equal to a multi-goal

[4] The dependence of $U$ on $a \in AS_2$ is implicit here, but comes in through the probabilities involved to define the utility; we will come back to this in a moment.

security strategy (MGSS), and vice versa, enabling the application of algorithms to compute MGSS [19] in order to get a PBE. This is the method applied in the following (using MGSS as a mere technical vehicle)

## 3 Computational Results

CutTheRope is most conveniently implemented in `R`, since the basic system already ships with most of the required functions. The full details and explanations of the code are available as a supplementary online resource [18].

First, let us assign consecutive numbers as representatives of the locations in the attack graph, displayed as circled numbers in Figure 1b. This constitutes the set $V = \{1, 2, \ldots, 10\}$, with $10 = v_0$, the target asset, and $v = 1$ being the (common) starting point for all attack paths.

In our example game, the defender has $n = |V| - 1 = 9$ strategies, excluding the trivial defense of always (and only) checking the target node $v_0$. The attacker, in turn, can choose from a total of 8 paths from 1 to $v_0 = ⑩$. This set shrinks accordingly when the attack starts from a point $\theta \in V \setminus \{v_0\} = \{1, 2, \ldots, 9\}$, to contain only the paths that contain the respective $\theta$. For $\theta = 1$, this gives the full set of 8 paths, while, for example, $\theta = 3$ leaves only 4 paths for the attacker. As before, let the number of paths per $\theta$ be $m_\theta$, then each type of opponent has a $n \times m_\theta$ payoff matrix. The entries therein are indexed by a node $i \in V$ being checked, and a path $\pi_{\theta,j}$. The attacker's utility distribution (2) by evaluating the Poisson density $f_{Pois(\lambda)}(x)$ for $x = 0, 1, \ldots, |V(\pi_{\theta,j})|$, and conditioning it on the potential cut at node ⓘ (if that node is on the path). Effectively, the conditioning amounts to setting all probability masses on the path from $i$ to $v_0$ to zero, and renormalizing the remainder of the vector (see [18, code lines 30-35]).

Now, the defender comes in and cuts the rope. This is merely another conditioning, i.e., zeroing the mass of all nodes that come after $i$, if $i$ is on the attacker's residual route ("last mile towards $v_0$").

The resulting masses are then assigned to the nodes in $V$, placing zero mass on all nodes that are never reached, either because the attacker would anyway have not come across the node, or if the rope has been cut before the node has been reached (see [18, code lines 36-39]).

The so-constructed distribution (see [18, variable `L`]) is the value $U(\boldsymbol{x} = \boldsymbol{e}_i, \theta = j, \lambda = 2)$ from (2), with $\boldsymbol{e}_i$ being the $i$-th unit vector (acting as a degenerate distribution for our purposes). Having this, it remains to sum up these with weights according to $\Theta$, in the final utility $U'$ as in (3). To this end, we adopt a non-informative prior, i.e., a uniform distribution $\Theta$ on the possible adversary types, i.e., starting points in $V \setminus \{v_0\}$, and iteratively compute the weighted sum (3) (in [18, code line 44]) from an initially constructed vector $U \in \mathbb{R}^{|V|}$ of all zeroes. Note that this, unlike the defender's strategy set, *includes* the target node $v_0$.

The remaining labour of setting up the game and solving for a multi-criteria security strategy, giving the sought perfect Bayesian pooling equilibrium is aided by the package `HyRiM` [21] for `R` ([18, code lines 46-52]).

The PBE obtained is in pure strategies, prescribing the defender to periodically patch potential local buffer overflows at machine 2 (optimal pure strategy being `local_bof(2)`), while the attacker is best off by choosing the attack path `execute(0)` $\rightarrow$ `ftp_rhosts(0,2)` $\rightarrow$ `rsh(0,2)` $\rightarrow$ `full_access(2)`. This matches the intuition of the best strategy being the defense of the target, by avoiding exploits thereon. Since all attack paths intersect at the node `local_bof(2)`, this equilibrium is not surprising. Still, it may appear odd to find the equilibrium not being the shortest among all attack paths. The reason lies in our choice of the attack rate to be $\lambda = 2$: for that setting, it is equally probable for the adversary to take 1 step (chance $f_{Pois(2)}(1) = 0.2706706$) or 2 steps (with the same chance $f_{Pois(2)}(1) = f_{Pois(2)}(2)$), so the attacker is indeed indifferent between these two options, based on its attack rate.

The equilibrium utility for the attacker is it to be located at positions $V = \{1, 2, \ldots, 10\}$ with probabilities $U^* \approx (0.573, 0, 0, 0, 0, 0.001, 0.111, 0.083, 0.228, 0.001)$ i.e., the effect of this defense is as desired; the adversary can get close to $v_0$, but has only a very small chance of conquering it.

A different solution is obtained if we restrict the defender's scope to checking only *some* FTP connections, remote- and secure shells. Under the so-restricted action space $AS_1' = \{$`ftp_rhosts(0,1)`, `ftp_rhosts(0,2)`, `sshd_bof(0,1)`, `rsh(1,2)`, `rsh(0,2)`$\}$, we obtain a PBE in mixed strategies being as in Table 1.

| attack path no. (from Figure 1c) | Pr(attack path) |
|---|---|
| 1 | 0.104 |
| 2 | 0.119 |
| 3 | 0.603 |
| 4 | 0.174 |
| 5 | 0 |
| 6 | 0 |
| 7 | 0 |
| 8 | 0 |

(a) attacker's equilibrium

| location $v \in V$ (from Figure 1b) | Pr(check $v$) |
|---|---|
| `ftp_rhosts(0,1)` | 0.504 |
| `ftp_rhosts(0,2)` | 0 |
| `sshd_bof(0,1)` | 0.181 |
| `rsh(1,2)` | 0.166 |
| `rsh(0,2)` | 0.149 |

(b) defender's equilibrium

Table 1: Example results

The payoff distribution density obtained under this equilibrium is the attacker being located on the nodes $(1, 2, 3, \ldots, 10)$ with probabilities $U^* \approx (0.545, 0.017, 0.030, 0.022, 0.012, 0.034, 0.128, 0.021, 0.045, 0.146)$.

## 4 Conclusions

CutTheRope has been designed for ease of application, but admits a variety of generalizations and possibilities for analytic studies. Examples include (i) probabilistic success on spot checks, (ii) probabilistic success on exploits, (iii) spot checking with random intervals, (iv) taking a fixed number of steps at any point in time, (v) multiple adversarial targets, etc. Cases (i)-(iv) all amount to a mere substitution of the Poisson distribution by: (i) a mix of distributions (one for the success, and one for the failure of a spot check), (ii) a product of probabilities to describe the chances to penetrate all nodes along a path, (iii) a geometric distribution describing the random number of spot checks between two events with exponentially distributed time in between, or (iv) a mix of distributions to describe the steps taken within the periodicity of the defender's activity. Using an artificial target node to represent multiple real targets, (v) also boils down to a change in the attack graph model, but no structural change to the game.

Generally, CutTheRope 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, CutTheRope has no defined start or finish for the defender ("security is never done"), while only one of the two players knows when the game starts and ends. The model is thus in a way complementary to that of 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.

## References

1. 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/
2. Alpcan, T., Baar, T.: Network Security: A Decision and Game Theoretic Approach. Cambridge University Press (2010)
3. BSI: IT-Grundschutz International. Bundesamt für Sicherheit in der Informationstechnik (2016), https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzInternational/itgrundschutzinternational_node.html
4. Carroll, T.E., Grosu, D.: A Game Theoretic Investigation of Deception in Network Security. In: 2009 Proceedings of 18th Int. Conf. on Computer Communications and Networks. pp. 1–6. IEEE, San Francisco, CA, USA (Aug 2009)
5. Dijk, M., Juels, A., Oprea, A., Rivest, R.L.: FlipIt: The Game of "Stealthy Takeover. J. Cryptol. **26**(4), 655–713 (2013)
6. Etesami, S.R., Baar, T.: Dynamic Games in Cyber-Physical Security: An Overview. Dynamic Games and Applications (Jan 2019)
7. Fang, X., Zhai, L., Jia, Z., Bai, W.: A Game Model for Predicting the Attack Path of APT. In: 2014 IEEE 12th Int. Conf. on Dependable, Autonomic and Secure Computing. pp. 491–495. IEEE, Dalian, China (Aug 2014)

8. Fudenberg, D., Tirole, J.: Game Theory. MIT Press
9. Huang, L., Zhu, Q.: Adaptive Strategic Cyber Defense for Advanced Persistent Threats in Critical Infrastructure Networks. arXiv:1809.02227 [cs] (Sep 2018)
10. Jajodia, S., Noel, S., Kalapa, P., Albanese, M., Williams, J.: Cauldron mission-centric cyber situational awareness with defense in depth. In: 2011 - MILCOM 2011 Military Communications Conf. pp. 1339–1344. IEEE
11. Kamhoua, C.A., Leslie, N.O., Weisman, M.J.: Game Theoretic Modeling of Advanced Persistent Threat in Internet of Things. J. of Cyber Security and Information Systems **6**(3) (2018)
12. Khouzani, M., Sarkar, S., Altman, E.: Saddle-Point Strategies in Malware Attack. IEEE J. on Selected Areas in Communications **30**(1), 31–43 (Jan 2012)
13. La, Q.D., Quek, T.Q.S., Lee, J.: A game theoretic model for enabling honeypots in IoT networks. In: 2016 IEEE Int. Conf. on Communications (ICC). pp. 1–6. IEEE (May 2016)
14. Lin, J., Liu, P., Jing, J.: 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. pp. 118–137. Springer (2012)
15. Lye, K.w., Wing, J.M.: Game strategies in network security. Int. J. of Information Security **4**, 71–86 (2005)
16. 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:1811.05622 [cs] (Nov 2018)
17. Qing, H., Shichao, L., Zhiqiang, S., Limin, S., Liang, X.: Advanced persistent threats detection game with expert system for cloud. J. of Computer Research and Development **54**(10), 2344 (2017)
18. Rass, S., König, S., Panaousis, E.: Implementation of Cut-The-Rope in R. `https://www.syssec.at/de/downloads/papers` (July 2019), *supplementary material to this work*
19. Rass, S., Rainer, B.: Numerical computation of multi-goal security strategies. In: Poovendran, R., Saad, W. (eds.) Decision and Game Theory for Security. pp. 118–133. LNCS 8840, Springer
20. Rass, S.: On game-theoretic network security provisioning **21**(1), 47–64
21. Rass, S., König, S.: HyRiM: Multicriteria Risk Management using Zero-Sum Games with vector-valued payoffs that are probability distributions, `https://cran.r-project.org/web/packages/HyRiM/index.html`
22. Rass, S., König, S., Schauer, S.: Defending against advanced persistent threats using game-theory **12**(1), e0168675
23. Rass, S., Knig, S., Schauer, S.: On the Cost of Game Playing: How to Control the Expenses in Mixed Strategies. In: Decision and Game Theory for Security, pp. 494–505. Springer, [S.l.] (2017)
24. Rass, S., Zhu, Q.: GADAPT: A Sequential Game-Theoretic Framework for Designing Defense-in-Depth Strategies Against Advanced Persistent Threats. In: Decision and Game Theory for Security, pp. 314–326. LNCS, Springer, 9996 edn. (2016)
25. Sela, A.: Fictitious play in 'one-against-all' multi-player games. Economic Theory **14**(3), 635–651 (1999)
26. Singhal, A., Ou, X.: Security risk analysis of enterprise networks using probabilistic attack graphs. https://doi.org/10.6028/NIST.IR.7788
27. Zhu, Q., Rass, S.: On Multi-Phase and Multi-Stage Game-Theoretic Modeling of Advanced Persistent Threats. IEEE Access **6**, 13958–13971 (2018)