Risk Assessment in Transactions under Threat as Partially Observable Markov Decision Process

Vassil Vassilev, Doncho Donchev and Demir Tonchev

Abstract This paper presents a theoretical model and algorithms for calculating the security risks for planning active counteractions in transaction processing under security threats. It is a part of an integrated cybersecurity framework, which combines AI-based planning of active counteractions with Machine Learning for the detection of security threats during transaction processing. The risk assessment is based on the optimal strategy for decision making which minimizes the security risks in controlled transactions modeled as Partially Observable Markov Decision Process (POMDP). By statistical reduction, this model is converted into a Markov Decision Process (MDP) with full information so that the algorithm for calculating the risks can use the standard dynamic programming. Although developed primarily for applications in fintech industry, this framework can be adapted to a wide range of business process workflows that incorporate both synchronous operations and asynchronous events caused by human errors, technical faults, or external interventions.

1 Introduction

Cybersecurity becomes critical for successful digital transformation of the businesses in many areas of human activity - fintech industry, e-commerce, business process management, healthcare, public services, etc. Over the last three years we have been working on a hybrid AI-based framework which combines the power of logical analysis of security policies, from one side, with machine learning for data analytics, on the other. Such a framework must secure the transaction processing by accounting for both the threat intelligence, obtained in advance from security experts, and the security risks assessed in real time. Our approach to planning is rooted in the traditional AI planning introduced by McCarthy in the situation calculus, but follows different approach from both conceptual and theoretical point of view, which allows to avoid some of the problems encountered in the original deterministic planning such as the qualification and frame problems. Instead of combining the information from the real world with the planning heuristics in a single representational language, like in the original situation calculus, we have adopted multi-level problem formalization which separates the *domain ontology* from the *security policies* and adds two more levels: *analytical level* of decision making for selecting appropriate actions and applying potential counteractions to the security threats, and *implementation level* for executing ML algorithms for security analytics to detect potential security threats at the different steps of the transactions [8]. For the first three levels of the framework we have adopted the standard languages of the Semantic Web - OWL, SWRL and RDF, which have direct logical interpretation [7], while the implementation level utilizes a variety of ML algorithms for detection [9]. In this article we will present an approach for assessment of the security risks at each step of the transactions in the presence of security threats, which is necessary for planning of suitable counteractions during execution of the transactions and progressing towards completion of the transaction.

Vassil Vassilev
Cyber Security Research Centre, London Metropolitan University, London, UK
e-mail: v.vassilev@londonmet.ac.uk

Doncho Donchev, Demir Tonchev
GATE Institute, Sofia University "St. Kliment Ohridski", Sofia, Bulgaria
e-mail: {doncho.donchev,demir.tonchev}@gate-ai.eu

The paper is organised as follows. First we will briefly review the research in risk assessment from cybersecurity perspective and will set the problem in the context of a hybrid AI-based cybersecurity framework which accounts both the security policies and the threat intelligence to execute active counteractions against the threats detected during transaction execution. After this preliminaries we will introduce the POMDP model, will preform statistical reduction to MDP model and will describe the algorithms for risk assessment, based on the optimal strategy for controlling the transaction under threats. We will illustrate the use of the algorithm by analyzing the decision threshold, which guides the choice of counteraction along the transaction based on the optimal strategy. After brief information about the current state of implementation of the framework we will finish with a discussion and our plans about the future research in this direction.

2 Brief review of the relevant research

The advances in heuristic planning for intelligent control of the transactions and the need to account stochastic factors which interfere with the execution of the transactions, such as errors, faults and intrusions, focused the attention on continuous planning and re-planning. Unfortunately, the heuristic planning faces the need to account the security risks which does not fit within the deterministic models used as a base for the planning algorithms. An adequate formalization of the stochastic planning problem requires working with POMDP model which is significantly more complex than the two popular deterministic models - the classical state-space search and the MDP [1]. An excellent overview of the different models and algorithms for non-deterministic planning from AI perspective is provided in [2]. Despite some recent adoptions of POMDP for the purpose of risk assessment [3, 5, 4, 6], the adoption of POMDP remains valuable for mostly offline analytics due to the need to the need to solve multi-step optimization problem of large complexity.

The major contribution of our research is in the integration of the purely deterministic method for controlling the transactions under threat with the stochastic method for decision making using the risk assessment as a heuristic function, which is based on the original POMDP model but reduced to a tractable MDP problem. This reduction makes possible to use more efficient recurrent algorithms for optimization, based on the standard dynamic programming methodology which for realistic transaction lengths can be executed in real time.

3 Controlling Transactions and Decision Making

Contemporary transaction processing requires planning and controlling the execution of a sequence of operations to reach the goal state, namely the commit point of the transaction. In our security framework [8] each step of the transaction is modeled as a separate *situation*. Along the multi-step transition from situation to situation the transactions face multiple challenges due to the unpredictability of the factors which may influence the process - security threats which may require neutralization, safety threats which may need mitigation or logical non-determinism for choosing alternative options. In accordance with our theoretical framework we are considering both synchronous activities (in our framework they are called *actions*) and asynchronous activities (it events). While the actions change the situations in a deterministic way, the event are the main stochastic factors since they may or may not trigger actions, and also can happen at any time. This way the analytical level can be modelled naturally as a directed AND-OR graph. Choosing suitable operation based on risk assessment when the transactions execute under security threats would allow to implement control algorithms with guaranteed chances for successful commit of the transaction.

As an illustration, Fig. 1 presents one such graph which models a typical transaction for reading the emails in the presence of potential security threats on analytical level. The graph nodes represent situations and are painted in white, green or red; the solid arrows represent the deterministic transitions from situation to situation, while the events and threats are associated with the situations in a non-deterministic way and painted in blue and black, respectively. Some of the actions are normal actions which progress the transaction towards its commit situation, which can be prescribed using suitable heuristics, while other actions are outside of the control since they are

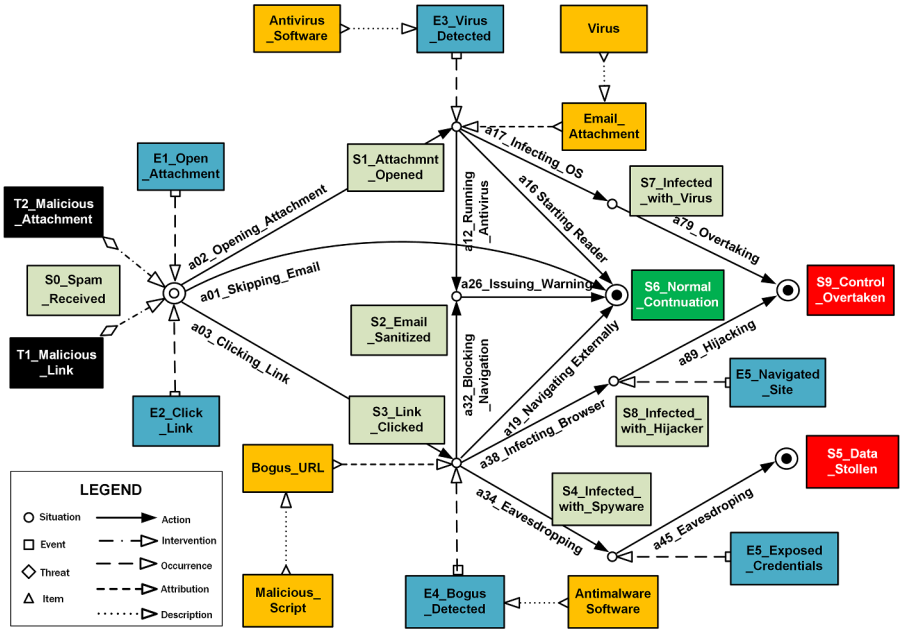

Fig. 1 Threat in the Email

triggered by asynchronous events or caused by security threats. The situations, events and threats can be described qualitatively and quantitatively using various *items*, colored in yellow.

This graph can be created entirely automatically using the domain ontology and the security policies of the framework. However, on order to implement the control strategy, we need to deal with the non-determinism. The graph contains multiple decision points which cannot be resolved without additional information and this is where we need to make informed decision choice of an action to be executed.

4 Transactions under Threat as POMDP

Before we specify the model we will make some assumptions which can be lifted at a later stage:

1. We will omit the descriptions, which use concepts of the type `Item` and will consider only `Situation`, `Action` and `Event` taxonomies.
2. There will be no distinction between situations, free of any threats, between situations, which are result of malicious actions and between transient situations. So the model will consider only the top classes of the taxonomy – `SafeSituation`, `DangerousSituation` and `TransientSituation`.
3. There will be no distinction between different malicious actions and between different counteractions — we will consider only the top roles in the action taxonomies `MaliciousAction` and `CounterAction`.
4. We assume that the counteractions always bring the system back to a safe situation in a single step. This also means that all transient situations are safe.
5. Only events relevant to the threats will be considered (class `Threat`). The non-threatening events will be addressed by the security policies on logical level.

The above assumptions makes possible to apply a reduction of the original POMDP problem with partial information to an MDP problem with full information and to use the recurrent algorithm

of dynamic programming for solving it. This way, we can have a quantitative evaluation of the risk in each situation incrementally.

Due to the presence of asynchronous events, which can be either unpredictable, but anticipated – like many malicious interventions, or unexpected, but predictable – such as human or technical errors, we must model the thransactions under threat as POMDP, rather than as MDP which requires full information. Our model has the following elements:

1. **State space** $S = \{safe, danger, deadend\}$ – corresponds to the different top-level types of situations from risk viewpoint
   a. $safe$ Situations along the normal transactions in absence of any threats
   b. $danger$ Situations in which the system is under the influence of security threats but is still able to recover
   c. $deadend$ Situations in which the system experiences severity and crashes completely under the security threats
2. **Control space** $C = \{noact, respond\}$ – corresponds to the different top-level types of counteractions for risk mitigation
   a. $noact$ – no control intervention, the system goes straight to the next situation according to the planned action in order to continue its normal track of execution of the current transaction
   b. $respond$ – counteraction, which brings the system back to a safe situation after malicious action
3. **Observation space** $Z = \{nothreat, threat, crash\}$ – corresponds to the different top-level types of events from security viewpoint
   a. $nothreat$ – asynchronous event, which is non-threatening and does not require counteraction
   b. $threat$ – detection of malicious intervention which requires counteraction
   c. $crash$ – losing control of the system without chance for recovery
4. **Transition kernel** $q(s_{n+1}|s_n, c_{n+1})$ – probability of the transition from situation s_n to situation s_{n+1} under control c_{n+1}, calculated as follows
   - $q(safe|safe) = p$, p is the probability for absence of threats after transition from a safe situation
   - $q(danger|safe) = 1 - p$, $1 - p$ is the probability for presence of threats after transition from a safe situation
   - $q(safe|danger, respond) = 1$ because the counteraction in a dangerous situation eliminates the threat
   - $q(deadend|danger, noact) = 1$ because the absence of counteraction in dangerous situation leads to an inevitable deadend of the system
   - $q(deadend|deadend) = 1$ since there is no way out of the deadend
5. **Occurrence kernel** $t(z_n|s_n)$ – probability of occurrence of event z_n in state s_n, calculated as follows
   - $t(nothreat|safe) = p_{11}$ – probability of not observing threat in a safe state
   - $t(nothreat|danger) = p_{12}$ – probability of not observing threat in a dangerous stage (false negative)
   - $t(threat|safe) = p_{21}$ – probability of observing threat in a safe state (false positive)
   - $t(threat|danger) = p_{22}$ – probability of observing threat in a dangerous state
   - $t(crash|deadend) = 1$ – probability of observing the system crash under threat

   Let's denote the matrix with entries $p_{ij}, i, j = 1, 2$ by P. Its transpose P^T is a stochastic matrix since
   $$p_{11} + p_{21} = p_{12} + p_{22} = 1.$$
6. **Rewards** – quantitative measures of the costs of the actions taken as follows

a. Current reward $r(c)$ calculated as follows: $r(noact) = 0, r(respond) = -c$ where $c > 0$ is the cost for using $respond$
b. Final reward $R(s)$ calculated as follows: $R(safe) = R(danger) = 1$ if either the transaction terminates normally or the threat occurs after finalizing it, and $R(deadend) = 0$ if the crash occurs during the transaction.

7. **Horizon** N – length of the transaction, calculated as the number of safe situations in it.

5 Optimal Strategy for counteracting and its cost

The solution of the risk assessment task can be obtained as a byproduct of the calculation of the optimal strategy for control of the transactions.

Definition *Security decision* $\phi(s)$ is a function which on each step of the transaction s chooses either $noact$ or $respond$.

The security decisions may modify the original transactions by enforcing $respond$ actions in some of the situations. Therefore, they can extend the transaction path. If the security decisions are wrong it might be even possible to end the transaction in a $deadend$ situation. In order to maximize the chances to make the right decisions we will account all information available at the time of decision making, which will turn the security decision into a stochastic function of the parameters of the POMDP model.

Definition *Decision policy* $\pi = (\phi(1), \phi(2), ..., \phi(N))$ is a collection of security decision functions such that on each step n of the transaction, $\phi(n)$ depends only on the past history till time n, and the prior probabilities of the states at time 0, that is before the transaction has begun.

We assume that the prior probability of state $deadend$ is 0, since otherwise any policy makes no sense. Therefore, the sum of the prior probabilities of the other two states is equal to 1, and the prior distribution of the states at time 0 is determined by the prior probability x of state $safe$. So, we are now looking for a decision policy π which maximizes the total reward $v^\pi(x) = E_x^\pi(R(state_N) - cK)$, where E_x^π is the expectation, corresponding to the policy π and the prior probability x, and K is the number of times when we apply the action $respond$. In the above expression $R(state_N)$ is the final income which we get in the last step of the transaction.

Definition *Value function* of the POMDP model is the function

$$v(x) = \max_\pi v^\pi(x)$$

Definition The policy π such that $v(x) = v^\pi(x)$ is an *optimal policy*.

The optimal policy π of the POMDP maximizes the chances to avoid a crash during the transaction, taking into account the total price of counteractions. It solves the following optimization problem:

$$v^\pi(x) = E_x^\pi(R(s_N) - cK), \tag{1}$$

where x the prior probability of the state $safe$ in the moment $n = 0$, s_N is the final state of the controlled process, and K is the total number of times when the counteraction has been used. Here, E_x^π is the mathematical expectation corresponding to π and x.

To calculate the optimal strategy we will follow the standard procedure for reducing the POMDP model with partially observable states to a MDP model with fully observable states which would

allow to apply the standard algorithm of dynamic programming [7]. The reduction can be done by following the steps bellow:

1. Constructing sufficient statistics for the POMDP model by solving the filtration equations
2. Building a model with fully observable states using the sufficient statistics from step one
3. Solving Bellman's equation for the MDP model built in step two making use of the dynamic programming algorithm

This gives us the optimal strategy in both POMDP and MDP models. Based on it we can now estimate the risks.

Definition: The risk corresponding to the prior probability x of the state $safe$ of the POMDP model is equal to $1 - v(x)$, where

$$v(x) = \sup_{\pi} v^{\pi}(x), \tag{2}$$

is the value function of the model.

So the risk in each state can be assessed if the optimal strategy is known. In the general case this is a difficult problem, but fortunately, for the special case of our POMDP there is an elegant solution based on statistical reduction of the POMDP model to deterministic MDP model.

Let f_n (resp. g_n), $n = 0, 1, ..., N-1$, be 3x1-vectors with elements equal to the prior (resp. posterior) probabilities of the states $safe$, $danger$ and $deadend$ during the transaction. We assume that $f_n(1)$ and $g_n(1)$ correspond to state $safe$, $f_n(2)$ and $g_n(2)$ – to state $danger$, and $f_n(3)$ and $g_n(3)$ – to state $deadend$.

We can think of these vectors as points in the two-dimensional simplex in $\mathbf{R}^3$ (Fig. 2). In order to exclude the trivial case of a system's breakdown before any transaction has begun, we assume that $f_0(3) = 0$. Thus, we have $f_0(1) = x$, $f_0(2) = 1 - x$, where x is the same as in formulas (1) and (2). The other vectors f_n and g_n satisfy the following relations:

- Since the state $deadend$ is absorbing, $g_n(3) = 0$ or 1. If $g_n(3) = 1$, then $f_m(3) = g_m(3) = 1$ for all $m > n$.
- Making use of the Bayes formula, the coordinates of the vector g_n can be calculated as follows:

$$g_n(1) = \frac{f_n(1)p_{21}}{f_n(1)p_{21} + f_n(2)p_{22}} := \Gamma^1(f_n(1), f_n(2)), \tag{3}$$

$$g_n(2) = \frac{f_n(2)p_{22}}{f_n(1)p_{21} + f_n(2)p_{22}}, \; g_n(3) = 0, \tag{4}$$

if $z_n = threat$;

$$g_n(1) = \frac{f_n(1)p_{11}}{f_n(1)p_{11} + f_n(2)p_{12}} := \Gamma^2(f_n(1), f_n(2)), \tag{5}$$

$$g_n(2) = \frac{f_n(2)p_{12}}{f_n(1)p_{11} + f_n(2)p_{12}}, \; g_n(3) = 0, \tag{6}$$

if $z_n = nothreat$;

$$g_n(1) = 0, g_n(2) = 0, g_n(3) = 1, \tag{7}$$

if $z_n = crash$.

On the other hand, if $g_n(3) = 0$ then the coordinates of f_{n+1} are

$$f_{n+1}(1) = pg_n(1), f_{n+1}(2) = (1-p)g_n(1), f_{n+1}(3) = g_n(2), \tag{8}$$

whenever $c_n = noact$;

$$f_{n+1}(1) = p,\ f_{n+1}(2) = 1 - p,\ f_{n+1}(3) = 0, \tag{9}$$

if $c_n = respond$ where $\Gamma^1(x, 1-x)$ and $\Gamma^2(x, 1-x)$ are the posterior probabilities to remain safe after detecting absent or present threats, respectively.

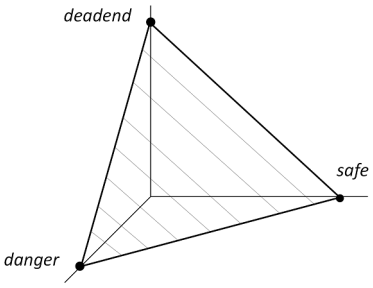

Fig. 2 Situation Simplex

The last equations show that if we consider the vectors $g_n, n = 0, 1,..., N-1$, as points in the two-dimensional simplex, they are located either in the vertex of the simplex, corresponding to state $deadend$, or on the edge, connecting the vertices corresponding to states $danger$ and $safe$. On the other hand, the location of the points on this edge is entirely determined by a single coordinate, say that which is equal to the posterior probability of the state $safe$. This observation plays an important role for reducing the POMDP model to a MDP model with fully observable states.

According to the general theory of POMDP (see [1]), sufficient statistics allow to reduce the initial POMDP problem to a fully observable MDP problem on the base of posterior probabilities $g_n, n = 0, 1, \ldots N-1$.

We consider the following fully observable MDP model. Its state space is the set $S = (0, 1) \cup \{*\}$, where $*$ is an isolated point.

Definition: The controlled process in the model with complete information is defined as

$$x_n = \begin{cases} *, & \text{if } g_n(3) = 1 \\ g_n(1), & \text{if } g_n(3) = 0 \end{cases},\ n = 0, 1, \ldots N-1.$$

Let us note, that in view of (3), (5), and the total probability formula, the initial distribution of x_0 is the following:

$$x_0 = \begin{cases} \Gamma^1(x, 1-x) & \text{with probability } p_{21}x + p_{22}(1-x) \\ \Gamma^2(x, 1-x) & \text{with probability } p_{11}x + p_{12}(1-x) \end{cases}.$$

The fact that P^T is a stochastic matrix implies that the distribution of x_0 is a proper probability distribution. The same holds for all distributions that appear in the definition of the transition kernel $t(\{y\}|x, c)$ of the model with fully observable states. The filtration equations (3)–(9), and the total probability formula motivate us to define it as follows:

$$t(\{y\}|x, c) = \begin{cases} px p_{21} + (1-p)x p_{22}, & y = \Gamma^1(px, (1-p)x) \\ px p_{11} + (1-p)x p_{12}, & y = \Gamma^2(px, (1-p)x) \\ 1 - x, & y = * \end{cases}$$

provided that $c = noact$,

$$t(\{y\}|x, c) = \begin{cases} pp_{21} + (1-p)p_{22}, & y = \Gamma^1(p, 1-p) \\ pp_{11} + (1-p)p_{12}, & y = \Gamma^2(p, 1-p) \end{cases},$$

provided that $c = respond$,

$$t(*|*, \cdot) = 1.$$

In all other cases we set $t(\{y\}|x, \cdot) = 0$. The final reward is

$$R(x) = 1, x \in (0,1), R(*) = 0.$$

The other elements of the model — state space C, running reward r and horizon N remain unchanged after the reduction.

Consider the functions

$$V_n(x) = \max_{\pi} E_x^{\pi}(\Sigma_{k=n}^{N-1} r(c_{k+1}) + R(x_N)), n = 0, 1, \ldots N-1. \tag{10}$$

They satisfy the Bellman's equation

$$V_n(x) = \max(V_n^{noact}(x), V_n^{respond}(x)), \tag{11}$$

and the final condition

$$V_N(x) = R(x). \tag{12}$$

In (11), $V_n^{noact}(x)$ and $V_n^{respond}(x)$ are one-step ahead estimates of both actions $noact$ and $respond$:

$$\begin{aligned} V_n^{noact}(x) &= (pxp_{21} + (1-p)xp_{22})V_{n+1}(\Gamma^1(px, (1-p)x)) \\ &+ (pxp_{11} + (1-p)xp_{12})V_{n+1}(\Gamma^2(px, (1-p)x)), \end{aligned}$$

$$\begin{aligned} V_n^{respond}(x) &= -c + (pp_{21} + (1-p)p_{22})V_{n+1}(\Gamma^1(p, 1-p)) \\ &+ (pp_{11} + (1-p)p_{12})V_{n+1}(\Gamma^2(p, 1-p)). \end{aligned}$$

Let us note that since after action $respond$ the system instantly falls into a $safe$ state ($x = 1$), the right-hand side of the last formula does not depend on x, but still depends on n.
The optimal strategy φ_{n+1} at any moment of time $n = 0, 1, \ldots, N-1$ is the following:

$$\varphi_{n+1}(x) = \begin{cases} noact, & \text{if } V_n(x) = V_n^{noact}(x) \\ respond, & \text{if } V_n(x) = V_n^{noact}(x) \end{cases}.$$

These equations can be solved backwards, starting with the state of successful completion of the transaction. For example, for $n = N-1$ we get:

$$V_{N-1}(x) = \max(1-c, x),$$

$$\varphi_N(x) = \begin{cases} noact, & \text{if } x \geq 1-c \text{ (above the threshold)} \\ respond, & \text{if } x < 1-c \text{ (bellow the threshold)} \end{cases}$$

The remaining iterations until reaching the beginning of the transaction can be performed recursively, taking the previously calculated solution as terminal.

Finally, the connection between the value functions in both models is given by the formula

$$\begin{aligned} v(x) &= (xp_{21} + (1-x)p_{22})V_0(\Gamma^1(x, 1-x)) \\ &+ (xp_{11} + (1-x)p_{12})V_0(\Gamma^2(x, 1-x)). \end{aligned}$$

6 Analysis of the results

In this section we will analyse the results of applying the optimal strategy to the problem for risk assessment. The model parameters used in the calculations are as follows:

- $p_{11} = 0.9$ is the probability for not detecting attack in a safe situation $t(nothreat|safe)$
- $p_{22} = 0.9$ is the probability for detecting an attack in a dangerous situation $t(threat|danger)$
- $p = 0.9$ is the probability of not having an attack after transition from a safe situation $q(safe|safe)$

- $r(respond) = -0.1$ is the cost of responding to a threat
- $N \in \{1..50\}$ is the horizon of the transaction.

Remaining steps	Probability Threshold	0.25	0.5	0.75	1.0
1	0.9	0.1	0.1	0.0552	0.0018064
2	0.87134	0.12866	0.127227	0.083137768	0.030414629
3	0.842985516	0.157014484	0.15416376	0.110777719	0.058717893
5	0.787180101	0.212819899	0.207178904	0.165176837	0.114422501
7	0.732558117	0.267441883	0.259069789	0.218422348	0.168945816
10	0.652789319	0.347210681	0.334850147	0.296180972	0.248570520
15	0.550343126	0.449656874	0.43217403	0.39604552	0.350831653
20	0.404677698	0.595322302	0.570556187	0.53804018	0.496233951
30	0.181781435	0.809352173	0.782307637	0.755319458	0.718727575
50	4.04E-05	0.98990967	0.979799118	0.969688567	0.959630849

Table 1 Risk thresholds of the optimal strategy for different prior probabilities of having threat

Tab. 1 presents the optimal policy threshold for making decision to counteract which can be done by comparing it to the posterior probability to remain safe at different steps of the transaction. The risk has been calculated for four different prior probabilities. Their choice reflects the most typical cases of potential distribution of the threats as follows:

- 1.0: No threats are expected in the beginning of the transaction. This is the case when we are operating clean computer, browser or ATM machine.
- 0.25: Low probability to start a transaction in a safe state. This is the case when it is very likely for threats to occur immediately after starting the transaction (for example infected computer, spyware in the browser or tampered ATM machine).
- 0.50: Equal probabilities for presence and absence of attacks at the beginning of the transaction. This is a case of maximum uncertainty about the threats, i.e., we have a weak threat intelligence.
- 0.75: More likely to start in a safe state at the beginning but possible intrusion at a later step. This is statistically safe prediction when using clean computer, browser or ATM machine.

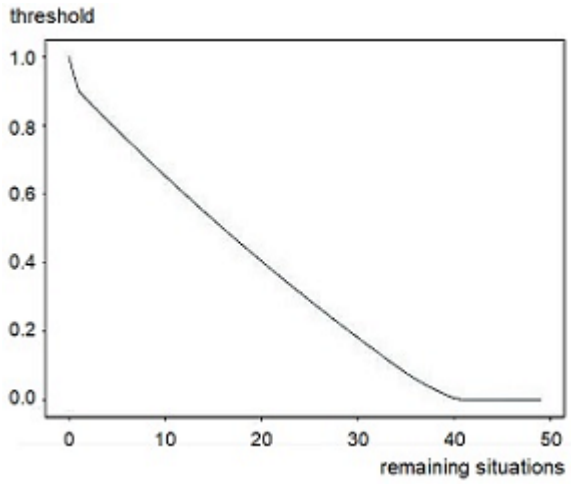

Fig. 3 Decision Threshold

The first column of the table contains the number of remaining steps till completion of the transaction, while the second - the threshold of the optimal strategy. Fig. 3 illustrates the evolution of the threshold in function of the remaining steps of the transaction. It shows that it is higher when there are fewer remaining steps of the transaction, because counteracting towards the end of the transactions is more efficient due to lower costs. The remaining columns of the table contain the estimations of the risk for fixed priory probabilities. They show that when increasing the prior probability the risks decrease, which matches the intuition and

is confirmed for other fixed values of the probability parameter as well. At the same time, the results also show that the risks increase with the length of transactions, which also matches the intuition. These results give enough evidence that estimating the risks on the base of the optimal strategy can be an adequate heuristic to compare alternative paths through the graph for planning countermeasures.

7 Conclusion and future plans

Our hybrid cybersecurity framework employs a number of enabling technologies. The risk assessment component presented here adds to it decision making heuristics for choosing an optimal counteraction to neutralize the security threats and commit the transaction despite the threats.

The method of assessing the security risks based on POMDP model presented here can be used for further analysis of the risk-related problems. Particularly interesting would be to investigate the impact of false negatives p_{12} and false positives p_{21} of the data analytics engines on the security risks and the possibility to account more information about the transactions for further tunning of the control strategy. We are also planning to add reinforcement learning capabilities to the framework for further tuning of the model and improving the algorithms for assessment.

Although developed primarily for applications in fintech industry, this framework can be adapted to a wide range of business process workflows - production line fault management, critical infrastructure protection, public safety management, autonomous agent control, etc.

Acknowledgments

The research reported here is a result of collaboration between the Cyber Security Research Centre (UK) and GATE Institute (Bulgaria). It has been funded partially under EU Horizon 2020 and Innovate UK projects. The Cyber Security Research Centre would also like to acknowledge the continuing support of Lloyds Banking Group, which has been the driver behind this security framework. The opinions and the results presented in this paper are, however, of the authors only and do not reflect the official policies of these organisations.

References

1. E. Dynkin and A. Yushkevich, “Controlled Markov Processes”, New York: Springer, 1979.
2. Masuam, A, Kolobov, “Planning with Markov Decision Processes: An AI Perspective”, San Rafael: Morgan & Claypool, 2012.
3. A. Mundt, “Dynamic risk management with Markov decision processes”, Karlsruhe: KIT Sc. Publ., 2008.
4. Y. Liang, Risk Management by Markov Decision Processes, PhD Thesis, University of Manitoba, Winipeg.
5. O. Kreidl, “Analysis of a Markov Decision Process Model For Intrusion Tolerance”, Int. Conf. Dependable Syst. and Networks, IEEE Xplore, 2010, pp. 156–161.
6. E. Jean-Baptiste, P. Rotshtein, M. Russell, “POMDP Based Action Planning and Human Error”, 11th IFIP Int. Conf. on AI Applications and In- novations (AIAI 2015), pp.250-265.
7. K. Bataityte, V. Vassilev and O. Gill, “Ontological Foundations of Modelling Security Policies for Logical Analysis”, IFIP Advances in Inf. and Comm. Technology, vol. 583. Springer, 2000, pp. 368–380.
8. V. Vassilev, V. Sowinski-Mydlarz, P. Gasiorowski et al., “Intelligence Graphs for Threat Intelligence and Security Policy Validation of Cyber Systems”, Advances in Int. Syst. and Computing, vol. 1164. Springer, 2000, pp.125–140.
9. V. Sowinsky-Mydlarz, J. Li, K. Ouazzane and V. Vassilev, “Threat Intelligence Using Machine Learning Packet Dissection”, 20th Int. Conf. on Security and Management (SAM’21). Springer, 2021 (in print).