Security Risk Management via Dynamic Games with Learning

Praveen Bommannavar
Management Science & Engineering
Stanford University
Stanford, California 94305
Email: bommanna@stanford.edu

Tansu Alpcan
Deutsche Telekom Laboratories
Technical University of Berlin
10587 Berlin, Germany
Email: alpcan@sec.t-labs.tu-berlin.de

Nick Bambos
Electrical Engineering
Stanford University
Stanford, California 94305
Email: bambos@stanford.edu

***Abstract*—This paper presents a game theoretic and learning approach to security risk management based on a model that captures the diffusion of risk in an organization with multiple technical and business processes. Of particular interest is the way the interdependencies between processes affect the evolution of the organization's risk profile as time progresses, which is first developed as a probabilistic risk framework and then studied within a discrete Markov model. Using zero-sum dynamic Markov games, we analyze the interaction between a malicious adversary whose actions increases the risk level of the organization and a defender agent, e.g. security and risk management division of the organization, which aims to mitigate risks. We derive min-max (saddle point) solutions of this game to obtain the optimal risk management strategies for the organization to achieve a certain level of performance. This methodology also applies to worst-case scenario analysis where the adversary can be interpreted as a nature player in the game. In practice, the parameters of the Markov game may not be known due to the costly nature of collecting and processing information about the adversary as well an organization with many components itself. In this case, we apply ideas from Q-learning to analyze the behavior of the agents when little information is known about the environment in which the attacker and defender interact. The framework developed and results obtained are illustrated with a small example scenario and numerical analysis.**

I. Introduction

Security has always been an important problem for organizations and the need for scalable and cost efficient solutions continues to grow. As information technology (IT) takes a greater role in the daily operations of a large company, it has become increasingly difficult, perhaps impossible, to be completely shielded from risk. A combination of the often complex interdependencies that are inherent in an organization and the greater degree of technological vulnerability make it difficult to manage these risks appropriately. The growing risks in recent years are evidenced by unprecedented levels of investment in security mechanisms. Indeed, not only have there been incidents compromising the integrity of industry operations but many government organizations also have found the need to give greater attention to risk management [1].

At present, the techniques utilized by organizations to handle threats are dominated by empirical approaches [2]. These methods are systematic, but without mathematical rigor. We aim to take a decision-theoretic approach to risk management. By quantitatively approaching the field of risk management we may not only (partially) automate the decision making process but also handle problems on a larger scale and more efficiently with computer-based solutions.

Despite the largely heuristic nature of the current state-of-the-art, there have been recent contributions to quantitative risk management [3]. One example is the SecureRank framework [4] for prioritizing vulnerabilities on a computer network. This paper builds on the Risk-Rank model of [5] in which risk levels are transferred between components of an organization due to interdependencies between these components. The framework quantitatively keeps track of the risk in an organization by utilizing a diffusion model similar to that of the well-known Page-Rank algorithm used by Google [6]. In [7], an optimization scheme is built on top of this framework, where a system administrator allocates resources to mitigate exposure to risk.

In this paper we move to a formulation with not only a defender that wishes to reduce risk levels, but also an attacker that aims to increase them. The tools of non-cooperative game theory provide a most appropriate platform to study the interactions between these agents. Other works also suggesting the application of game theory to security problems include [8], [9] and [10]. Here we specifically look at zero-sum Markov games. The attacking agent may also be considered as a so-called Nature player to capture the effect of random phenomena that work to increase risk levels. Such a scenario corresponds to a worst-case analysis. The min-max policy we obtain as a result of the Markov games allows the defender to take actions to keep the risk below some level despite the actions of the nature or malicious player.

In many cases and scenarios, the agents operate under limited information and are unable to correctly assess the Markov model. Neither the attacker nor the defender may know the effect of their actions on rewards or future states. Indeed, organizations are riddled with missing information about their processes, and it is often costly to obtain this information in one place. Motivated by this scenario, we apply current results in Q-learning [11], [12] to the zero-sum Markov game defined. Hence, the attacker and defender are modeled as agents who learn from their actions and get a more accurate understanding of the game they are playing as time progresses.

In Section II we review the model of [5] which we shall build upon to describe the dependencies between components of an organization and the way they transfer risk to each other. Section III elaborates on this model and puts it into a game theoretic framework. A situation in which agents must learn the environment of the game is described in Section IV, and a numerical example is given in Section V. Finally, we conclude in Section VI by summarizing our findings and offering directions for future work.

II. Model

In this paper we work with a mathematical formulation of our problem with several layers of abstraction (see Fig. 1). First, we revisit the Risk-Rank model of [5], which captures the evolution of relative risks in an organization. Then, due to issues of observability, we group regions of risk together into states and develop a Markov model. We continue building on this framework by introducing an infinite horizon zero sum game on top of this Markov model to study interactions of an attacker and defender who wish to raise and lower security levels, respectively. Finally, we briefly investigate reinforcement (Q-) learning to analyze the behavior of these agents when the parameters of the game are not known, as is commonly the case due to the complexity and cost of measuring them.

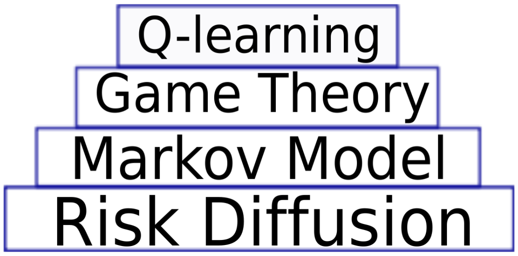


Fig. 1. Layers of our model

A. Probabilistic Risk Framework

To study the way different units of a business interact with each other and transfer risk to one another, [5] has modeled the components of an organization as nodes of a directed graph, X, consisting of M_X nodes each of which carries some amount of risk. As in [7] we shall also consider an additional node in our model, a so called *risk sink* which signifies how much risk is outside of the system so that our model considers $M_X + 1$ entities. Dependencies in the organization are represented by edges in the graph, $\mathcal{E}_X$. The risk for each of these nodes is represented by what is called a relative risk probability vector

$$v^X(t) = [v_1^X(t), \dots, v_i^X, \dots, v_{M_X+1}^X(t)]$$

where v_i^X is the risk carried by component i of organization X. We take $\sum_{i=1}^{M_X} v^X(t) = 1$ for all t and let the risk propagate as

$$v^X(t+1) = Hv^X(t)$$

where H is a column stochastic matrix describing the effect of risk moving through the organization. H is determined by interconnections between the business units $\mathcal{E}_X$. That is, if there is a connection to business component j from i, we let the matrix entry H_{ji} indicate the strength of this connection. In the special case of the risk sink, the interpretation of directed edges to business units is that certain components of an organization are exposed to risk from external sources. We interpret directed edges from business units to the risk sink as the propensity for risk to leave the system.

This formation opens the possibility for analysis on several levels. A fundamental question in this framework is whether any sort of equilibrium exists and how to interpret it. Indeed, [5] investigates this question by first considering modified dynamics

$$v^X(t+1) = \alpha Hv^X(t) + \beta v^X(0) \tag{1}$$

where $\alpha + \beta = 1$ so that the current risk level is taken to be a convex combination of initial estimates and the effect of risk propagation in the organizational network. This is similar to the Page-Rank model used in the Google search engine [6]. It is seen that under some conditions, the risk level converges to

$$v^{X*} = \lim_{t\to\infty} v^X(t) = \beta(I - \alpha H)^{-1} v^X(0)$$

We also have convergence for the special case of $\alpha = 1, \beta = 0$, under conditions of irreducibility and finite state space. One can further analyze diffusions from an optimization perspective, in which an administrator is able to choose among several transition matrices $\{H_1, \dots, H_i, \dots, H_K\}$ for each time step at various costs $\{c_1, \dots, c_i, \dots, c_K\}$ to obtain desired system performance [7].

B. Markov Model and Dynamics

In this paper, we convert the given risk diffusion dynamics of (1) into a discrete-time, finite state space Markov model. Although the dynamics specified in previous subsection more precisely quantify the risk level seen by each business unit of an organization, in practice one cannot discern such levels of precision. That is, reporting of risk levels is generally done with a finite number of quantized levels. In view of this, we partition the $M_X + 1$ dimensional simplex into K *risk regions*. Each risk region groups several values of risk into a cluster which shall be treated as one state.

Therefore, we now consider a new state space upon which to perform analysis: let the K partitions of the probability simplex in $\mathbf{R}^{M_X+1}$ comprise the states

$$S = \{s_1, \dots, s_i, \dots, s_K\}$$

where each s_i corresponds to region i of the simplex. This can be seen for $\mathbf{R}^4$ in Fig. 2.

With this new state space, we can also specify new dynamics. Just as H in the previous section dictated the diffusion of risk throughout organization (that is, the path taken over the simplex), we now consider mechanisms through which the risk state $s(t)$ evolves over time. Now, in the Markov chain setting, we may also introduce randomness into the model to capture the fact that unforeseen events can also have an effect

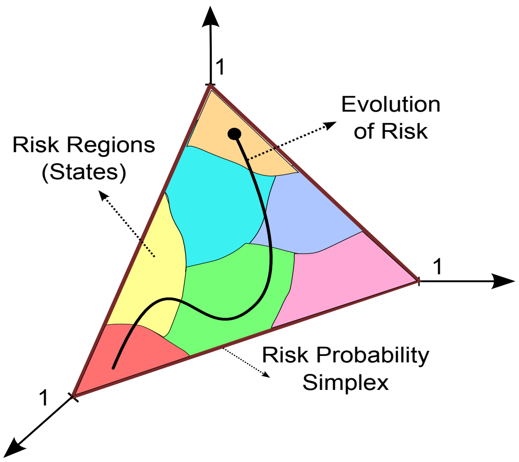


Fig. 2. Evolution of risk

on the evolution of risk. For instance, even though the risk state may indicate low vulnerability levels, there is also some probability of natural phenomena (damage to infrastructure due to lightening strikes, productivity loss from power outages) that would drastically increase the risk of several business units.

Therefore, we now consider a transition matrix $P \in \mathbf{R}^{K \times K}$ that probabilistically governs the evolution of the risk from one state to the next in a time homogeneous Markov chain, denoted by using the standard definition $P_{ij} = \mathbf{P}[s(t+1) = s_j | s(t) = s_i]$.

III. Zero-Sum Markov games

We now adopt a game theoretic approach [10] to analyze the interactions between an attacker and defender who are interested in driving the state of the system to different levels. We consider a defending agent, $\mathcal{D}$, in an organization, who allocates resources towards keeping the risk level low. On the other hand, business units often face attacks from malicious attackers who use aim to harm the organization, e.g. by bringing its security level down. Furthermore, random events posing a security risk can be viewed as malicious actions from a "nature player". Although this nature player is not a conscious agent, it provides a way for conducting a worst-case analysis. We model all of these harmful agents as one attacker, $\mathcal{A}$, since their objectives are approximiately aligned, i.e. their actions increase the risk level of the organization X.

Each agent has a finite set of actions from which to choose at each time step: the attacker decides on an action $a \in A^{\mathcal{A}}$ and the defender chooses $d \in A^{\mathcal{D}}$. Let there be $N^{\mathcal{A}}$ actions for the attacker and $N^{\mathcal{D}}$ actions for the defender. The evolution of the state $s(t)$ now depends on the actions of each agent, and so we construct for each action of the attacker and defender a matrix $M(a,d)$ which affects the distribution of the state, $p^S(t)$, as

$$p^S(t+1) = M(a,d)p^S(t) \tag{2}$$

Now that we have defined the dynamics of our problem, let us make the nature of the game more precise. At each stage, t, given that the state is $s(t)$, the attacker and defender play a zero-sum game

$$G(s(t)) = [G_{a,d}(s(t))]_{N^A \times N^D} \tag{3}$$

That is, entry (a,d) of $G(s(t))$ gives the cost to $\mathcal{D}$ for being in state $s(t)$ and for the combination of actions (a,d). Note that, this is the exact amount of benefit for the attacker $\mathcal{A}$. Mathematically, we write the expected aggregate cost to the defender, $\bar{Q}$, as:

$$\bar{Q} := E\left[\sum_{t=1}^{\infty} \alpha^t G_{a(t),d(t)}(s(t))\right] \tag{4}$$

where $a(t) \in A^{\mathcal{A}}$, $d(t) \in A^{\mathcal{D}}$, $s(t) \in S$, and $\alpha \in (0,1)$ is a discount factor indicating the importance of future payoffs.

The strategies of the attacker, $\mathcal{P}^A$ and defender, $\mathcal{P}^D$, are state dependent (but not time dependent, since in an infinite horizon framework we need only look for stationary policies) and are defined at probability distributions over the attack and defense action sets:

$$p^A(s) = [p_1^A(s), \ldots, p_{N_A}^A(s)]$$
$$p^D(s) = [p_1^D(s), \ldots, p_{N_D}^D(s)]$$

for each state $s \in S$. Let us, for brevity, write $(p^{\mathcal{A}}, p^{\mathcal{D}})$ to mean the policies for the attacker and defender for each state. We use Markov decision methods to find saddle point strategies for the game. Since the cost to the defender is the exact opposite of that of the attacker, we need only focus on the computation of the strategy for the defender. A saddle point $(p^{\mathcal{A}*}, p^{\mathcal{D}*})$ has the property that

$$\bar{Q}(p^{\mathcal{A}}, p^{\mathcal{D}*}) \leq \bar{Q}(p^{\mathcal{A}*}, p^{\mathcal{D}*}) \leq \bar{Q}(p^{\mathcal{A}*}, p^{\mathcal{D}}) \tag{5}$$

for all policies $(p^{\mathcal{A}}, p^{\mathcal{D}})$. In our context, the property (5) means that the defender can choose the policy $p^{\mathcal{D}*}$ and be guaranteed a certain level of performance *no matter what* the attacker does. It also implies that with the fixed behavior of $p^{\mathcal{A}*}$ for the attacker, the performance of the policy $p^{\mathcal{D}*}$ is the best the defender can do.

It is possible to use standard methods in Dynamic Programming (DP) to get a saddle point policy for the defender. First, we can write the optimal cost-to-go by using the DP recursion

$$\begin{aligned} Q_{t+1}(a,d,s) = G_{a,d}(s) + \alpha \sum_{s' \in S} M_{s,s'}(a,d) \\ \min_{p^D(s')} \max_{a} \sum_{d \in A^D} Q_t(a,d,s') p_d^D(s') \end{aligned}$$

which converges to the the optimal cost-to-go function, Q^*, as $t \longrightarrow \infty$. This can be split into

$$V_t(s) = \min_{p^D(s')} \max_{a} \sum_{d \in A^D} Q_t(a,d,s') p_d^D(s') \tag{6}$$

$$Q_{t+1}(a,d,s) = G_{a,d}(s) + \alpha \sum_{s' \in S} M_{s,s'}(a,d) V_t(s) \tag{7}$$

By iterating between equations (6) and (7) we would be able to converge to the saddle point. It now remains to determine how to solve (7) and also get the associated probability distribution for the defender. To this end, we introduce the following Linear Program (LP):

$$\begin{aligned} \min_{p^D(s)} \quad & V_t(s) \\ s.t. \quad & \sum_{d\in A^D} Q_t(a,d,s)p^D_d(s) \leq V_t(s), \forall a \in A^{\mathcal{A}}, \\ & \sum_{d} p^D_d = 1, p^D_d \geq 0, \forall d \in A^{\mathcal{D}}. \end{aligned}$$

for each state $s \in S$. Using the value of $V_t(s)$, we can iteratively use equation (6) and with the LP to converge to the desired saddle point solution.

Obtaining the corresponding saddle-point strategy of the attacker can be done by switching the min and max in (6) with maximization over $p^{\mathcal{A}}(s)$ and minimization over d; one needs only adjust the associated LP as well. Since the solution is a saddle point, the values Q^* and V^* do not change.

In our context, finding the saddle point is valuable because if played by one agent, the opponent will have no incentive to deviate from it, which leads to a performance bound.

IV. Q-learning

It is often difficult to determine the transition probabilities between states. Indeed, surveys may be made to assess the degree of interconnectivity between business units, but in practice, the parameters of our model will be not be perfectly known. Therefore, we are interested in studying situations in which both agents are able to more accurately understand the model as time passes. Several approaches to learning in games have been studied recently in the literature; here we focus on Q-learning.

In Q-learning, the defender and attacker are able to converge to optimal policies despite not knowing transition probabilities, by iteratively updating a so-called Q-function. Such techniques have been studied thoroughly in papers concerning Markov Decision Processes (MDP) and Dynamic Programming (DP). Q-learning in games, on the other hand, has been introduced by Littman in [11] and more thoroughly discussed in [13]. In [12], convergence results are obtained for zero sum games by applying results from Littman and Szepesvari [13], an approach we take in our framework as well.

The attacker and defender players progressively update their strategies to arrive at the saddle point equilibrium. Recall that in the previous section we made use of a cost-to-go function Q_t, that eventually converges to the optimal cost-to-go Q^*, which satisfies

$$\begin{aligned} Q^*(a,d,s) = G_{a,d}(s) + \alpha \sum_{s'\in S} M_{s,s'}(a,d) \\ \min_{p^D(s')} \max_{a} \sum_{d\in A^D} Q^*(a,d,s')p^D_d(s'). \end{aligned}$$

Unlike the previous section, however, agents no longer know transition probabilities and cannot obtain this function by using equations (6) and (7). Instead, as in [11], we again make use of a Q-function to keep track of cost-to-go in terms of current state s, attacker action a and defender action d, but with updates based on experience. We focus on the defending agent. After experiencing the combination $(s_t, a_t, d_t, s_{t+1}, c_t)$ where c_t is the cost incurred at time t, and the other parameters denote the usual quantities, the defender updates the function:

$$\hat{Q}^d_{t+1}(a_t,d_t,s_t) = (1-\beta_t(a_t,d_t,s_t))\hat{Q}^d_t(a_t,d_t,s_t) \quad (8)$$

$$+ \beta_t(a_t,d_t,s_t)(c_t + \alpha V^d_t(s_{t+1})) \quad (9)$$

where

$$V^d_t(s') = \min_{p^D(s')} \max_{a} \sum_{d\in A^D} \hat{Q}^d_t(a,d,s')p^D_d(s')$$

where $\hat{Q}^d_t$ is the Q-function for the defending agent at time t and the parameter β_t indicates how rapidly the agent updates the Q-function with new information. The attacker, since the agents are playing a zero-sum game, has a similarly defined update for its own function, $\hat{Q}^a_t$. Note that the components of $\hat{Q}_t$ which are not in the observation combination remain unchanged from step t to $t+1$. We no longer make use of transition probabilities, but use new information sequentially as the game evolves. The attacker keeps track of its own Q-function, but we need only specify one function since it is a zero sum game ($\hat{Q}^a_t = -\hat{Q}^d_t$). The function V_t can be computed using a Linear Program in the same way as in the previous section.

We now turn to the issue of convergence - will these updates result in the attacker and defender eventually having the optimal function Q^*? As shown in [11] for general Markov games and in [12] for zero-sum games, there is indeed convergence under certain conditions on the learning rate β_t:

1) $0 \leq \beta_t(s,a,d)$
2) $\sum_{t=0}^{\infty} \beta_t(s,a,d) = \infty$, and $\sum_{t=0}^{\infty} \beta^2_t(s,a,d) < \infty$ with probability 1.
3) $\beta_t(s,a,d) = 0$ if $(s,a,d) \neq (s_t,a_t,b_t)$

If these are true, then the values defined by (8) converge to Q^*, the optimal Q-function with probability 1. As in [12], the norm for Q-functions is defined as follows: for a fixed state s, we let

$$||Q(s,\cdot,\cdot)|| = \max_{a\in A^{\mathcal{A}}, d\in A^{\mathcal{D}}} Q(s,a,d)$$

and the distance between Q-functions is given by

$$||Q_1 - Q_2||_\infty = \max_{s\in S} ||Q(s,\cdot,\cdot)||$$

In our context, $\hat{Q}^d_t \rightarrow Q^*$ with probability 1; the attacker and defender eventually learn their optimal strategies as time goes on, a desirable property.

V. Illustrative Example

We now illustrate the theory that has been thus far developed with an example scenario. In this example, a company sells IT and communication services to its customers. Due to the presence of malicious attackers hoping to disrupt business operations or stealing customer data, however, the underlying hardware and software components necessary to conduct the transaction can be compromised. Interdependencies between these components make the management of this risk complex, and therefore ripe for the application of our new analytical tools.

A. *Components and Costs*

Let us consider the scenario in which a company D sells its customers IT services. Due to having numereous customers, this process is repeated at each given time step over a business operating time frame. For the purposes of the business, the problem of interest is a continuous (repeated) one over a long time period so that we apply the infinite horizon framework of Section II. Malicious agents (collectively referred to as A) aim to attack business components so as to disrupt service and cause damage to D. Each attack is costly for the malicious agent as well, as it takes effort to conduct the attack and there is also a risk of being caught, if unsuccessful. If successful, however, D is pushed into a state of higher risk so that the business transaction is more likely to be interrupted. Such interruptions are costly for D since they represent both a loss of revenue as well as reputation.

In this example, D is composed of three business units which D relies on to ensure smooth functioning of the service. Unit f_1 is a database system keeping records of the customers. Unit f_2 corresponds to the device manufacturing system and unit f_3 is the network infrastructure involved in delivering service to the device. These units, while separate, are highly dependent on one another since several different technologies must come together to perform the aimed task (Fig. 3). We can place these units in a graph with the addition of a node corresponding to a risk-sink. The afore mentioned dependencies govern how much risk is present in each node. As mentioned, however, it is convenient to move to a Markov model so that we have a state space S where each element contributes to a region of risk.

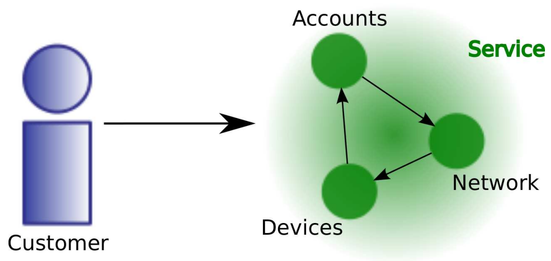


Fig. 3. A customer purchases a service that relies on multiple components which affect each other.

We use parameters for the Markov chain associated with this example as specified in the tables below. The risk states are $S = \{s_1, s_2, s_3\}$, and each has two matrices associated with it, one for the costs of the zero-sum game played at that state and the other for transition probabilities. Here we have written this as one matrix, where the first number defines the zero-sum game and the triple following it defines the transition probabilities. We suppose that defender and attacker each have three actions available.

For state s_1, a state of relative safety, we let the costs be lower than in the other states. The action space of the defender corresponds to greater overall levels of investment in defense d_i as i increases. The attack space, on the other hand, has been chosen to reflect an attack on each business unit.

	a_1	a_2	a_3
d_1	1, (0.5,0.4,0.1)	2, (0.3,0.4,0.3)	3, (0.5,0.1,0.4)
d_2	3, (0.7,0.2,0.1)	2, (0.4,0.4,0.2)	2, (0.6,0.1,0.3)
d_3	5, (0.8,0.1,0.1)	5, (0.6,0.2,0.2)	5, (0.7,0.1,0.2)

For s_2, we consider a risk region in which there is a moderate level of risk for f_2 (device manufacturing) and f_3 (network infrastructure), but the risk level for f_1 (data base) is relatively low. The transitions reflect the level of defense investment and the target of attack, as well as the current state.

	a_1	a_2	a_3
d_1	7, (0.2,0.7,0.1)	9, (0.1,0.5,0.4)	3, (0.2,0.1,0.7)
d_2	5, (0.4,0.6,0.0)	5, (0.3,0.4,0.3)	7, (0.4,0.3,0.3)
d_3	4, (0.6,0.4,0.0)	4, (0.5,0.3,0.2)	9, (0.7,0.1,0.2)

And finally let us take $s = 3$ to correspond to a state in which the risk level is higher for f_2 and f_3 but lower for f_1. This is reflected in the table below.

	a_1	a_2	a_3
d_1	3, (0.6,0.3,0.1)	7, (0.2,0.5,0.3)	6, (0.1,0.4,0.5)
d_2	5, (0.7,0.3,0.0)	5, (0.3,0.4,0.3)	5, (0.4,0.3,0.3)
d_3	7, (0.9,0.1,0.0)	5, (0.6,0.3,0.1)	9, (0.7,0.1,0.2)

Fig. 4 gives a pictorial representation of the interaction between the attacker and defender as they influence the risk state. The next state of the Markov chain depends on the actions taken by A and D at the current time.

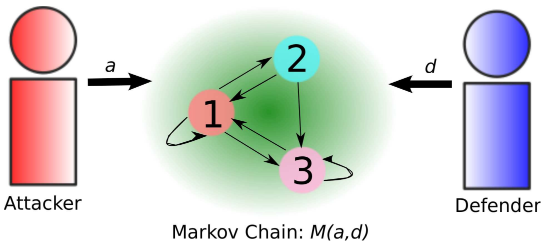


Fig. 4. An attacker and defender influence the Markov chain of risk states.

B. *Numerical Results*

We now present numerical results of our example. The example in a Q-learning framework may be simulated, with updates to the Q-function eventually converging to the optimal Q-function. Plotting the value of the game reveals that several thousand iterations are needed for convergence in this example, but the value of the game reaches 10% of convergence relatively quickly (Fig. 5). When agents are informed a priori of the Markov model parameters (no learning), the optimal Q-function can be determined from the iteration given in Section III within 100 iterations.

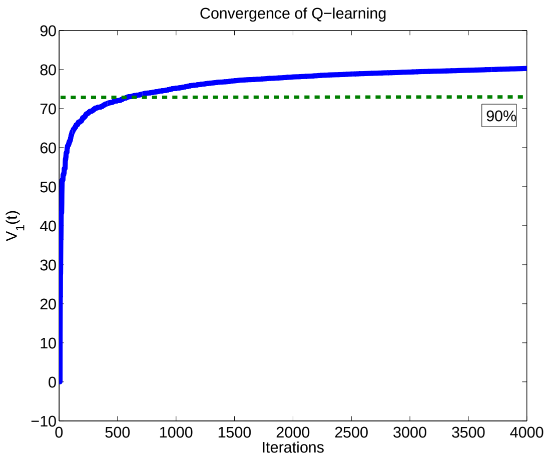
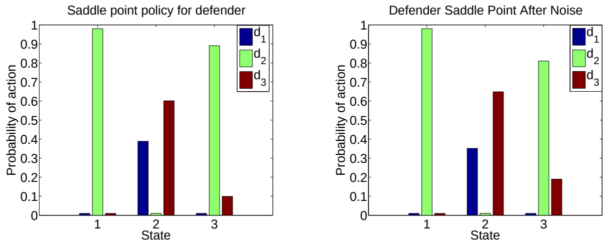


Fig. 5. Convergence of game value for state 1 with Q-learning.

Let us also calculate the optimal policy. The policy consists of a probability distribution for each action while in each state, for each agent. Here, we give the optimal policy for the defender agent in a bar graph depicting the probability of choosing each action while in each state (Fig. 6).

We also consider a scenario in which the attacker and defender have inaccurate estimates for the costs of the game. Specifically, we perturb the values in the cost matrices given above to within ten percent of their true values.

The resulting graph in (b) of Fig. 6 illustrates that the calculation of the optimal policy is robust to inaccurate estimates of costs. Indeed, in practice the defender and attacker will not have access to the true cost matrix; the policies they apply using the methodology above, however, are close to the policies obtained by having access to the true costs.



(a) Optimal policy

(b) With noisy cost estimates

Fig. 6. Bar graph representation of optimal defender policies

VI. Conclusion

In this paper, we have consider the problem of risk management in IT organizations through the lens of game theory, building on the mathematical foundations of [5], [7] capturing risk management in a quantitative framework. Our model takes risk management and considers several layers of abstraction. First, the Risk-Rank dynamics are introduced where the risk diffuses from one business unit or process to another. We then move to a Markov model in which states represent risk regions and transitions are probabilistically defined. Subsequently, we continue by considering a dynamic zero-sum sum game to model the interactions between attacker and defender players. In the last layer of our model, we introduce Q-learning to capture the scenario in which agents do not have access to the parameters of the Markov model.

This framework can serve as an aid to decision makers that must allocate limited resources for security. Indeed, each organization must balance the trade-off between investment in security and risk to vital business units from malicious agents. As infrastructures and computing systems become more complex, computer-aided decision making may find greater use, where our framework can play a role. Future work may consider dealing with very large state spaces as often appear in this application area.

Acknowledgements

The authors thank Christine Roesner and Karl-Friedrich Thier for the helpful discussions. This work is supported by the Deutsche Telekom Laboratories. The authors also thank Quanyan Zhu for input on the Q-learning simulation.

References

[1] General Accounting Office. Information Security: Computer Attacks at Department of Defense Pose Increasing Risks. GAO/AIMD-96-84, May, 1996.

[2] K. Dillard, J. Pfost, and S. Ryan, “Microsoft MSSC and SCOE: The security risk management guide,” Online, 2006. [Online]. Available: http://www.microsoft.com/mof

[3] P. R. Garvey, “Analytical Methods for Risk Management: A Systems Engineering Perspective,” ser. Statistics: a Series of Textbooks and Monographs. Boca Raton, FL, USA: Chapman and Hall/CRC, 2009.

[4] R. A. Miura-Ko and N. Bambos, “Securerank: A risk-based vulnerability management scheme for computing infrastructures,” in *Proc. of the IEEE Conference on Communication ICC*. IEEE, 2007.

[5] T. Alpcan and N. Bambos, “Modeling dependencies in security risk management,” in *CRiSIS*. IEEE, 2009.

[6] A. Langville and C.Meyer, “A Survey of Eigenvector Methods for Web Information Retrieval,” SIAM Review, vol. 47, no. 1, pp. 135 - 161, 2005.

[7] J. Mounzer, T. Alpcan, and N. Bambos, “Dynamic control and mitigation of interdependent IT security risks,” in *Proc. of the IEEE Conference on Communication (ICC)*. IEEE Communications Society, May 2010.

[8] S. Guikema, “Game Theory Models of Intelligent Actors in Reliability Analysis,” In Intl. Series in Operations Research & Management Science, J.M.P Cardoso and and P.C Diniz., Springer US, 2008; Vol. 128, pp. 1-19.

[9] S. Guikema, T. Aven, “Assessing risk from intelligent attacks: A perspective on approaches,” Reliability Engineering & System Safety, Vol. 95, No. 5, May 2010, pp. 478-483.

[10] T. Alpcan and T. Basar. *Network Security: A Decision and Game Theoretic Approach*. Cambridge University Press, 2011.

[11] M. L. Littman, “Markov games as a framework for multi-agent reinforcement learning,” in *Proc. of the Eleventh Intl. Conference on Machine Learning (ICML)*, pp. 157 - 163, September 1994.

[12] Q. Zhu and T. Basar, “Dynamic Policy-Based IDS Configuration,” in *IEEE Proc. of 47th Conf. on Decision and Control (CDC)*, 2009.

[13] C. Szepesvari and M. L. Littman, “A unified analysis of value-function-based reinforcement-learning algorithms,” Neural Computation, no. 11, pp. 2017-2059, 1999.