Risk Management for IT Security: When Theory Meets Practice

Anil Kumar Chorppath
Technical University of Munich
Munich, Germany
Email: anil.chorppath@tum.de

Tansu Alpcan
The University of Melbourne
Melbourne, Australia
Email: tansu.alpcan@unimelb.edu.au

***Abstract*—A Layer-Based Risk Tool (LBRT) for IT security management in a corporate environment is presented and discussed. The Risk-Rank algorithm is modified for implementation in this tool by taking practical considerations into account. The focus is shifted to a security requirement-based approach during actual assessment of operational risk in the organization and absolute risk values are computed instead of relative risk probabilities. In addition, a risk mitigation algorithm is proposed to find the optimum set of measures under certain budget constraints. A dynamic programming formulation is presented and a shortest path solution is obtained based on Dijkstra's algorithm. The risk assessment and mitigation algorithms are illustrated and evaluated with numerical examples.**

I. Introduction

Management of IT security risks is a young and vibrant field with substantial research challenges and opportunities. Early IT and security risk management research has been mostly empirical and ad-hoc in nature. The goal of the analytical approach is to go one step further and create a solid quantitative foundation for security and IT risk management [1], [2]. Therefore, quantitative analysis of risks in complex systems of an organization is an important necessity. In this paper, we present a risk assessment and mitigation tool and discuss how earlier theoretical results [3] developed have been modified for implementation within this tool.

We modify the Risk-Rank algorithm [3] for implementation in a Layer-Based Risk Tool, which is developed within an IT-intensive complex corporate environment such that various ontologies of risk factors are arranged in different layers. The algorithm has to be tweaked to take into account practical considerations such as following a security requirement approach instead of an attack-based one. In many IT organizations an important issue is the compliance with the security requirements. Therefore, in practice the level of compliance with respect to the widely accepted security requirements and best practices [4] plays the primary role in security and IT risk assessment.

Subsequently, a risk mitigation algorithm based on *dynamic programming* [5] is developed to select a set of measures which minimize the aggregate risk subject to budget constraints. This turns out to be equivalent to solving a shortest path algorithm in a graph. This optimized risk mitigation algorithm is useful for the managers of IT organizations who optimize investment decisions and measures for risk reduction. The risk mitigation algorithm also accounts for the dependencies between the objects while computing the individual costs of the measures.

This work has been supported by Deutsche Telekom Laboratories.

A. Related work and Contribution

There already exist numerous works on system risk approach for security decision making [6]. Recently, there has been growing interest in quantitative and analytical methods for risk management, for eg. [1], [2], [7]. Several metrics for IT-Security risk management have been proposed in [8].

A Risk-Rank algorithm has been introduced to model the interdependencies of the objects and the risk cascading and diffusion effects in [3]. It is developed using diffusion processes over graphs [9], and corresponds to a modified version of the Page-Rank algorithm [10] used by the Google search engine. There are already generic and domain oriented risk assessment and mitigation tools such as the Virtual Machine Servicing Tool (VMTS), which has been released by Windows for IT security [11]. In [12] a risk mitigation strategy through Markov decision process has been investigated. It aims to enable IT managers to perform more comprehensive evaluations of their risk exposures with increased effectiveness through analytical methods.

The contributions of this paper are:

1) A Layer-Based Risk Tool (LBRT) is presented for IT security management taking into account all complexities in a corporate environment.
2) The paper studies integration of modified Risk-Rank algorithm into LBRT, based on absolute risk values and security requirement approach for layered systems.
3) A risk mitigation algorithm is proposed considering the dependencies of objects and other practical aspects.

II. Layer-Based Risk Tool

The Layer-Based Risk Tool (LBRT) is a software package developed for internal use in an IT-intensive corporate environment to help assess and mitigate IT security risks, and ensure compliance with security requirements. It can be seen that there are different layers of Locations, Teams, Data and Others. The objects are distributed in these layers. It models classes of objects such as routers, servers and other devices

978-1-4673-0229-6/12/$31.00 ©2012 IEEE

as basic entities. Clearly, many of these objects depend on each other operationally to realize certain business processes. Therefore, the risks on these objects may propagate from one to another due to cascading effects. Initially, however, these dependencies have not been taken into account and each risk factor is handled independently on each object class.

The ontology of the LBRT is depicted in Figure 1. It defines a basic security ontology for a better understanding of security concepts and their relationships. Objects such as routers, servers, etc. are the main entities in the proposed model. Each object is mapped to a list of *security requirements* that must be fulfilled. The *level of completeness* is a value that quantifies to which extent the security requirements are already met in the designated object. Security requirements are measures that minimize the impact and likelihood of various threats. A catalog of security threats exist which provide foundation for the risk calculation. Two values exist for the risks on objects: *current risk* and *initial risk*. The initial risk is the risk associated with an object without considering which security requirements are already implemented. Current risk is the risk associated with an object after the level of completeness evaluation of the security requirements. Security requirements are grouped in categories and mapped also to *asset types* which are further mapped to objects. Specific risk and security requirements considered in the tool are *Confidentiality*, *Integrity* and *Availability* (CIA). At this point, each of these aspects are handled independently.

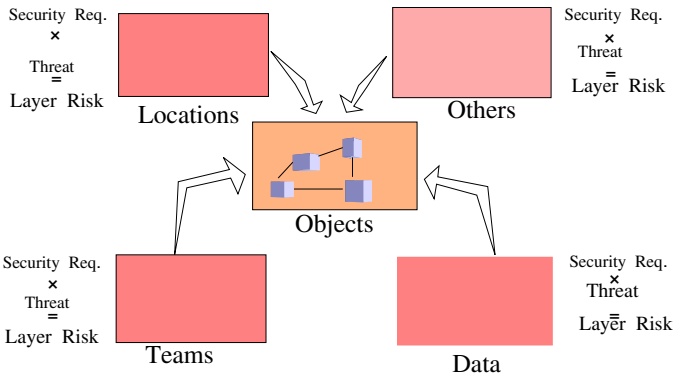

Fig. 1. The Ontology of the Layer-Based Risk Assessment Tool.

As a result of our interaction with the developers of the LBRT software package, we have identified two main directions of collaboration. First, we have introduced a quantitative approach to handle dependencies between objects and risk factors to be implemented in LBRT. Secondly, we have investigated an analytical and model-based risk mitigation framework to improve the heuristic ones already implemented. These two topics are discussed in the next subsequent sections in detail.

III. Risk Diffusion Model

Due to dependencies, security risks may diffuse from one object to another in a complex and networked systems such as the one at hand. This diffusion process may in turn cause cascaded risks from the immediate ones. The Risk-Rank (RR) algorithm [3] is a general algorithm that can be taken as a starting point to analyze both the immediate risks and diffusion effects. The RR algorithm ranks the nodes in the system based on their risk probabilities, while taking into account cascade of risk and enable the balance between direct and induced risks to manage the relevant tradeoff. It computes the final risk vector according to an iteration starting from the normalized initial risk vector.

Consider K objects or asset classes and a finite number of, N, risk diffusion steps. The matrix A whose elements belong to $\mathbb{R}^+$ models the dependency between the risks on objects as a representation of a directed graph. Its elements have a positive value when the corresponding objects have risk dependency from one to another. It should be noted that the diagonal elements are zero. Otherwise, the risk on an object would multiply itself regardless of the dependencies and grow without bound.

Here, we depart from the RR algorithm and capture the evolution of risks on interdependent objects with absolute values instead of relative risk probabilities. This change was motivated by the fact that LBRT tool focuses on absolute values of risks rather than relative risk probabilities. The objective still is, however, modeling how risks propagate due to diffusion processes between objects. The absolute values of risks give an indication of the degree to which the security requirements are met and they will help also in the mitigation stage to identify the correct set of measures. Therefore the algorithm considers a security requirement approach instead of an attack-based one in the Risk-Rank model and in many IT organizations an important issue is the compliance with the security requirements. Also it is important to note that various objects are arranged in different layers in the LBRT tool but the RR algorithm consider only one layer.

Let the K-dimensional initial risk vector to be $X(0)$, and the risk values of the objects to be given by $X(n)$ at the diffusion step n. Then, the risk evolution can be simply modeled as a linear system

$$X(n+1) = A * X(n), \text{for } n = 1, .., N-1. \qquad (1)$$

In fact, some risk diffusion processes are much more limited, i.e. some objects transfer their risk only in one or two steps. Therefore, an even more realistic model is a time-varying one

$$X(n+1) = A(n) * X(n), \text{for } n = 1, .., N-1, \qquad (2)$$

where some of the entries of the time-varying dependency matrix $A(n)$ needs to be zeroed out. In many cases, the total number of risk diffusion steps, N, is bounded above by 5 to 10 at most.

An example to explain how the dependency matrix A is obtained is given next. We have the following $K = 4$ objects denoted as CE, AGS, PE and RR, which represent different types of routers. The initial risk vector $X(0)$ for this example is taken as, $X(0) = [2.2, 2.1, 3.0, 0.5]^T$. All these values are obtained from assessments of field experts.

The dependency between the devices has the following

grade of impact (1 = minimal effect, 2 = medium effect, 3 = service offline / device disconnected). Let us map the impact to numbers between 0 and 1, i.e. $1 \Longrightarrow 0.1, 2 \Longrightarrow 0.2, 3 \Longrightarrow 0.3$. Therefore,
CE $\Longrightarrow$ AGS Impact: 0.2
AGS $\Longrightarrow$ PE Impact: 0.3
CE $\Longrightarrow$ PE Impact: 0.3
PE $\Longrightarrow$ RR Impact: 0.3.
Thus, the corresponding dependency matrix A for this example is

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0.2 & 0 & 0 & 0 \\ 0.3 & 0.3 & 0 & 0 \\ 0 & 0 & 0.3 & 0 \end{pmatrix}. \quad (3)$$

Note that the zero elements in the matrix correspond to the independency of the respective objects. In the simulation section the evolution of risks are computed for this case.

Let us define and evaluate some risk metrics which are useful for the decision making of risk managers. The *risk criticality* of an object is the sum of the absolute risk value of the object and the total transfer of risk by that object to other objects due to dependency. Let us calculate the total risk transfer of the 4 objects for the example given above. Let $X1 = A * X(0)$. Then $X1 = [0.8, 1.26, 1.32, 0]^T$. Then, the total risk transfer of PE $= 0.8 + (2.1 + 1.26) * 0.6 = 2.7$, AGS=1.26, CE= 1.2 and RR=0. Also, the risk criticality of PE=5.7, AGS=3.36, CE= 3.4 and RR=0.5. We can observe that even if an object has low absolute risk value, the risk criticality can be high due to the risk transfer it causes due to the dependency.

Monte Carlo Methods

It is not realistic to assume that the risk values of each object can be obtained exactly. The dependency values are based on information from not so reliable sources such as the employees or past incidents. Therefore, it is realistic to assume that these values can be known only approximately and lie on an interval with some error from the exact value. As a way to capture this uncertainty, the values of the elements in A are generated randomly for sufficiently large number of times on certain given intervals [13].

The final values of risks can be calculated for each randomly generated sample value of A. One can utilize the average over all these different final risk values or take the minimum or maximum of final risk values as a basis depending on the goal. In an optimistic case, the minimum risk value or the mean value minus some constant times standard deviation can be used. Most likely case is the mean value and in the worst case approach the maximum risk values or mean value plus some constant times standard deviation can be used. The results illustrating these calculations for a numerical example are plotted in the simulation section.

IV. Risk Mitigation

Once the risks on objects are identified and quantified, the risk mitigation measures are employed to reduce the risks below a threshold level. Appropriate actions should be taken to perform risk mitigation and control as the next step once the assessment phase is completed. An optimized risk mitigation strategy can be used by the decision makers to improve the investment decisions for the risk mitigation or reduction measures. Hence, it is shown how a modified Risk-Rank approach can also be used to evaluate risk mitigation strategies. More specifically, the question of how to control the risk diffusion process defined in the previous subsection over time is addressed in order to achieve a more favorable risk distribution across the assets of an organization. The different aspects of risk mitigation are visualized in Figure 2.

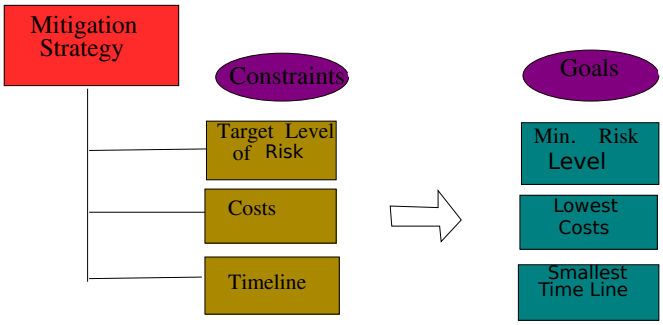

Fig. 2. Risk Mitigation Approach of LBRT.

The mitigation measures suggest policies and rules for allocation of security resources or updating system configuration. These actions change the dependencies in the organization, and hence directly affect the structure. The objective of risk control is to achieve a more favorable risk values and reduce the risk mitigation costs accumulated over time. The theory of Markov Decision Process has been used to model the transition of the state of the objects from one to another while applying mitigation strategies in [12]. We follow a dynamic programming approach in which the cost of each mitigation measure is the cost of transition from one state to another. While the approach in this paper is still based on dynamic programming, there are some differences between the model here aimed for practical implementation and the one in [12].

It is important to note that the dependency structure between the objects and the diffusion effects discussed in the section III can be integrated into this formulation. The diffusion equation (1) can be used to obtain the effect of each measure in all the objects considering the cascading effects. Then, these calculated total risk reduction factor on all the objects can be used for the mitigation algorithm.

Goals and Constraints

Let us consider the available mitigation measures are $M_1, M_2, ..., M_L$. The time is divided into discrete slots and the mitigation problem is considered for a finite horizon of T time periods. Here, we consider the following three metrics associated with each measure:

1) **Cost or Budget of implementation** defines how much must be spent for the implementation of the countermeasure.
2) **Timeline** defines the period of time required for the im-

plementation of the countermeasure, taking into account the start date and the end date.

3) **Risk-Reduction Factor** which is defined as the amount of risk that got reduced after the implementation of the measure.

Let C_i be the cost or budget, t_i be the time line and r_i be the risk-reduction factor associated with the i^{th} counter measure. Let R_t be the risk at the end of time period t. The goal is to select the measures from a set of given measures to find the cost-effective mitigation strategy. The constraints are based on the

1) costs associated with each measure
2) time line to complete the measure
3) target level of risk to be achieved.

Sometimes is not possible to reach the desired target level of risk with the specific costs and one needs to adjust the costs or time line in order to obtain a feasible solution. In order to reach the target level of risk within the given time line one may have to increase costs. A related objective is to investigate all possible feasible solutions given different restrictions.

Assumption: As a starting point, we assume that a measure is taken only once and all of the measures finish in the same time.

Problem formulation: the optimization problem is to minimize the final risk subject to the budget constraints by selecting the proper subset of measures within the finite time line, i.e,

$$\min_{B \in M} R_T$$

subject to

$$\sum_t C^t \leq C^* \tag{4}$$

where M is the set of all measures and B is the set of measures taken in the time line until time T, and C^* is the total budget.

Now, we redefine the problem as an optimization over discrete state system with finite states. Let r_{ij}^t be the risk reduction at stage t from state $i \in S_t$ to state $j \in S_{t+1}$. A finite state system can be defined based on the states. Let any state i belong to a state space S_t for each instant t. Each state at an instant is associated with the risk at that point. The measures can be considered as the controls taken on each state, and depending on each one there is a transition to another state with a new risk value. For finite number of measures and due to the assumptions above, the number of possible states at an instant is finite and in the same order of magnitude as the number of measures. The effective cost of transition from state i to j in the time instant t is given by

$$a_{ij}^t = c_{ij}^t - r_{ij}^t.$$

This problem is equivalent to finding the shortest path with the minimum cost in a graph using Dynamic Programming (DP) algorithm [5]. We create an artificial terminal node and find the shortest path between this node and the source node. The DP algorithm takes the form

$$J_T(i) = a_{iF}^T, \forall i \in S_T,$$

$$J_t(i) = \min_{j \in S_{t+1}} [a_{ij}^t + J_{t+1}(i)], \forall i \in S_t, t = 0, 1, 2....T-1,$$

subject to the constraints in equations (4). a_{iF}^T is the effective cost of transition from state i to the terminal node F in the time instant T and $J_t(i)$ is the total cost to the terminal node from the node i at the time t. The optimal cost is $J_0(s)$ where s is the source node.

The shortest path with the minimum cost $J_0(s)$ can be obtained using a standard algorithm such as Dijkstra's algorithm [14]. Essentially, the algorithm gives the set of countermeasures out of the whole set, which minimizes the risk at the end of the time line subject to the budget constraints. The shortest path gives the best budget to risk reduction tradeoff.

V. Numerical Examples

In this section, first we plot the risk evolution for the example in Section III for which the dependency matrix A is given in equation (3) .

In Figure 3, the risk evolution for the 4 devices are shown. As explained in the example in Section III the initial risk vector is fixed and the iteration given by equation (1) is run for 3 diffusion steps. In Figure 4, the risk evolution

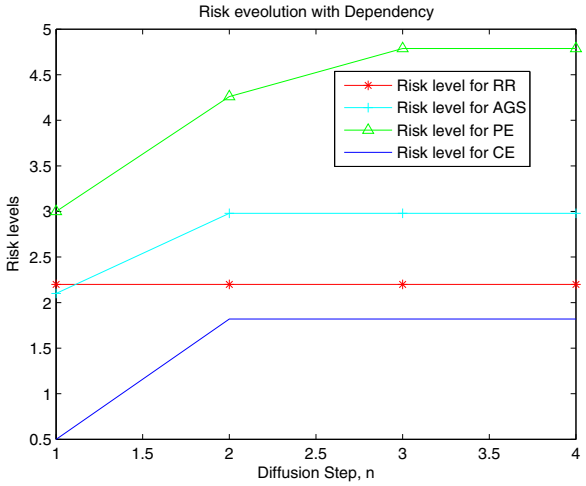

Fig. 3. Risk evolution over time steps.

computed using the Monte Carlo method is depicted. The 10000 samples are obtained by uniform random sampling of the hypercube using intervals of 0.5 wide and the average values for each diffusion steps are obtained as explained in Section III. Standard deviation from the mean value is also shown which accounts for the error due to the randomization in the input values of matrix A.

Next the risk mitigation algorithm as a shortest path algorithm is illustrated in Figure 5 for one object. First, the cost matrix is constructed for the all possible paths between the states in the graph. The paths (sequences of risk mitigation measures) from the source and destination which violate the budget constraint are removed. Next, along with the source

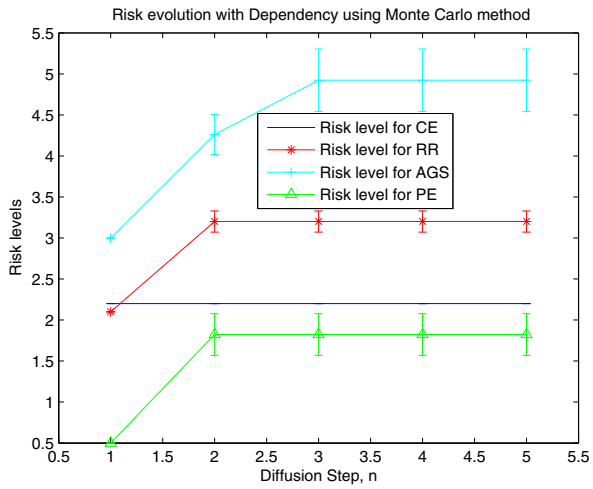

Fig. 4. The evolution of the risk obtained by the Monte Carlo method.

and destination node the cost matrix is input to the Dijkstra's algorithm. Out of 3 possible measures M_1, M_2, M_3, 2 optimum measures are selected as the shortest path in the graph. The shortest path between the initial node 1 and the artificial terminal node 8, having the lowest cost is shown as the red line.

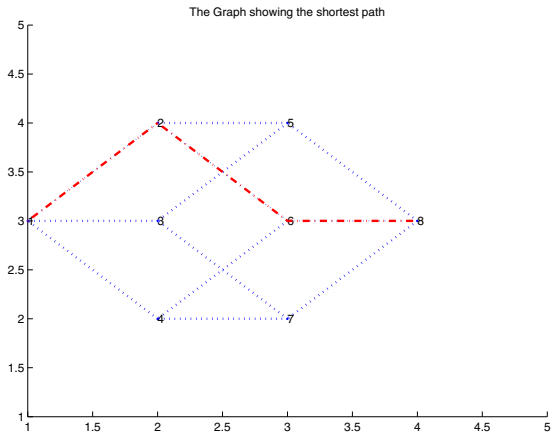

Fig. 5. The graph of the risk measures with the shortest path of minimum cost.

VI. Discussion and Conclusion

A Layer-Based Risk Tool (LBRT) for IT security management in a corporate environment is presented and discussed. During the interaction with the developers of the LBRT, it has become clear that the Risk-Rank algorithm presented in [3] needs to be modified both for risk assessment and mitigation for possible implementation. During the assessment phase, the approach is shifted to a security requirement-based one instead of an attack-based one. Furthermore, the relative risk probabilities are replaced by absolute risk values, and hence, a (time-varying) linear system formulation is obtained to compute transfer of risks between objects.

An important practical issue in assessment of risks is the lack of data or relatively low reliability of it. Therefore, instead of using single absolute risk dependency values ranges of values are considered to account for ambiguity in the inputs. Subsequently, Monte Carlo methods are utilized to obtain mean, best case, and worst-case solutions.

The risk mitigation algorithm to find the optimum set of measures within the budget constraints also deviates from the earlier Markov decision process-based one in [12]. Although, it is still a dynamic programming formulation, the more practically applicable solution is a shortest path one which is obtained based on Dijkstra's algorithm.

The risk assessment and mitigation algorithms are illustrated and evaluated with numerical examples. Future directions include considering the CIA requirements together as dependent requirements and including more practical considerations like different completion time for different measures and ordering of measures etc.

VII. Acknowledgement

We thank Virgilio Giner-Albarracin and Mario Schroen for discussions and suggestions.

References

[1] P. R. Garvey, *Analytical Methods for Risk Management: Analytical Methods for Risk Management*. Chapman and Hall/CRC, 2008.

[2] T. Alpcan and T. Basar, *Network Security: A Decision and Game Theoretic Approach*. London, UK: Cambridge University Press, 2010.

[3] J. Mounzer, T. Alpcan, and N. Bambos, "Integrated security risk management for IT-intensive organizations," in *Proc. of 6th Intl. Conf. on Information Assurance and Security (IAS 2010)*, Atlanta, GA, USA, August 2010.

[4] A. G. Tarantino, *Governance, Risk, and Compliance Handbook*. Hoboken, NJ, USA: John Wiley & Sons, 2008.

[5] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. 1 (Optimization and Computation Series)*, 2nd ed. Athena Scientific, 2000.

[6] D. W. Straub and R. J. Welke, "Coping with systems risk: security planning models for management decision making," *Management Iinformation Systems Quarterly*, vol. 22, no. 4, pp. 441–470, 1998.

[7] R. A. Miura-Ko, B. Yolken, J. Mitchell, and N. Bambos, "Security decision-making among interdependent organizations," in *Proc. of the 21st IEEE Computer Security Foundations Symposium*, Pennsylvania, USA, 2008, pp. 66–80.

[8] S. Fenz, "Ontology-based generation of IT-security metrics," in *Proc. of the 2010 ACM Symposium on Applied Computing*, ser. SAC '10. New York, NY, USA: ACM, 2010, pp. 1833–1839. [Online]. Available: http://doi.acm.org/10.1145/1774088.1774478

[9] T. Alpcan and N. Bambos, "Modeling dependencies in security risk management," in *Proc. of 4th Intl. Conf. on Risks and Security of Internet and Systems (Crisis)*, Toulouse, France, October 2009.

[10] A. Langville and C. Meyer, "A Survey of Eigenvector Methods for Web Information Retrieval," in *SIAM Review*, 2005, pp. 135–161.

[11] [Online]. Available: http://technet.microsoft.com/en-us/library;http://cc501231.aspx

[12] 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)*, Cape Town, South Africa, May 2010.

[13] R. Tempo, G. Calafiore, and F. Dabbene, *Randomized Algorithms for Analysis and Control of Uncertain Systems*. London, UK: Springer-Verlag, 2005.

[14] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, pp. 269–271, 1959.