Intelligent Risk Identification and Analysis in IT Network Systems

Masoud Mohammadian

Intelligent Risk Identification and Analysis in IT Network Systems

Masoud Mohammadian

University of Canberra
Faculty of Information Sciences and Engineering
Canberra, ACT 2616, Australia
masoud.mohammadian@canberra.edu.au

Abstract. With ever increasing application of information technologies in every day activities, organizations face the need for applications that provides better security. The existence of complex IT systems with multiple interdependencies creates great difficulties for Chief Security Officers to comprehend and be aware of all potential risks in such systems. Intelligent decision making for IT security is a crucial element of an organization's success and its competitive position in the marketplace. This paper considers the implementation of an integrated attack graph and a Fuzzy Cognitive Maps (FCM) to provide facilities to capture and represent complex relationships in IT systems. By using FCMs the security of IT systems can regularly be reviewed and improved. What-if analysis can be performed to better understand vulnerabilities of a designed system. Finally an integrated system consisting of FCM, Attack graphs and Genetic Algorithms (GA) is used to identify vulnerabilities of IT systems that may not be apparent to Chief Security Officers.

Key words: Network Security, Risks Analysis, Intelligent Systems, Attack Graphs

1 INTRODUCTION

Current IT systems are accessed anytime from any locations using the Internet. This approach increases the security risks in ever increasing IT applications and networks. IT systems are complicated with large number of interdependent networks and IT facilities. In such systems there is a need for proactive and continuous security risk assessment, identification, verification and monitoring.

A graphical representation of an IT system can improve the understanding of the designer of a system and mitigate risks of attack to designed systems. Such a graphical representation can assist in documenting security risks and identifying possible paths attackers may consider to attack a system for their undesirable goals.

Attack graphs [1, 8] are designed after analyzing an IT system purpose, its components and any set of potential attacker undesirable goals. These goals may include system's disruptions, intrusion and misuse by an attacker [2, 3]. During design, implementation and verification of an attack graph possible attacks and undesirable goals of diverse attackers are considered. The skill, access, and goals of attackers are also considered. Attack graphs are created based on several notations. These notations are attacker's goals, trust boundaries, sub-goals and paths in an attack graph to reach attacker's goals from a part of a system.

Sub-goals may be used by attackers to move through a system to reach their goals. Paths through an attack graph are identified to reach attacker's goals. With complexity of existing systems drawing attack graphs are becoming increasingly difficult. Due to the complexity and difficulty of designing attack graphs an attack graph may be flawed. Attack graphs provides a graphical representation of an IT system which make it easier to understand however an attack graph does not provide any facilities to analyze and assess different risks and possible attacks that may exist in a systematic way. An attack graphs documents the risks known at the time the system is designed. However an attack graph does not provide facilities to perform concrete risk analysis such as what-if and scenario analysis to test the designed system for possible risk of attacks. In this paper, a Fuzzy Cognitive Map (FCM) is used with graph attacks to provide facilities that will enable the system architects to perform what-if scenario analysis to better understand vulnerabilities of their designed system. Fuzzy Cognitive Maps (FCM) [4, 5, 6] are graph structures that provide a method of capturing and representing complex relationships in a system. Application of FCM has been popular in modeling problems with low or no past data set or historical information [6]. This paper proposes also a Genetic Algorithm (GA) to automatically create attack scenarios based on the given IT system. These scenarios are then evaluated using a FCM and the results are provided for analysis and mitigation of security risks. The novelty of the proposed systems based on attack graph, FCM and GA is to provide the automatic facilities to identify security vulnerabilities in complex IT systems. Our proposed approach comprises of:

(i) a formal graphical representation of the IT systems using attack graphs,
(ii) conversion of an attack graph into Fuzzy Cognitive Maps (FCM) to allow calculation and what-if scenario analysis,
(iii) a GA which creates attack scenarios to be passed into a fuzzy cognitive maps to assess the results of such attacks.

Our novel approach proposes the use of attach graphs, FCM and GA to represent a give IT systems and to score each attack. In particular, a security risk value is assigned to every given attack. Such a measure of security risk provides the ability to re-assess such IT systems and modify such systems to reduce security risks.

2 Attack Graphs, Fuzzy Cognitive Maps and Genetic Algorithms

A graphical representation of a system can improve the understanding of the designer of a system and mitigate risks of attack to designed systems. Attack graphs [1, 8] are designed after analyzing a system purpose, its components and any set of potential

attacker undesirable goals. Attack graphs are created based on several notations. These notations are attacker's goals, trust boundaries, sub-goals and paths in an attack graph to reach attacker's goals from a part of a system. Attacker's goals are identified on an attack graph using octagons placed at the bottom of an attack graph. Trust boundaries separate components of a system that are of different trust levels. Sub-goals are represented using AND and OR nodes. An AND node is represented by a circle and an OR node is represented by a triangle. Sub-goals may be used by attackers to move through a system to reach their goals. Paths through an attack graph are identified to reach attacker's goals [1]. With complexity of existing systems creating attack graphs are becoming increasingly difficult. Due to the complexity and difficulty of designing attack graphs an attack graph may be flawed.

Attack graphs provides a graphical representation of a system which make it easier to understand however an attack graph does not provide any facilities to analyze and assess different risks and possible attacks that may exist in a systematic way. Fuzzy Cognitive Maps (FCM) [4, 5, 6] is graph structures that provide a method of capturing and representing complex relationships in a system. A FCM provides the facilities to capture and represent complex relationships in a system to improve the understanding of a system. A FCM uses scenario analysis by considering several alternative solutions to a given situation [9, 10]. Concepts sometimes called nodes or events represent the system behavior in a FCM. The concepts are connected using a directed arrow showing causal relations between concepts. The graph's edges are the casual influences between the concepts. The development of the FCM is based on the utilization of domain experts' knowledge. Expert knowledge is used to identify concepts and the degree of influence between them. A FCM can be used in conjunction with attack graph to provide the system architects with possibilities for what-if analysis to understand the vulnerability of the system and perform risk analysis and identification. Using FCM it is possible to identify paths through an attack graph to reach attacker's goals. Concepts in a FCM represent events and concepts are connected together with edges that describe relationships between concepts. These relationships increase or decrease of likelihood of a concept (i.e. event) to occur when other concept/s (events) occurs. Values on each edge in a FCM represent strengths or weakness of the relationship between concepts. The values on each edge are in the interval range of [–1, 1] which indicate the degree of influence of a concept to another concept. A positive value represents an increase in strength of a concept to another concept while a negative value indicates decrease in its influence to another concept. Each FCM has an activation threshold. This activation threshold provides the minimum strength required in a relationship to trigger and activate a concept. Drawing a FCM for a system requires knowledge of system's architecture. The activation level of concepts participating in a FCM can be calculated using specific updating equations in a series of iterations. A FCM can reach equilibrium or it can exhibit limit cycle behaviour. Once the system reaches equilibrium the decision-makers can use this information to make decisions about the system.

If limit cycle is reached decision-making is impossible. When limit cycle reached the experts are asked to consider the FCM and provide advice in changing the weights or one or more concepts of the FCM. The mathematical model behind the graphical representation of the FCM consists of a $1xn$ state vector I. This state vector

represents the values of the *n* concepts and nxn weight matrix W_{IJ} represents value of weights between concepts of C_i and C_j. For each concept in a FCM a value one or zero is assigned. One represent the existence of that concept at a given time and zero represent none-exist of the respective concept. A threshold function is used in FCM. The threshold function used in this paper is sigmoid function [6] as shown below:

$$C_i(t_{n+1}) = S\left[\sum_{K=1}^{N} e_{KI}(t_n) C_k(t_n)\right] \qquad (1)$$

Having constructed an attack graph, analysts can create a FCM based on the developed attack graph and allocate values to each edge and perform analysis of various risk and vulnerabilities in a given system using What-If analysis [4, 5]. An attack graph contains several notations as explained earlier. For an analyst to be able to perform What-If analysis an attack graphs must be converted into a FCM. A FCM have some representational limits. A FCM does not represent AND and OR operators. Therefore to convert an attack graph to a FCM graph, AND and OR operators needs to be removed without changing the design of the system. This needs careful consideration. Sub-goals on an attack graph are joined using AND/OR operators. Therefore to remove AND/OR operators each sub-goal using AND operator needs to be represented as a concept by joining all concepts that are joined using AND operator on a FCM. OR operators are removed and concepts joined by OR operator are directly connected to the subsequent concept of the OR node. Paths through an attack graph are represented using edges on a FCM with weight attached to them. Paths connect concepts in FCM. The weights on the edges are then assigned by the system architects accordingly. Genetic Algorithms [7] are powerful search algorithms based on the mechanism of natural selection and use operations of reproduction, crossover, and mutation on a population of strings. A set (population) of possible solutions, in this case, a coding of the attack scenarios for an IT system, represented as a string of zero and ones. New strings are produced every generation by the repetition of a two-step cycle. First each individual string is decoded and its ability to solve the problem is assessed. Each string is assigned a fitness value, depending on how well it performed. In the second stage the fittest strings are preferentially chosen for recombination to form the next generation. Recombination involves the selection of two strings, the choice of a crossover point in the string, and the switching of the segments to the right of this point, between the two strings (the cross-over operation). A string encoded this way can be represented as:

1	0	1	1	----------	1

C1 C2 C3 C4 C_n

Each individual string represents an attack scenario. Each scenario is evaluated by a FCM representing a given IT system based upon a fitness value which is specific to the system. At the end of each generation, (two or more) copies of the best performing string from the parent generation is included in the next generation to ensure that the best performing strings are not lost. GA performs then the process of

selection, crossover and mutation on the rest of the individual strings. The process of selection, crossover and mutation are repeated for a number of generations till a satisfactory value is obtained. We define a satisfactory value as one whose fitness value differs from the desired output of the system by a very small value. In our case we would like to find out the string that represents the most undesirable attack on our given IT system.

3 Simulation

In this paper a case study based on an IT system by S. Gupta and J. Winstead [1] is adapted. This case study will be used to represent how an attack graph can be converted into a FCM. The FCM graph created then is used to perform What-If analysis using GA. Figure 1 displays an attack graph presented by [1]. In this attack graph designer's goal was to create an IT system to protect sensitive data at a distributed set of sites with variety of constraints on development time, hardware availability, and limited business process changes [1]. The designers used encryption technology to encrypt financial information on customer sites using a set of symmetric keys distributed globally throughout the enterprise. All systems in this architecture are required to be able to read encrypted messages. The issue of physical protection of the system on customer sites was also considered. An important requirement of the systems was the availability of system in case of failure and lack of connectivity [1]. The designers with assistance of external reviewers of the system identified attack goals [1]. The attack goals identified were:

- unauthorized access to a sensitive file on the system for direct financial gain,
- unauthorized access to the encryption keys to gain access to sensitive data on other systems and
- unauthorized control of the host itself for use as a launching point for other attacks [1].

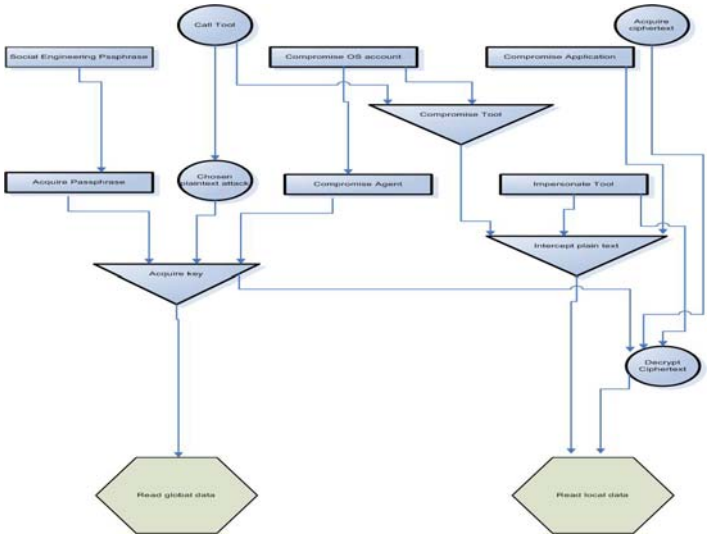

Fig1. Attack Graph for case study [1]

The attack graph in Figure 1 can be converted to a FCM (as shown in Figure 2) to provide the system architects with possibilities for what-if analysis and to understand the vulnerabilities and risks associated with this system. Using FCM it is now possible to identify and evaluate each path through FCM for each attacker's goals. The relationships details among all concepts in Figure 2 can be displayed using the following matrix E. The opinion of the experts and system designer is required to determine the weights of the different causal links and the initial activation level for each concept. In this scenario the author has carefully considered the system and provided the weights for the FCM shown in Figure 2. To simplify further and ease FCM in Figure 2 the following abbreviations are used for each concept: C1 = Social Engineering Passphrase, C2 = Call Tool and Chosen plaintext attack, C3 = Call Tool, C4 = Compromise OS account, C5 = Compromise Application, C6 = Acquire ciphertext and Decrypt Ciphertext, C7 = Acquire Passphrase, C8 = Compromise Agent, C9 = Compromise Tool, C10 = Impersonate Tool, C11 = Acquire Key, C12 = Read global data, C13 = Read local data. Now what-If analysis can proceed by using the matrix E. In this scenario the threshold is set to be 0.5.

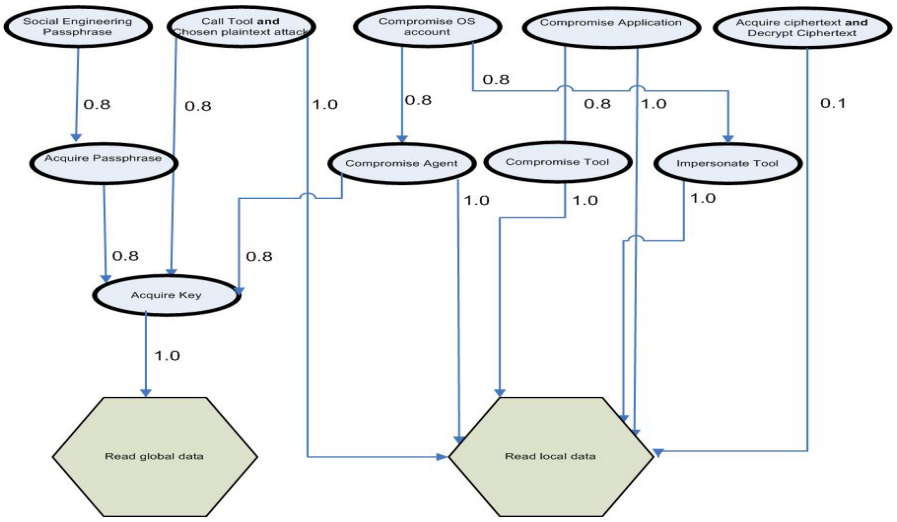

Fig 2. A FCM displaying the routes an attacker could take to compromise the system with weights on each route (based on the attack graph in Figure 1)

For example consider the following scenario. What happens if the event C1 (i.e. Social Engineering Passphrase occurs) occurs? This scenario can be presented using vector I_0 representing this situation by I_0 = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

In vector I_0 the concept C1 is represented as the first element in the vector and it is set to 1 and all other elements are set to be zero representing other events that has not happened. It is assumed that C1 happens and no other event has happened. Now I_0*E can provide the solution for this situation as follows: I*E =[0, 0, 0, 0, 0, 0, 0.8, 0, 0, 0, 0, 0, 0]= I_1

which conclude that if C1 happens then it will increase the possibility of C7 (i.e. Acquire Passphrase) to occur by 0.8. This process continues: I_1*E =[0, 0, 0, 0, 0, 0, 0,

0, 0, 0.8, 0, 0, 0]=I_2 which concludes that if C7 happens then it will increase the possibility of C11 (i.e Acquire Key) by 0.8. Now I_2*E = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1] = I_3 which conclude that if C10 happens then it will increase the possibility of C11 (i.e. Read global data) by 1 (or 100%). This means that the attacker will be able to read the global data.

$$E = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0 & 0.8 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

Fig3. Matrix representing value of connecting edges of FCM from Figure 2

Several simulations were performed using different scenarios generated by GA. The details are shown in Table 1. Table 1 displays the consequences of different scenarios. For GA mutation rate were set to 0.06 and crossover rate was set to 0.5.

What if the following event occurs	Consequences
C2	C2 $\xrightarrow{80\%}$ C11 $\xrightarrow{100\%}$ C12
C3	C3 $\xrightarrow{100\%}$ C13
C4	C4 $\xrightarrow{80\%}$ C8 $\xrightarrow{100\%}$ C13
C5	C5 $\xrightarrow{100\%}$ C13 Also C5 $\xrightarrow{80\%}$ C9 $\xrightarrow{100\%}$ C13
C6	C6 $\xrightarrow{100\%}$ C13
C7	C7 $\xrightarrow{80\%}$ C11 $\xrightarrow{100\%}$ C12
C8	C8 $\xrightarrow{80\%}$ C11 $\xrightarrow{100\%}$ C13
C9	C9 $\xrightarrow{100\%}$ C13
C10	C10 $\xrightarrow{100\%}$ C13
C11	C11 $\xrightarrow{100\%}$ C12

4 CONCLUSION

Attack graphs are designed to provide a graph of paths that attackers may take to reach their undesirable goals and to attack a system. Attacker's goals may include system's disruptions, intrusion and misuse. With complexity of existing systems

drawing attack graphs are becoming increasingly difficult and as such an attack graphs may be flawed. Attack graphs do not provide any facilities to analyze and assess different risks and possible attacks that may exist in attack graphs in a systematic way. Fuzzy Cognitive Maps (FCM) is employed in this paper to provide the facilities to capture and represent complex relationships in a system and to improve the understanding of a system designer to analyze risks. Using a FCM different scenarios are considered. The proposed FCM is used in conjunction with attack graph to provide system architects with possibilities for what-if analysis to understand the vulnerability of their designed system. What-if scenarios were generated using a GA.

From simulation results it was found that the FCM is capable of making accurate predictions attack for the given system. The GA provided many different attack scenarios for the FCM. Using such Scenarios an expert can inspect and make any modifications if necessary to the given IT system based on analysis of each attack scenario. The research work performed in this paper is unique in the way the attack graphs, FCM and GA are integrated. The application of this method to several other IT network security analyses is currently under consideration.

REFERENCES

1. Gupta, 1. S. and Winstead, J.: Using Attack Graphs to Design Systems, IEEE Security and Privacy, IEEE Computer Society Publishing (2007)

2. Peterson, G. and Steven, J.: Defining Misuse within the Development Process, IEEE Security & Privacy, vol. 4, no. 6, pp. 81–84 (2006)

3. Peeters J., Dyson. P.: Cost- Effective Security, IEEE Security & Privacy, vol. 5, no. 3, pp. 85–87 (2007)

4. Kosko. B.: Fuzzy Engineering, Prentice Hall, Upper Saddle River, USA (1997)

5. Kosko. B.: Fuzzy Cognitive Maps, International Journal of Man-Machine Studies, Vol. 24, pp. 65–75 (1986)

6. Aguilar. J.: A Survey about Fuzzy Cognitive Maps Papers" International Journal of Computational Cognition, voI 3, no. 2, ρρ. 27-33 (2005)

7. Goldberg, D.. Genetic Algorithms in Search, Optimisation and Machine Learning. Reading, Massachusetts: Addison Wesley, USA (1989)

8. Swiler, L. P., Phillips, C. Ellis, D., Chakerian. S.: Computer-attack graph generation tool. In DISCEX II'01: DARPA Information Survivability Conference and Exposition Conference and Exposition.Vol 2, pp 307-321 (2001)