Combining defense graphs and enterprise architecture models for security analysis

Teodor Sommestad, Mathias Ekstedt, Pontus Johnson
Department of Industrial Information and Control Systems
Royal Institute of Technology (KTH)
{teodors, mathiase, pontusj}@ics.kth.se

Abstract

Security is dependent on a mixture of interrelated concepts such as technical countermeasures, organizational policies, security procedures, and more. To facilitate rational decision making, these concepts need to be combined into an overall judgment on the current security posture, as well as potential future ones. Decision makers are, however, faced with uncertainty regarding both what countermeasures that is in place, and how well different countermeasures contribute to mitigating attacks. This paper presents a security assessment framework using the Bayesian statistics-based Extended Influence Diagrams to combine attack graphs with countermeasures into defense graphs. The approach makes it possible to calculate the probability that attacks succeed based on an enterprise architecture model. The framework also takes uncertainties of the security assessment into consideration. Moreover, using the extended influence diagram formalism the expected loss from each attack can be calculated.

1. Introduction

To efficiently protect systems against attacks decision makers need to be able to assess the current security posture of the enterprise's systems as well as the security posture after potential improvements. For security investments the expected consequence of attacks prior and after a security investment is a metric that enables rational decision making [20]. While the targets of attackers and the number of attacks attempted is typically difficult for decision makers to influence, the probability of attacks succeeding can to great extent be controlled through the deployment of countermeasures and safeguards.

Decision makers are however faced with two types of uncertainty when assessing the protection against attacks. Firstly, there is an uncertainty in how the security mechanisms influence each other and how they contribute to enterprise-wide security. As stated in [21], there is today no algebra on perimeter security. Secondly, there is an uncertainty as to whether information and indicators collected during a security assessment is credible and as a consequence of this how credible the assessment result is [22].

This paper presents a framework for quantitative assessment of system security that takes both these types of uncertainty in consideration. The framework utilizes methods developed for analysis of enterprise architectures and combines these with the concept of defense graphs to infer the security posture of system. The security posture is assessed in terms of expected success rate of an attack and the expected loss of an attack.

The structure of this paper is as follows. In section two and three related work is described. Section four describes how extended influence diagram can be used for assessing security by enhancing attack graphs with countermeasures. An example of the framework covering password protection of a computer application is here presented. Section five presents the metamodel related to the extended influence diagram that has been presented in section four. Subsequently section six presents an example of how this metamodel can be instantiated into a concrete model and how it facilitates security assessment. Finally, in section seven, conclusions are drawn.

2. Attack graphs and defense graphs

Attack trees are graphical notations evolved from fault trees and illustrates attackers' goals together with possible ways to reach these goals [14][15][16]. The attacker's main goal is depicted as the root of the tree and the steps to reach this goal are broken down into sub-goals of the attack through "AND" and "OR" relationships. The "AND" relationship requires the attacker to accomplish all underlying sub-goals to achieve the goal, whereas "OR" relationship only

IEEE
computer
society

requires one of the underlying sub-goals to be accomplished to achieve the goal.

These tree and graph structures have been applied in several ways to assess security of systems and to assess system vulnerabilities and risks. Both [16] and [13] has proposed the use of attack trees during system development to analyze the security. In addition to this, plenty of analysis techniques based on attack trees has been suggested, see for example [17] [24][25]. Also, model checking techniques have been developed to generate attack graphs by using scanner tools [18][19] and these have been augmented with some analysis functionality.

Attack graphs can easily become extensive and to more compactly represent them Liu and Hong [1] have used Bayesian networks to express them and to calculate the probability of an attack against computer networks being successful based on vulnerabilities within it. These "Bayesian attack graphs" can be used to answer questions about the current security posture and facilitate comparison to previous measurements, but does not include controllable attributes and does not answer questions about how to improve the security posture.

While it is typically difficult to directly control what actions an attacker will chose and how frequent their attempts are decision makers can to great extent control the difficulty to perform undesired actions through countermeasures. Hence, a natural extension of attack graphs is to include these controllable countermeasures in the graph. In [13] countermeasures are modeled together with trees depicting threats, and in the theses by Foster [16] and Schechter [14] countermeasures are included in the tree structures. The concept of including countermeasures in the tree structure has also been used in [2], to create something called "Defense trees".

Techniques has been presented which use defense trees for strategic evaluation of security investments [2], modeling strategic games in security [23] as well as modeling of conditional preference of defense techniques using conditional preference nets [3].

Security assessments involve uncertainty regarding casual relationships between security related variables and uncertainty regarding the accuracy of collected data. Bayesian statistics is a formalism well equipped for combining disparate concepts and managing the uncertainty present in security assessments. However, no prior work has been found on using attack trees, nor similar structures, enhanced with countermeasures together with Bayesian statistics to assess security of systems.

3. Extended Influence Diagrams metamodels and abstract models

This section describes the formalism Extended Influence Diagrams, metamodels associated to these and abstract models which combines these two.

3.1 Extended Influence Diagrams

Extended Influence Diagrams are graphic representations of decision problems coupled with a probabilistic inference engine. These diagrams may be used to formally specify enterprise architecture analysis [4]. The diagrams are an extension of influence diagrams, [5][6] which in turn are an enhancement of Bayesian networks [7][8]. In Extended Influence Diagrams, random variables associated with chance nodes may assume values, or states, from a finite domain such as {High, Medium, Low} or {True, False}. A variable could for example be "encryption strength" or "use of digital signatures". These variables are connected with each other through causal or definitional arcs. Causal arcs capture relations of the real world, such as "stronger encryption yield higher confidentiality". Definitional relationships are on the other hand defined by the modeler, who also specifies how the defined property is defined by its parents [4]. The security concept, which could be regarded as abstract, can for example be defined through preservation of confidentiality, integrity and availability. Extended Influence Diagrams support probabilistic inference in the same manner as Bayesian networks do; given the value of one node, the values of related nodes can be inferred.
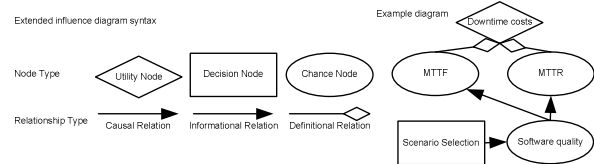
Figure 1 -The notation of Extended Influence Diagrams.

Extended Influence Diagrams further includes decision nodes and utility nodes. Decision nodes may as chance nodes assume one of several predefined and mutually exclusive states and can be coupled with chance nodes to express the capability to influence chance nodes' state. Utility nodes are used to express the utility associated with each combination of states in chance and decision nodes. A value expressing utility, positive or negative, is assigned to each states of influencing chance nodes and decision nodes. For more comprehensive treatments on Bayesian networks,

350

influence diagrams and Extended Influence Diagrams see [4][5][6][7][8][9] and [10].

3.2 Metamodels for Extended Influence Diagrams

Extended Influence Diagrams usually include a number of attributes influencing the assessed property. These are typically associated with physical artifacts of the real world, such as "computer" or "person", or sometimes with more abstract concepts such as "process" or "dataflow". As a support for assessing these, a metamodel can be used. This metamodel expresses these concepts together with the attributes they hold. Furthermore, since relationships between attributes of entities are caused by real relationships between the entities they are held by, entity relationships are of relevance to the assessment and thus also included in the metamodel. Based on the nature of the entity relationship, multiplicities are associated to it. Hence, the metamodel contains: entities, entity relationships, and attributes held by the entities.

4. Using Extended Influence Diagrams for attack and defense graph modeling

To illustrate how attack graphs, consequences and countermeasures can be jointly modeled using Extended Influence Diagrams, a model of access control on a standalone computer is here presented. We limit the example to one type of adversary and logical access control, where logical access control can be password based.

4.1 Expressing defense trees through Extended Influence Diagrams

The probability that an attacker succeeds when attempting to gain access to a system depends on its architecture and design. As pointed out by [14] the structure of attack trees depends on the countermeasures that are deployed since countermeasures introduce new ways of attacking a system. For example, if access control is applied for a system the adversary will have to find ways to bypass it. If further controls, possibly of a different type, also is applied, the adversary has to overcome even more hinders.

For password protection three general strategies are here assumed to exist. In the first two of these the attacker performs a brute force attack or carries out a dictionary attack. With the third strategy the attacker finds out the password by other means, for example by social engineering. An attack tree depicting these goals is given in Figure 2.
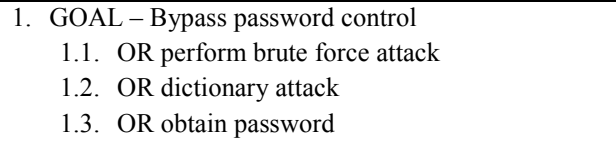
1. GOAL – Bypass password control
   1.1. OR perform brute force attack
   1.2. OR dictionary attack
   1.3. OR obtain password

Figure 2 - Attack tree for password control.

For a computer protected with password on operating system level as well as on application level the attack tree would be as depicted in comprise of two similar sets of attack trees, one for each protection mechanism..

Extended Influence Diagrams can be used to express attack graphs. The steps in an attack can be illustrated by chance nodes with the states "successful" and "failure". The AND-relationships and OR-relationships in the attack graph can be expressed using definitional relations and specified through conditional probability tables. Moreover, the consequences of a successful attack can be taken into consideration and expressed through utility nodes (cf. Figure 3).
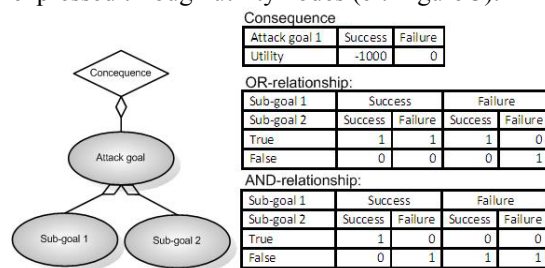

Figure 3 - Attack trees expressed through Extended Influence Diagrams

The probability of an attacker succeeding in bypassing a leaf of the attack tree is dependent on the capability of the attacker and the countermeasures that are in place. To assess the probability of an attacker succeeds in achieving its goal(s); the state of countermeasures needs to be taken into consideration. Therefore the attack tree in Figure 1 is enhanced with the countermeasures that influence the probability of an attacker succeeding with his sub-goals and his ultimate goal (cf. Figure 4). As noted above, some of the countermeasures are an integral part of the attack graph and will introduce attack goals to the attack graph based on the hinder(s) they form.
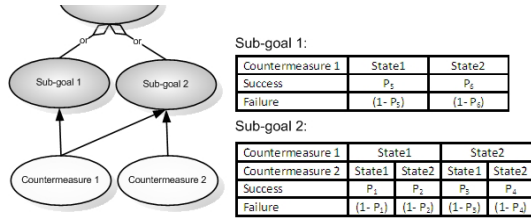
Figure 4 – The influence of countermeasures on the difficulty in succeeding attacks and can be expressed through Extended Influence Diagrams

Sub-goal 1:

Countermeasure 1	State1	State2
Success	P_5	P_6
Failure	$(1-P_5)$	$(1-P_6)$

Sub-goal 2:

Countermeasure 1	State1		State2	
Countermeasure 2	State1	State2	State1	State2
Success	P_1	P_2	P_3	P_4
Failure	$(1-P_1)$	$(1-P_2)$	$(1-P_3)$	$(1-P_4)$

4.2 A defense tree example

In this example it is assumed that the difficulty of obtaining passwords is directly dependent on three attributes. The existence of default passwords that grants access is one factor; the use of passwords in other systems is another factor that makes it easier to find them out. Also important is if password holders are susceptible to social engineering or not. Factors that affect the difficulty of cracking passwords in a brute force attack are the strength of passwords and if there is a limitation to the number of attempts that an attacker can try passwords using standard logon functionality. The strength of passwords are also believed to influence the difficulty of performing brute force attacks together with the existence of password hashing, the size of salt added to the passwords. The same attributes are of relevance to the difficulty of performing dictionary attacks, but presumably in another way.

The efficiency, functioning and strength of technical countermeasures are in many cases dependent on the quality of processes and humans surrounding them. For instance, passwords do not offer strong protection if they are not kept confidential, are default passwords, or if they are weak. The presence of weak passwords could in turn be a reflection of how whether the employees have received security training and if password policies hold desirable qualities. Hence, the difficulty of succeeding in an attack may also be affected by the processes carried out within the organization. For instance, a security awareness and training program can be introduced to increase employees' knowledge of how passwords should be constructed.

Figure 5 depicts the structure of an extended influence diagram expressing the abovementioned. Provided evidence on the states of the influencing attributes and conditional probabilities of their relationships a probability of success can be inferred using Bayesian statistics
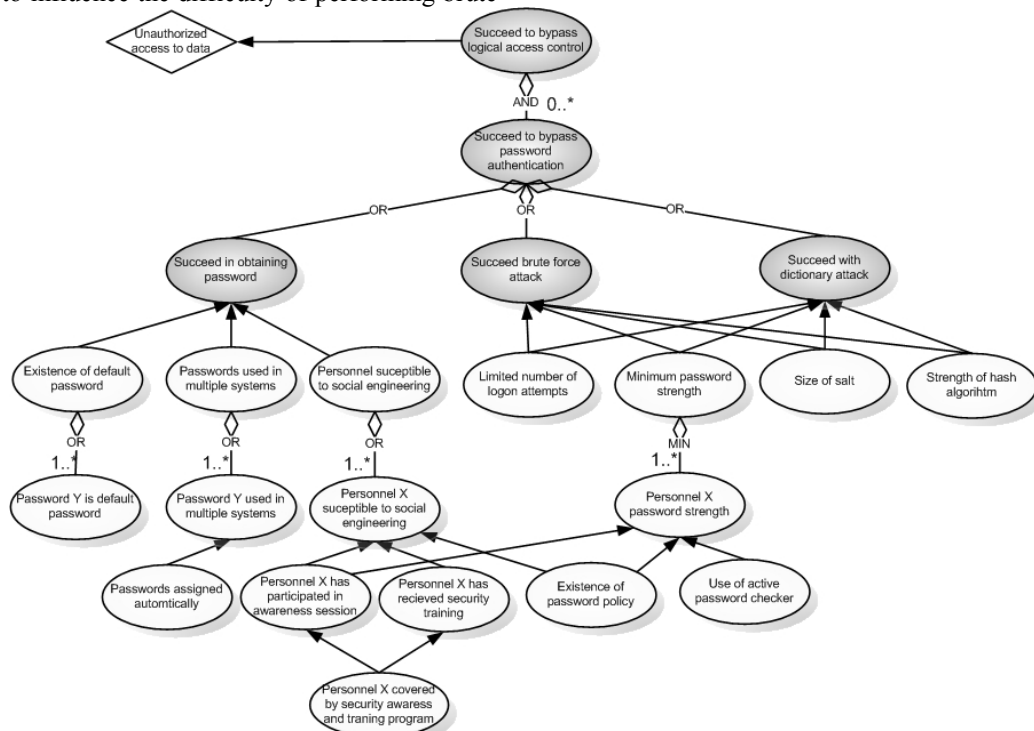

Figure 5 – Defense graph for access control expressed through an extended influence diagram. Grey nodes represent the attack graph and white nodes represent countermeasures.

352

5. A metamodel for assessment support

The attributes of the extended influence diagram presented in Figure 6 is related to concepts that are relevant for logical access control security. In this section a metamodel capturing these concepts are presented.

5.1 Entities and attributes in the metamodel

The attributes of the extended influence diagram refers several concepts that needs to be investigated in a security assessment. The objective is to protect some *data* and vital for this is the *password authentication mechanisms* which protects software such as *application* and *operating systems*. The password authentication mechanism uses passwords to grant or deny access and these *passwords* could or should be governed by a *password policy*. The *persons* who own the passwords have an influence on security related attributes according to the extended influence diagram and should also be considered in the assessment. Furthermore, if password holders are covered security *training and awareness program* is influencing the probability that these individuals have participated in training or awareness sessions. Hence, this aspect should also be included.

The entities are included because they hold attributes that are of relevance to the assessment. The entity *password* is relevant because they should be *strong*, a property that is believed to be influenced by whether they are governed by a *password policy* or not. The persons holding the passwords are relevant since
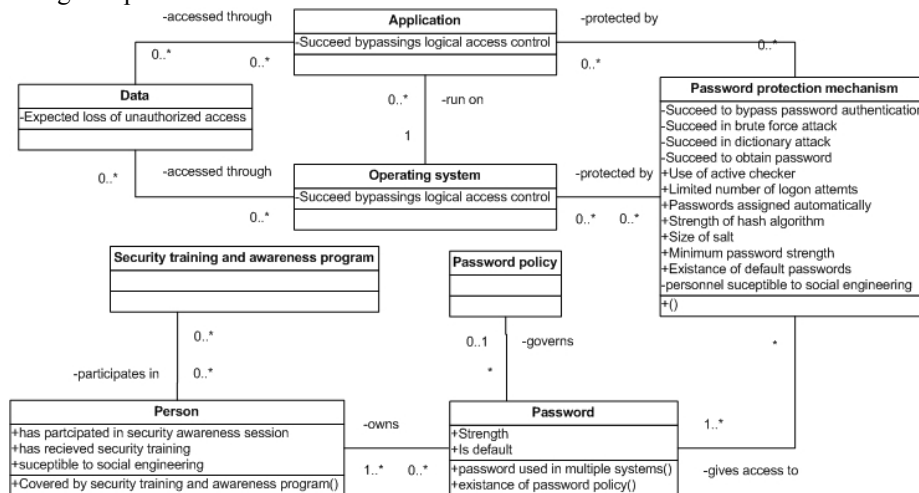
their participation in awareness sessions and security training is influencing other attributes. The password authentication itself is holding attributes such as hash algorithms and if there is an active password checker in use.

Some of the attributes (e.g. *existence of password policy*) can be assessed from an instantiated model through model checking. These attributes have been included as functions in the UML notation of Figure 6. Other attributes need to be assessed directly based on evidence of their state. For instance will evidence on the *use of active password checkers* be information that influences the belief in the states "true" and "false" of this attribute.

5.2 Attribute relationships and entity relationships

Relationships among attributes in the extended influence diagram arise from a relationship among the entities they are held by. For instance, in an instantiated metamodel the entity relationship *owns* between a person and a password would imply that the person's attributes influence strength of the passwords. If a password's strength is influencing the minimum password strength of an authentication mechanism depends on whether it *gives access to* that particular password authentication mechanism. In the same way, a password protection mechanism's attributes will only influence the difficulty of bypassing logical access control of software if it protects that specific software. Hence, entity relationships that causes attribute relationships are of relevance to the assessment and included into the metamodel.

Figure 6 – Metamodel coupled to the extended influence diagram in Figure 5. The union of these two makes out the abstract model. Methods in this diagram represent attributes that can be assessed through model-checking.
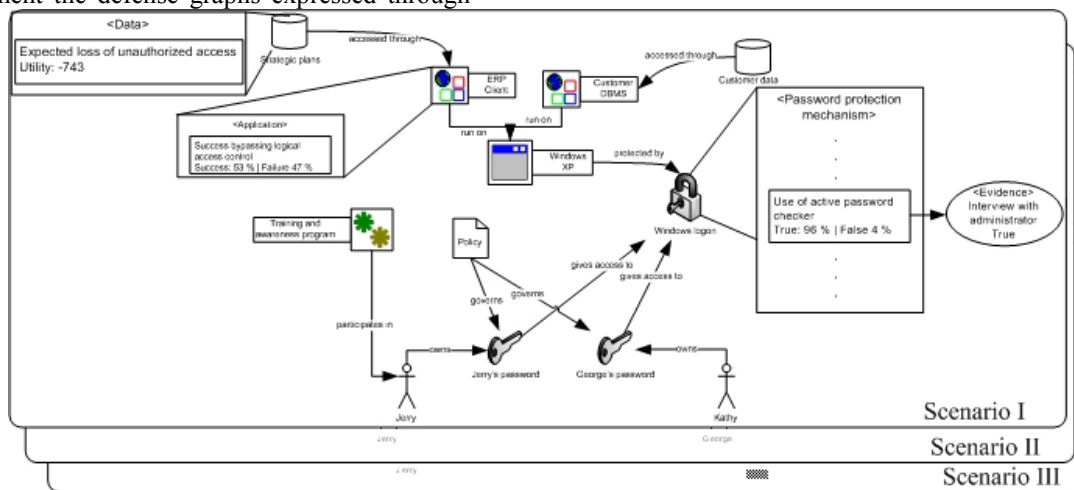
353

6. A concrete model

A security assessment typically involves data collection in terms of interviews, documentation studies, log reviews, penetration tests and more. The purpose of this is to collect information (evidence) about matters that are believed to influence security to facilitate analysis. One part of this information collection serves to identify the entities that need to be investigated and their relationship to each other. Another part of the data collection concerns the quality of security influencing attributes and analyzing how these qualities influence security.

In Figure 7 evidence obtained from an interview on the use of automatic password checker is depicted as an ellipse. Table 1 expresses the significance of this of evidence by describing the expected outcome of the interview based on the possible states of assessed attribute. In this example, the system administrator would give the answer "true" with 95 percents probability if there is an automatic password checker. With 10 percents probability the system administrator would wrongfully answer "true" even if there was no automatic password checker.

Table 1 - A conditional probability table specifying credibility of evidence on the use of automatic password checker.

Automatic password checker	T	F
True	0.95	0.10
False	0.05	0.90

Based on the evidence captured in the assessment the defense graphs expressed through

Extended Influence Diagrams can be used to answer several questions:

- The probability of an attack succeeding can be inferred based on information about existing countermeasures.
- The consequence of potential attacks adjusted for their success rate.
- An index of the security as the quota between lowest possible success rate adjusted consequence (optimal solution) and the existing one.

Moreover, by elaborating with what-if scenarios decision makers can assess the security provided by different scenarios. This can be done by linking decision nodes to controllable attributes and specify the impact of decisions on the state of the affected attributes.

7. Conclusions

Information on the expected loss prior and after a potential security investment enables rational decision making. Using the framework presented herein, decision makers can create models of both current and potential future scenarios based on metamodels covering concepts relevant to security. From attacks plausible in these scenarios and the countermeasures they include, an assessment of security can be derived for each scenario.

The use of Bayesian statistics enables uncertainty to be considered in both the modeling of a security related attributes and in the influence of these attributes on the possibility for an attacker to compromise the system. Moreover, the

Figure 7 - A concrete model of the assessed enterprise. Entities and evidence on their attributes enable the probability of attacks success to be inferred and the expected loss to be calculated.

354

formalism Extended Influence Diagrams enable consequences of attacks to be specified and taken into consideration. Together with data on the frequency of attacks, this facilitates expected loss to be calculated for both the current enterprise architecture, and potential future scenarios.

8. References

[1] Y. Liu and M. Hong, Network vulnerability assessment using Bayesian networks, Proceedings of Data Mining, Intrusion detection, Information assurance and Data networks security, Orlando, Florida, USA, 2005, pp

[2] S. Bistarelli, F. Fioravanti, P. Peretti, Defense trees for economic evaluation of security investments, Proceedings of Availability, Reliability and Security (ARES), Vienna, Austria, 2006, pp. 8.

[3] S. Bistarelli, F. Fioravanti, P. Peretti, Using CP-nets as a Guide for Countermeasure Selection, Proceedings of the 2007 ACM symposium on Applied computing, Seoul, Korea, 2007.

[4] P. Johnson, R. Lagerstrom, P. Narman, M. Simonsson., Enterprise Architecture Analysis with Extended Influence Diagrams. Information System Frontiers, 9(2), Springer Netherlands, 2007, pp. 163-180.

[5] R. Shachter, Evaluating influence diagrams. Operations Research, 34(6), Institute for Operations Research and the Management Sciences, Hanover Maryland, 1986, pp. 871-882.

[6] R.A Howard, J.E. Matheson, Influence Diagrams. Decision Analysis, 2(3), Institute for Operations Research and the Management Sciences, Hanover Maryland, 2005, pp. 127–143.

[7] Neapolitan, R. Learning Bayesian Networks. Prentice-Hall, Inc. Upper Saddle River, NJ, USA 2003.

[8] Jensen, F.V., Bayesian Networks and Decision Graphs, Springer New York, Secaucus, NJ, USA 2001.

[9] P. Johnson, Lagerström, R., Närman, P.: Extended Influence Diagram Generation. Enterprise Interoperability II – New Challenges and Approaches, Springer London, 2007, pp. 599-602.

[10] R. Shachter, Probabilistic inference and influence diagrams. Operations Research, 36(4), Hanover Maryland, 1988, pp. 36-40.

[11] M.J. Druzdzel and L.C. van der Gaag, Elicitation of Probabilities for Belief Networks: Combining Qualitative and Quantitative Information, Proceeding of the 11th Conference on Uncertainty in Artificial Intelligence, 1995, pp. 141-148.

[12] M.J. Druzdzel and L.C. van der Gaag, Building probabilistic networks: where do the numbers come from?, IEEE Transactions on Knowledge Data Engineering, 12(4), 2000, pp. 481–6.

[13] M. Howard and D. C. LeBlanc. Writing Secure Code, Microsoft Press, Redmond, WA, USA, 2002.

[14] S. E. Schechter. Computer Security Strength & Risk: A Quantitative Approach. PhD thesis, Harvard University, 2004.

[15] B. Schneier. Attack trees: Modeling security threats. Dr. Dobb's Journal, 1999.

[16] N. L. Foster. The application of software and safety engineering techniques to security protocol development. PhD thesis, Univ. of York, Dep. Of Computer Science, 2002.

[17] S. Jha, O. Sheyner, and J. Wing. Two formal analyses of attack graphs. In Proc. of the 15th Computer Security Foundation Workshop, June 2002.

[18] O. Sheyner "Scenario Graphs and Attack Graphs," Carnegie Mellon University, April 2004. PhD Thesis.

[19] O. Sheyner and J.M. Wing, "Tools for Generating and Analyzing Attack Graphs," Proceedings of Workshop on Formal Methods for Components and Objects, 2004, pp. 344-371.

[20] J.J.C.H. Ryan and D.J. Ryan, Expected benefits of information security investments, Computers & Security, 25(8), 2006, pp 579-588.

[21] R. Vaughn, R. Henning, and A. Siraj, Information assurance Measures and Metrics: State of Practice and Proposed Taxonomy. Proceedings of 36th Hawaiian International Conference on System Sciences, 2003, pp. 331-334.

[22] E. Johansson, Assessment of Enterprise Information Security - How to make it Credible and Efficient, PhD Thesis, Royal Institute of Technology (KTH), 2005.

[23] S. Bistarelli, M. Dall'Aglio, P. Peretti, "Strategic games on defense trees", Formal Aspects in Security and Trust, Springer Berlin / Heidelberg, 2007, pp. 1-15.

[24] P. Ammann, D. Wijesekera, and S. Kaushik. Scalable graph-based vulnerability analysis. In Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS '02), pages 217–224, Washington, DC, November 2002.

[25] Philips, C., Swiler, L.P, Graph-Based System for Network-Vulnerability Analysis, Proceedings of the 1998 workshop on New security paradigms, 1998.