A Threat Analysis Methodology for Security Evaluation and Enhancement Planning

Stango, Antonietta; Prasad, Neeli R.; Kyriazanos, Dimitris M.

Published in:
Third International Conference on Emerging Security Information, Systems and Technologies, 2009. SECURWARE '09

DOI (link to publication from Publisher):
10.1109/SECURWARE.2009.47

Publication date:
2009

Document Version
Publisher's PDF, also known as Version of record

A Threat Analysis Methodology for Security Evaluation and Enhancement Planning

Antonietta Stango, Neeli R. Prasad
Aalborg University, CTiF
Aalborg, Denmark
e-mail: {as, np}@es.aau.dk,

Dimitris M. Kyriazanos
Telecommunications Laboratory, School of Electrical and Computer Engineering
National Technical University of Athens, Greece
e-mail: dkyri@telecom.ntua.gr

***Abstract*—Threat analysis gives how potential adversaries exploit system weakness to achieve their goals. It identifies threats and defines a risk mitigation policy for a specific architecture, functionality and configuration. In a threat analysis security metrics are a challenging requirement in order to determine the status of network security performance and to further enhance it by minimizing exposure to considerable threats and vulnerabilities. In this paper the authors propose a generic methodology for threat analysis and security metrics in order to prioritize threats and vulnerabilities and proceed with security enhancement planning in Personal Networks (PNs).**

Keywords: threat analysis, vulnerabilities, assets, security metrics.

I. INTRODUCTION

In the more recent years the huge diffusion of new technologies and internet increases the need of security, because communication networks are used to transfer increasingly sensitive information that can be valuable and confidential, requiring protection against human misuse and also attracting attention of malicious people.

Network security is the process by which digital information assets are protected, where the word security means protection against attacks by malicious outsiders or insiders. All networks to achieve its fullest potential need to be protected from threats and vulnerabilities. The process to identify the threats, which consists in identifying how potential adversaries exploit system weaknesses to achieve their goals [1], and find appropriate countermeasures, is the threat analysis.This process is necessary for specifying a solid and complete set of security requirements so as to build all needed security mechanisms efficiently protecting the system. Moreover, when conducted on an existing system, a correct evaluation of the threats and vulnerabilities allows to prioritize them, assess the security of the system and propose an optimal enhancement plan.

A. Related work

The research in threat analysis has to mature as there are only a few established techniques to aid a formal threat analysis procedure and most of them are related only to software security. Moreover existing work do not integrate threat modeling - that is the process of identifying and documenting threats in a system - with a formal threat analysis. Swiderski and Snyder [2], for example, describe widely the threat model, but there is no method for determining the assets value and no risk analysis has been conducted, furthermore it targets only software applications. For explicitly modeling and analyzing security threats during requirements analysis a goal oriented approach has been proposed [3], related to software applications. Threat modeling [4] is also used as a step toward addressing the completeness of the security requirements, and the process is also extended to suit complex, networked systems. The characterization of the system can be done with Data Flow Diagrams, by a Network Model [4] or also with an high level architecture diagram [5] depending on the system, software application or networked system. The authors of [4] envisage the risk management and also the mitigation of threats. In software architectures the threats have been modeled also with misuse case [6]. UML sequence diagrams are exploited to describe the decision process of an attacker who would go through to compromise or misuse the system, and also to evaluate the architecture and the constraints that can be imposed to mitigate the threats. A framework for threat model for Personal Network is presented in [1], the methodology is general, but not complete, and the mitigation of threats is not included. Another practical solution is made by PTA Technologies [7], it consists of a software tool to assist software developers in assessing system risks and building a risk reduction policy for their systems. This tool is able to conduct a complete threat analysis, nevertheless the vulnerabilities are not ranked and the final values given to the threats correspond to financial fund losses, - a debatable methodology. CVSS [8] provides a vulnerability scoring system for rating IT vulnerabilities, but not for threats and it does not provide a methodology for threat analysis.

The aim of this work is to improve the existing techniques by integrating threat modeling with a formal threat analysis and proposing a generic methodology valid for both networked systems and applications, with a new proposed approach in the characterization of the system.

The authors propose the use of the UML use case diagrams along with the UML sequence diagrams to give an overview of the system and to analyze the technical background of the use cases. The use cases diagrams allow to describe what the system is able to do, covering all functionalities found in the scenarios, from the point of view of the user, whereas the sequence diagrams extend this view by including all entities involved in the system. A preliminary version of this methodology has been applied for a Federation of Personal Network (PN-F) [9]. A personal network is a person-centric network that provides access to

personal resources, services, and contents. The interaction between PNs raises the federation of personal networks.

The application of the proposed methodology to PN-Fs highlights that it is very difficult to prioritize threats and vulnerabilities with quantitative evidence because of lack of effective metrics, and the complex and sensitive nature of security [10]. In order to solve the last issue the authors further propose a combined methodology to rank threats and vulnerabilities combining attack trees and the CVSS scoring system.

In the proposed approach the criticism attack trees receive for (i) not being able to model cycles and (ii) being too complex and unmanageable for complicated systems and attack scenarios are considered.

Despite these drawbacks, attack trees have been used in and proposed for threats identification throughout the literature [11][12], since they offer a clear representation of (inter)dependencies of states reached - a common drawback in other methods. In order to tackle with attack tree modeling problems mentioned, the sequence of operations should not be within the scope of the modeling, but instead the final state reached. Moreover, tree-pruning and simplifying complexity techniques should be applied as in [11].

In this context, our proposal applies a user-centric, use case-based approach to reduce complexity, focusing on the user actions (legitimate or malicious) rather than the network and system components. Moreover, the trees are reserved to show state dependencies while a vulnerability level is assigned to each state. Within this scope, CVSS for the qualitative assessment of all tree nodes - states rather than just use CVSS as a tool for primary metrics in a more complex algorithm [12] is used. In addition, CVSS keeps the proposed tree more realistic and updated in terms of attacker effort and expertise needed in comparison with tree-geometry based algorithms [11]. The goal is to provide a KISS principle compliant (Keep it Short and Simple), stand-alone complementary security management tool useful for an experienced system administrator, a security analyst and the non-expert user of a PN.

The remainder of this paper is organized as follows: in section 2 a description of the threat analysis methodology is given, followed by security metrics for ranking threats and vulnerabilities, as presented in Section 3. Section 4 summarizes the conclusions.

II. PROPOSED THREAT ANALYSIS METHODOLOGY

The threat analysis is a formal process of identifying, documenting and mitigating the security threats of a system, which can be divided in three main phases: threats modeling, assets mapping and building a mitigation plan. The proposed methodology includes the formalization of all these aspects with a new approach in the characterization of the system.

Threat modeling is a method of assessing and documenting the security risk associated with an application that involves also understanding the goals of an adversary in attacking a system based on the assets of interest. This allows to enumerate the threats and also to discover the vulnerabilities. The threat modeling is very useful especially if is done in the earliest stage of the system development and then, as the applications evolve and requirements are better defined, the lists of threats and vulnerabilities can be updated as needed.

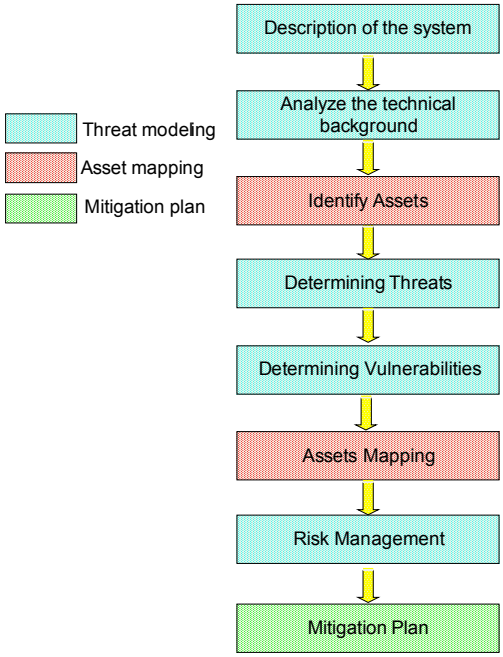

Figure 1. Steps of Threat analysis

Asset mapping involves documenting the tangible and intangible resources of the system and identify the related entry points of the system. The assets value is used as basis for calculating threat risks and for prioritizing countermeasures, ergo assets need to be prioritized. It is often easier for the analyst to identify system assets via the process of analyzing specific threats. This implies an iterative approach of mapping assets and enumerating threats.

The third phase of the threat analysis is building a mitigation plan, namely selecting from the list of all the proposed countermeasures, the most effective combination. The analysts will decide which of the proposed countermeasures will be included in the actual mitigation plan according to their experience. In order to assist with the computation of countermeasures' contribution to mitigating the threat's risk, the analyst is asked to a) estimate the mitigation level each countermeasure provides to the threat as if it is the only countermeasure in the mitigation plan and b) estimate the total level of mitigation provided to the threat's risk by all countermeasures in the mitigation plan [7]. The result of this analysis is a set of countermeasures that mitigate the threats identified. In the Figure 1 the proposed methodology specially adopted for PNs is shown step by step.

A. Step 1. Description of the system: network overview and use cases

To describe a system it is needed to understand every component and its interconnections, defining the scenarios and the use cases.

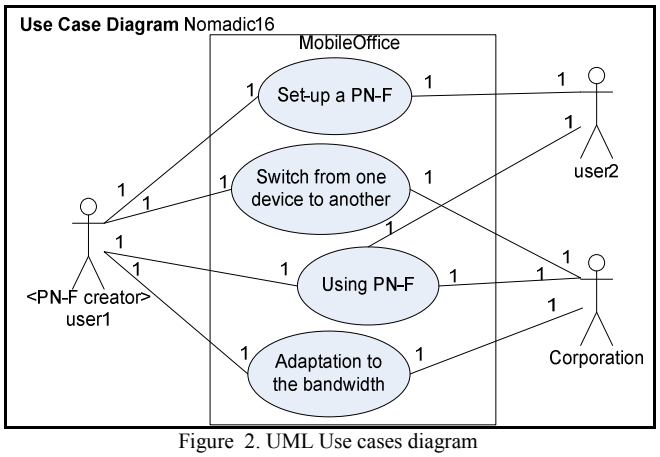

Figure 2. UML Use cases diagram

TABLE I. DESCRIPTIVE TABLE OF UML USE CASE DIAGRAM

Use case name	**Set-up a PN-F**
Goal In Context	Collaborative work, Service discovery
Preconditions	The actors are carrying portable devices, which incorporate MAGNET Air Interface/WiFi/UMTS capabilities as well as general office accessories. The devices are interconnected with each other.
Successful End	A PN-F is created
Failed Condition	No federation
Primary Actors	Two colleagues (PN-F creator, user)
Secondary Actor	
Trigger	The creator ask for a federation
Main flow	
1	User1 (the creator) ask for the PN-federation by selecting the PN-federation definition in the user GUI
2	The user GUI asks for the identifiers of the Personal Networks allowed to participate in the PN-federation
3	User1 gives the PN identifiers of both colleagues.
4	User1 opens the PN manager, which recognizes the colleague's device as a foreign node.
5	By using the PN manager the creator selects the device of User2 and sends an invitation to it for the PN-federation with the PN manager command: Invite to PN-Federation.
6	User2 opens the PN manager and accepts the received invitation.
7	When the PN manager of User2 has shown a message about a secure PN-federation connection between the personal networks, User2 opens the PN directory manager and selects devices and the directories in the devices, which will be included in the PN-federation. These devices are the PDA and the laptop.
8	User2 sends this information to the creator by the PN manager command: PN-federation participation devices.
9	The creator selects the PDA and laptop by using the PN directory manager. He sends this information to user2 by the PN manager command: PN-federation participation devices.
Extensions	
7.1	No secure connection for PN-F federation.
7.2	No federation.

To help the system characterization the UML use cases diagram and the related descriptive tables are proposed, as they allow to describe what the system must be able to do, by showing the interaction between the use cases and the actors involved. The use case description is in a table that contains all the information related to the system and the use case, i.e. the goal, the devices and involved technologies, the description of the actor and the stages to realize the use case. In Figure 2 an example of the UML use cases diagram for the scenario "Nomadic@work" [9] and in Tab.1 the descriptive table related to the use case "Set-up a PN-F".

B. Step 2. Analyze the technical background of the use cases

To analyze the timing sequence of all the devices and actors involved in the use case, the UML sequence diagrams are suitable, as suggested in [1]. A sequence diagram shows object interactions organized according to their timing sequence. The sequence view describes the system in execution. It can be used to model the behavior of the system by representing the realization of a use case scenario. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects.

In this step it is possible to analyze how the technologies are used and who is using them.

.

C. Step 3. Identify Assets

In this step everything that can be damaged or violated in the network should be determined. Assets can be tangible or abstract, general or related to a use case. The assets and their protection are very important, as organizations often tend to focus on threats rather than on protecting assets, which leaves the entire system vulnerable [13].

In general the assets depend on the situations and on the users, but it is possible also to identify some general assets for the system, e.g. the IDs of the owner or the devices itself. The new approach of using use case diagrams and the descriptive tables allows to identify the general and specific assets of each use cases by analyzing them. In this step the assets are enumerated and stored in a table as a record with ID, name and a brief description. In the next steps, this table can be checked and updated in case some new assets are found.

D. Step 4. Determining Threats

Using the information gathered so far it is possible to start to identify the threats and the potential threat-sources of the system. A threat-source is defined as any circumstance or event with the potential to cause harm to a system. According to [14] the threat sources are classified in: natural, human, and environmental.

By analyzing the use cases, technical functionalities and sequence diagrams, it is possible to identify the threats and threat-sources. Afterwards the threats have to be correlated with the assets and with the entry points. The output of this step is the threats profile, similar the solution that Swiderski and Snyder suggest in [2]. In the table every threat is

associated with an ID, a name or classification, the source of threat, the assets involved and the entry points associated, an example can be found in [9]. Finally the threats must also be analyzed to determine whether the system is susceptible to them.

E. Step 5. Determining Vulnerabilities

The goal of this step is to develop a list of system vulnerabilities that could be exploited by the potential threat-sources. When all the threats and their scenarios have been described it is possible to deduct what the threats are exploiting. A table will be filled in with the main real vulnerabilities and the corresponding threats that are exploiting in the specific use cases analyzed in the previous sections. Each vulnerability will be assigned an ID, followed by the description, the name and the corresponding threat. Another way to evaluate if the system is susceptible to the threats that have been identified is to use the attack trees. This approach has been adopted in section 3.

F. Step 6. Asset Mapping

In this step the list of assets determined in the step 3 is checked to determine if all the assets have been included. It is important also determining the valiance of the assets and the risk that the owner of the assets is willing to accept [1], and based on these values prioritizing them. To assign a value to the assets is not easy because the value can be personal and the priorities of the people can be different, nevertheless here it has been suggested three different values:

- *High*, assets with this value have to be protected with a high level of security; they are directly linked to the control of the system, with services that require highly secure level, or that have a big financial value.
- *Medium*, assets linked to access to common services, not critical, but still important with an intermediate financial value.
- *Low* value for assets of minor importance.

The values have to be assigned taking in to the account the scenarios and the particular use cases.

G. Step 7. Risk Management

The risk management helps to balance between what it is acceptable and what it is possible.

From the threat and vulnerability list it is possible to extract the information about which threat pose the highest risk value. The aspects, which have to be taken into account to assess the risk, are the impact, the damage to the assets when the threat would materialize, the size of the vulnerabilities and the likelihood that the threat will attempt to materialize. The method explained in section 3 helps to rank threats and vulnerabilities, which can be accepted, transferred or mitigated as suggested in [4].

H. Step 8. Mitigation Plan

The last step of the threats analysis is the construction of a mitigation plan that involves the selection of the countermeasures. In this step the threats selected for mitigation must be addressed by one or more countermeasures. To build a mitigation plan it is necessary to identify the countermeasures, i.e. have a list of the countermeasures and a map of the relationship between countermeasures and vulnerabilities, and from this list to select the most effective combination. The decision of which of proposed countermeasures will be included in the actual mitigation plan is taken by the analyst.

The result of the process is a set of proposed countermeasures that would mitigate the threats that were identified. Since the implementation of all the proposed countermeasures is, in most of the cases, impractical due to constraints in budget, time and resources, the goal of a beneficial threats analysis process is to propose the set of the most cost-effective countermeasures against the identified threat [7].

III. METRICS AND MEASUREMENTS TO PRIORITIZE THREATS AND VULNERABILITIES

Valid security metrics -necessary in the last steps of the proposed methodology- are quite a challenging requirement in a threat analysis in order to determine the status of network security performance and to further enhance it by minimizing exposure to considerable threats and vulnerabilities. As it was shown on section 1.1 solutions offered by state of the art techniques on this area are either inadequate or debatable, due to the lack of a standard approach or metric system. Towards these goals, security specialists need to prioritize the threats in the context of the specific system and therefore subsequently need:

- A structured overview of the security context, which would encapsulate all vulnerabilities of the system, their interrelationship and their interaction with security (or worst-case non-security) components in the system.
- A standard measurement expressing the severity and risk of every vulnerability, and therefore the significance of the corresponding threat.

For this purpose, the proposed solution is adopting a combination of Bruce Schneier's Attack Trees [15] and of the Common Vulnerability Scoring System (CVSS) [16][8].

Attack Trees represent attacks against a system in a tree structure, with the goal as the root node and different ways of achieving that goal as leaf nodes. In this way, there can be many potential paths in the tree for an attacker via exploitation of different vulnerabilities and in various ways. In order to build the complete structure of all applying attack trees to the system, assets, threats and vulnerabilities need to be carefully analyzed according to the methodology described previously in this paper. As a second step, tree nodes need to be assigned values in order to evaluate which are the critical paths that need to be counter measured in the attack tree structure. Once this is achieved, optimal paths for attackers are revealed, underlining in this way security enhancement priorities for the analysts. Since attacks may differ significantly, corresponding measurement and value assignment on the node trees is a challenging task. E.g. very sophisticated attacks on a crypto-algorithm [17], network based attacks like Distributed Denial of Service (DDoS) [18][19] and social engineering attacks or user negligence [20] are all attacks which vary significantly in terms of quality and quantity of resources required. Moreover, as each system has its own security requirements, the impact of a specific attack also changes according to the case. A DoS in a banking system would mean significant financial damage for the organization while a DoS in a chat server would mean just a few complaining users. It is therefore evident that a standard measurement needs to be followed, so as to establish a common and stable view for evaluating security. For this reason, the authors are proposing CVSS as the standard way for assigning values on the Attack Tree nodes.

The CVSS is a vendor agnostic, industry open standard designed to convey vulnerability severity and help determine urgency and priority of response. The CVSS assessment divides the vulnerability concerns into three areas:

- Base, for qualities directly characterizing the vulnerability.
- Temporal, for characteristics concerning the vulnerability exploitability and lifetime.
- Environmental, for characteristics dependable to the specific system and environment.

Since the CVSS provides an open framework for vulnerability scoring, security specialists and users throughout the world can exchange views based on the standard measurement, forming also public CVSS vulnerability bulletins for everyone to collect information from. By choosing CVSS for filling the values on the Attack Trees, the authors aimed into combining the benefit from all features belonging to these metric tools, with the main goal being mainly to use attack trees for providing an integrated view of the security of the system, rather than a case specific

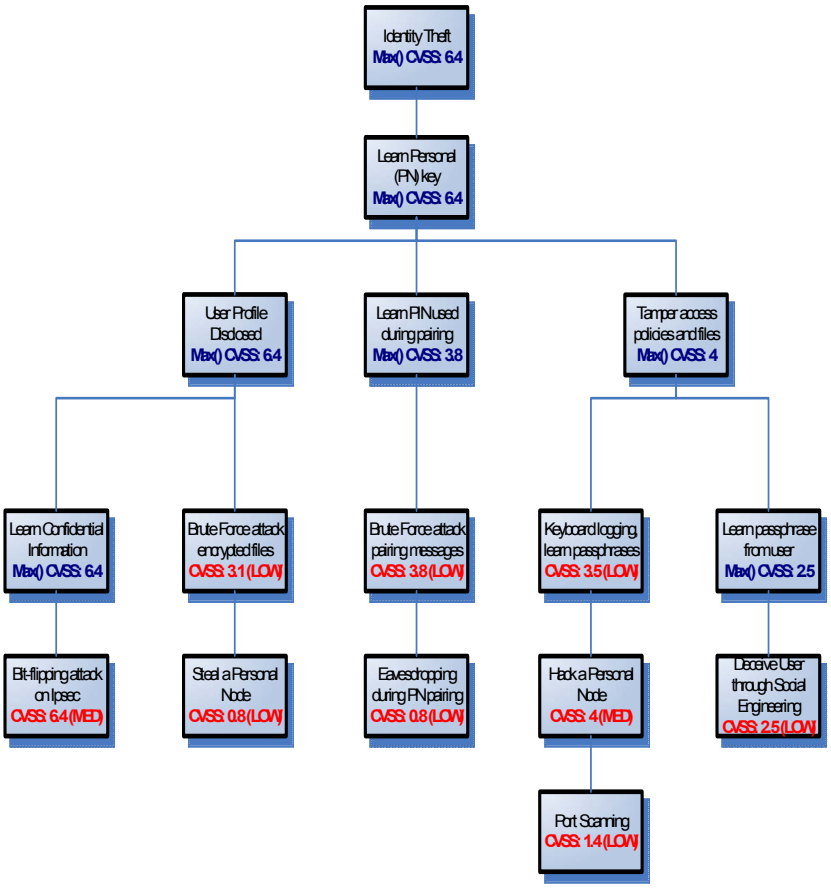

Figure 3. PN Identity Theft Attack Tree

one (e.g. network only security, software and operating systems security), and CVSS for providing individual metrics for each threat connected with each of the tree nodes. Using an open standard further facilitates the construction of the trees, since the whole community contributes in the proper update of the Attack Trees through the CVSS vulnerability bulletins. In order to better understand the CVSS-based Attack Tree security metric methodology and their connection with the proposed threat analysis methodology in this paper, an example case is depicted in Figure 3 for a Personal Network (PN) [21] Identity Theft scenario. The example focuses on the asset: "Identity" of a Personal Network. All threats and vulnerabilities and their interrelations are analysed and the corresponding attack tree is formed. Next, the CVSS scores on tree nodes are either calculated according to CVSS equations either retrieved from vulnerability databases in case of a reported threat such as the Bit-flipping attack on IPsec (vulnerability identified as CVE-2005-0039 [22]). Actual CVSS scoring is performed for values in red, since these nodes correspond to a vulnerability exploitation risk, while nodes in blue correspond to a direct consequence of successful attack(s) and inherit the max CVSS score of all their child nodes scores. Once the tree is properly filled with scores, evaluation and enhancement planning can be achieved by identifying vulnerability paths on the tree leading to the

highest CVSS scores. Using CVSS for quantifying the threat, the threats prioritization is now possible, with the highest priority threat being the bit-flipping attack vulnerability on the IPsec protocol. The added value of using attack trees is clear during the mitigation plan of the system. E.g. according to CVSS alone, hacking a Personal Node is a considerable threat, and is in fact the second priority. However, according to the attack tree this attack is actually a next step following the port scanning attack, which has quite low CVSS score, mainly because firewalls and correct security configuration counter this attack to a great extent. Following this rationale, and always according to security policies and requirements of the specific system, administrators may choose to propose a mitigation plan enhancing nodes in the tree with lower CVSS values, but part of a more consistent attack path (e.g. social engineering attack path).

IV. CONCLUSIONS

In this paper has been proposed a generic methodology that integrates threat modeling with a formal threat analysis for PNs. The methodology is divided in three phases: threat modeling, asset mapping and mitigation plan. A new proposed approach to characterize the system and to identify the assets has been adopted utilizing the UML use case diagrams, whether general or use case specific. The identification of the threats, entry points and vulnerabilities concludes the threat modeling phase. The asset mapping follows in order to be used in risk assessment. The need to quantify threats and vulnerabilities in this phase has been conducted via use of a combined methodology to rank threats and vulnerabilities using both attack trees and CVSS scoring system, as depicted by the provided example for a Personal Network Identity Theft scenario. Finally the mitigation plan can be carried out.

This proposed work is useful in every system to accomplish a complete threats analysis, evaluating threats and vulnerabilities with standard measurement, with emphasis on user-centric architectures such as that of a PNs. Future work will focus on the validation of the proposed approach and the development of a proof-of concept stand-alone complementary security management tool useful for an experienced system administrator, a security analyst and the non-expert user of a PNs..

ACKNOWLEDGMENT

The authors would like to thank the Danish project SUBWAY (Secure SDR-enabled wireless networks ability) founded by CSDR (Center for Software Defined Radio).

REFERENCES

[1] N. R. Prasad “Threat Model Framework and Methodology for Personal Network”, Communication Systems Software and Middleware, COMSWARE 2007.

[2] F. Swiderski, W. Snyder, “Threat Modeling”, Microsoft Press 2004.

[3] E. Oladimeji, S. Supakkul and L. Chung, “Security Threat Modeling: A Goal-Oriented Approach,” Proc. SEA’06, Dallas, TX, Dec. 2006 pp. 178-185.

[4] S. Myagmar, A. Lee, and W. Yurcik, “Threat modeling as a basis for security requirements,” In Proceedings of the Symposium on Requirements Engineering for Information Security, Paris, Aug 2005.

[5] J.D. Meier, A. Mackman, M. Dunner, S. Vasireddy, and A. Murukan, “Improving Web Application Security: Threats and Countermeasures”, Microsoft Press, 2003.

[6] J. Pauli and D. Xu, "Threat-driven Architectural Design of Secure Information Systems", Proc. of the 7th International Conference on Enterprise Information Systems, ICEEIS 2005, Miami, FL, USA, May 2005

[7] Ygor Goldberg, “Practical Threat Analysis for the Software Industry”, http://www.securitydocs.com/library/2848.

[8] CVSS documentation, Forum of Incident Response and Security Teams, http://www.first.org/cvss/cvss-guide.html.

[9] http://www.ist-magnet.org/ MAGNET beyond/D4.4.2 “Analysis, Verification and Evaluation”, June 2008

[10] S. A. Butler, “Security attribute evaluation method: a cost-benefit approach”, Proceedings of the 24th ICSE, May 2002

[11] Indrajit Ray and Nayot Poolsapassit, “Using Attack Trees to Identify Malicious Attacks from Authorized Insiders”, Computer Security – ESORICS 2005, Lecture Notes in Computer Science, Volume 3679/2005, pp. 231-246, Springer Berlin / Heidelberg, 2005.

[12] Igor Kotenko and Mikhail Stepashkin, “Attack Graph Based Evaluation of Network Security”, Communications and Multimedia Security, Lecture Notes in Computer Science, Volume 4237/2006, pp. 216-227, Springer Berlin / Heidelberg, 2006

[13] J. Steven, G.Peterson, “Security lesson learned from Société Générale”, IEEE Security & Privacy, 2008.

[14] G. Stoneburner, A. Goguen, A. Feringa, “Risk Management Guide for Information Technology Systems”, Recommendations of the National Institute of Standards and Technology, July 2002.

[15] Schneier Bruce, "Attack Trees". Dr Dobb's Journal, v.24, n.12, December 1999, web source retrieved on 01-09-2008.

[16] Patriciu Victor-Valieriu, Priescu Iustin, Nicolaescu Sebastian, “Security Metrics for Enterprise Information Systems” Journal of Applied Quantitative Methods, JAQM, Vol 1, No. 2, Winter 2006.

[17] Bruce Schneier, Applied Cryptography, Second Edition. John Wiley & Sons, p. 151, 1996.

[18] Lee Garber, "Denial-of-Service Attacks Rip the Internet," Computer, vol. 33, no. 4, pp. 12-17, Apr., 2000.

[19] Mirkovic, J. and Reiher, P., “A taxonomy of DDoS attack and DDoS defense mechanisms”, SIGCOMM Comput. Commun. Rev. 34, 2, April 2004.

[20] Gragg, David. “A Multi-Level Defense against Social Engineering.” SANS Reading Room, March 13, 2003, URL: http://www.sans.org/rr/paper.php?id=920 (Aug 12, 2003).

[21] Kyriazanos, D., et Al., “Overview of a personal network prototype”, Annual Review of Communications: Volume 59, p.521-534, Intl. Engineering Consortiu, 2007.

[22] Vulnerability Summary for CVE-2005-0039, http://web.nvd.nist.gov/view/vuln/search?execution=e1