INFORMATION SECURITY ATTACK TREE MODELING

An Effective Approach for Enhancing Student Learning

Jide B. Odubiyi and Casey W. O'Brien

Department of Computer Science, Bowie State University, Bowie, MD – jodubiyi@cs.bowiestate.edu

Network Technology Department, Community College of Baltimore County, Baltimore, MD – cobrien@ccbcmd.edu

Abstract: This paper presents a framework for enhancing student learning about the vulnerabilities of information assets of a business enterprise using attack tree modeling. Using this framework, students get an overview of the methodology as well as learn how to implement it with a well-known list of information security vulnerabilities. As a result, students can provide input into threat modeling strategies and operating procedures and thus, increase overall confidentiality, integrity, and availability of computer and network systems. The paper concludes by describing a corresponding system capability metric that a system administrator, student, or red team can use to test the vulnerabilities of the systems within a business enterprise.

Keywords: Attack Trees, Threat Modeling, Intrusion Scenarios, Attack Scenarios, Business Enterprise, System Capability Metric, Failure Mode and Effect Analysis

1. INTRODUCTION

For years, engineers have relied on the analysis of failure data to improve their designs by employing the engineering principle of Failure Mode and Effect Analysis (FMEA). The main goal of this principle is to identify risks and initiate concerted efforts to control or minimize these risks. By knowing the risks, the project plan can be created more realistically.

FMEA techniques [1] help to identify failure potential in a design or process before it actually happens. This is done by following a procedure which helps to identify design features or process operations that could fail. This procedure requires the identification of all potential failures, including the causes and the effects of those failures. Using knowledge of the design, process, and/or system, estimates of the probability of these failures can be identified, together with the severity of the effects of the failure, and the probability of detecting the failure before it becomes critical.

While engineers have historically been good at using failure data to improve their designs and systems, software engineers and information systems administrators have not. In general, IT professionals dealing with the security of systems do not use similar failure data - attack data - to improve their computer and network systems and the related components that are a part of those systems. The reasons for this vary from organizations that are weary of divulging such information (for fear of decreased public confidence), to fears that attackers will employ the attacks against their systems, to a lack of detailed and reliable data on attacks [2].

Despite weariness on the part of organizations to disclose attacks on their systems, attack data have become more available in recent years, due in large part to the public and media's increased attention on information security issues. In addition to the public and media interest in information security-related issues, other resources like the SANS Internet Storm Center [3] and Security Focus [4], as well as many Computer Emergency Readiness Teams (CERTs) and Coordination Centers (CCs) throughout the world bring together timely and comprehensive security-related information. These resources serve as a forum for in-depth discussions and announcements of computer and network security vulnerabilities. They detail what they are, how to exploit them, and how to fix them.

Countless articles and research publications have been written detailing threat models that can be used to test the security of an enterprise system. The authors of these publications argue that understanding all of the different ways a system can be attacked can help IT professionals responsible for securing systems design and implement countermeasures, to thwart those attacks. In addition, if these same IT professionals can understand who the attackers are and what their abilities, motivations, and goals are, they can implement the proper countermeasures to deal with the real threats [5].

A fundamental understanding of threat modeling can help system developers/integrators build robust and reliable systems. If they know the possible threats, they can use an attack tree to test their systems. For example, after building a Web site, a system administrator can readily apply a threat model to test it. The same is true of students learning about the vulnerabilities of the information assets of a business enterprise. By developing several attack trees (which constitute an attack forest of a business enterprise), students get an appreciation for the hundreds of possible vulnerabilities in a given enterprise system, and the challenges faced by system administrators when securing their computer and network systems. Ultimately, students would then use this model to further test the security of their systems. Several organizations, including The General Accounting Office and the U.S. Department of Homeland Security, employ Red Teams to identify vulnerabilities in their information assets [6].

2. TEACHING THREAT MODELING

Students cannot build and test the security of computer and network systems unless they understand the threats posed to those systems. Therefore, it's critical that students in Information Assurance (IA), Computer Science, network technology, and other IT related programs be taught not only how to design and build secure systems, but also how to identify the means, motives, and opportunities of their adversaries, identify the threats posed to the systems they are securing, and ways to test it.

When the above mentioned academic programs are infused with threat modeling concepts and implementation strategies for testing the security of systems, students can do more. They can extend the application of this framework to design systems that meet the security objectives of an organization, decide on trade-offs during key design decisions, and help reduce the risks of security-related issues arising during the implementation and operations phases.

3. THE CLASSROOM ENTERPRISE

The following is an example of how we introduced and enhanced student's understanding of threat modeling. We asked students to compile a list of information assets managed by the university or a typical medium business enterprise. The list of information assets includes several Web servers and Database Management Systems for Payroll, Strategic Plans, etc. We then provided the students with 32 information system vulnerabilities including the Twenty Most

Critical Internet Security Vulnerabilities [11] and asked them to develop attack trees for them. By developing several attack trees which constitute an attack forest of a business enterprise, students moved from a conceptual grasp of threat modeling, to an appreciation for the hundreds of possible threats to a system, to the challenges faced by system administrators, to methodologies for testing the security of a system.

3.1 The Adversary Model

Before students are able to design, build, and implement secure systems, they must understand the means, motives, and opportunities of their adversaries [5, 9]. Salter, et al [9] developed an Adversary Model to help organizations thoroughly understand their adversaries and characteristics. This model characterizes adversaries in terms of their resources, access, risk tolerance, and objectives. The learning objective for students is that countermeasures are only needed for attacks that meet the adversaries' resources and objectives, but to be useful, the countermeasures must also meet the organization's needs for cost, ease of use, compatibility, performance, and availability.

3.2 Identifying and Classifying Assets

The task of identifying assets that need to be protected within an organization is one of the less glamorous aspects of information assurance. Unless an organization know its assets, where they are located, and what their value is, it's difficult to decide on the amount of time, effort, and money that should be spent on securing them. In addition, organizations won't know what the adversaries of their systems want until they've identified the sensitive information and related assets on those systems. Students should explore asset classification models, which can be categorized as follows: the identification of an organization's assets, the accountability of those assets, preparing a framework for classifying those assets, and implementing the classification framework.

3.2.1 The Enhanced Telecom Operations Map for the Telecom Service Industry

Figure 1 can serve as an excellent resource to give the learner an appreciation for the processes required for managing and protecting a telecommunication business enterprise. The enhanced Telecom Operations Map (eTOM) has been developed by The Telecom Management Forum as a generic framework for organizing a telecommunication business organization into three major areas. The three areas consist of (a) Planning and Lifecycle Management of the Strategy, Infrastructure, and Product development; (b) Operations Management, and (c) Enterprise Management—Business Support management. Each of the three areas consists of computer systems, telecommunication networks and software that must be protected. The responsibility for protecting these assets falls on the shoulder of the chief security officer or the Information Technology (IT) department. As a result of mergers and acquisitions, there can be tens of software products just for the Billing operation. At one time, the former WorldCom had over 30 billing systems. Each billing system consisted of database management systems, Web servers, and accounting software. Other operations support systems (OSSs) consist of diverse software products. The eTOM framework provides an environment for students to identify OSSs to analyze for vulnerabilities.

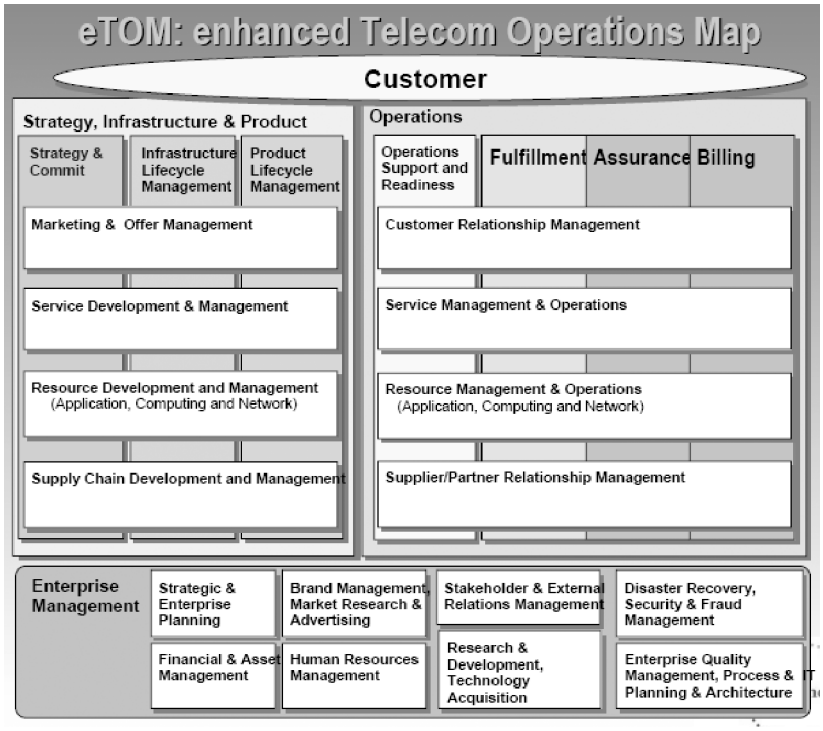

Figure 1. The Enhanced Telecom Operations Map® (eTom) (Source: The Telemanagement Forum) [15]

3.3 Creating an Architectural Overview

When students are creating an architectural overview of an enterprise system, they need to be explicit about what the network and related systems are designed to do, how they plan on engineering it to achieve that functionality, and what technologies are required to implement the design. This helps them identify the common technology-specific threats and implement solutions to overcome them [7]. In addition, having students create a diagram of how information flows throughout the organization allows them to discover the critical components and procedures of the enterprise system. The student can use the eTOM model in Figure 1 or the instructor may identify a similar map for another industry.

3.4 The Sum of Its Parts

Once students have an overview of the architecture of the enterprise, they can use it to break down the system into its constituent components. The more an organization knows about its system, the easier it is to discover threats against it. The steps involved in breaking down a system into its constituent parts starts with creating a security profile, that is, an analysis of the system's strengths and weaknesses. Students will need to validate all data being sent across their systems. This includes a comprehensive examination of the trust boundaries of the enterprise network, the flow of data in and out of the enterprise, and any entry points into the system that will ensure that the flow of information is done in a secure manner [8].

3.5 Identifying Threats Using Attack Trees

The first step is to identify the threats that might affect an organization's system and potentially compromise its assets. This includes identifying network, host, and application threats. Network threats can be assessed by having students investigate how the data passes through routers, firewalls, switches, and other network devices. Students need to understand the logic and syntax of these device's configuration files. In addition, students need to be able to determine what it takes to get past or compromise each device. Host investigations should include common configuration categories applicable to all server and operating system resources (patches, files/directories, ACLs). Finally, students should examine the enterprise's application software [7].

There are other ways students can identify the threats to a given enterprise system, namely using a list of known threats [11] and using attack trees with attack patterns. We encourage using a combination of both to identify threats to a system. When only using a previously prepared list of known threats, the resulting threat list only reveals the common, known threats. Using attack trees, in conjunction with a list of known threats, can help identify other potential threats.

3.6 Attack Trees and the SANS Top-20 Vulnerabilities

An attack tree provides a method for representing attacks (and similar vulnerabilities) on a system in the structure of a tree. The goal of the tree is the root node. The leaf nodes represent different paths to achieve the goal. As depicted in Figure 1, a business enterprise typically consists of hundreds of information assets that are vulnerable to attacks. Each of these assets can be modeled as an attack tree resulting in an attack forest [5]. The root of each tree in the attack forest constitutes an event that can potentially damage the enterprise system and its related resource's confidentiality, integrity, and availability. The SANS twenty most critical Internet security vulnerabilities provide a rich environment for the students to learn about well documented attacks [11]. The SANS top-20 2004 vulnerabilities consist of two top-10 vulnerable services in Windows and UNIX operating systems. The SANS top-20 list is updated annually and provides useful information about each vulnerability, systems affected, and how to protect the services. Students can use the protection measures to develop attack scenarios.

3.6.1 The Mail Transport Service and Enterprise Service (NIS/NFS) Vulnerabilities

Two of the twenty critical vulnerabilities identified by SANS are the Mail Transport Service (MTS) and the misconfigured enterprise services: Network File System (NFS) and Network Information Service (NIS).

The Simple Mail Transfer Protocol (SMTP) is one of the oldest Internet application protocols that employ Mail Transport Agents (MTAs) as servers to send emails from senders to recipients. Examples of these MTAs are sendmail, QMail, Courier-MTA, Postfix, and Exim. They are vulnerable to buffer/heap overflow attacks when they are not patched or when their patches are out of date. An attacker can compromise the Mail Transport Service by exploiting open relays (i.e., a situation that permits the sender and receiver that are not part of the domain to send and receive mail). A third major vulnerability of the MTS is exploitation of non-relay problems such as misconfiguration of the user account database, thereby exposing the service to spam attacks.

The NFS and NIS services are used in UNIX servers to hold directories of files systems and to provide locations of distributed databases respectively. They are vulnerable to buffer overflow attacks due to unpatched services, Denial of Service (DoS) attacks and weak authentication. Given the goal of attacking either the MTA or the three sub-goals listed above, the student can be assigned to develop attack scenarios for the services using the approach described in the next paragraph.

3.7 Modeling an Attack Tree

The Web server is one of the SANS top-20 most critical vulnerabilities due to add-on software modules such as CGI scripts, PHP bugs, servelets, etc. The process for modeling an attack tree for a Web server's vulnerability is described in this paragraph. Since each tree has a root node that represents the attacker's goal, and the leaf nodes represent different paths to the root, each child node represents the steps an attacker can take. Modeling the attack tree involves associating a logical AND and a logical OR with each node. In essence, a node of an attack tree can be decomposed into an AND or an OR node. An AND node or an OR node decomposition can be represented in graphical or textual formats. Figure 2 is a textual representation of one of the top vulnerabilities to Windows-based systems [11]:

GOAL: (G0) Gain Privileged Access to a Web Server Using a Known Vulnerability [8]

AND G1. Identify organization's domain name.

G2. Identify organization's firewall IP address

OR 1. Interrogate domain name server

2. Scan for firewall identification

3. Trace route through firewall to Web server

G3. Determine organization's firewall access control

OR 1. Search for specific default listening ports

2. Scan ports broadly for any listening port

G4. Identify organization's Web server operating system and type

OR 1. Scan OS services' banners for OS identification

2. Probe TCP/IP stack for OS characteristic information

G5. Exploit organization's Web server vulnerabilities

OR 1. Access sensitive shared intranet resources directly

2. Access sensitive data from privileged account on Web server

Figure 2. Web Server Attack Description

As presented in Figure 2 above and stated earlier, a node of an attack tree can be decomposed into an AND or an OR node. Both the AND and the OR decompositions can be represented in graphical or textual format as shown in Figures 3 and 4 below. All the attack sub-goals (such as G1, G2, G3, …., G5) must be achieved for the exploit to succeed.

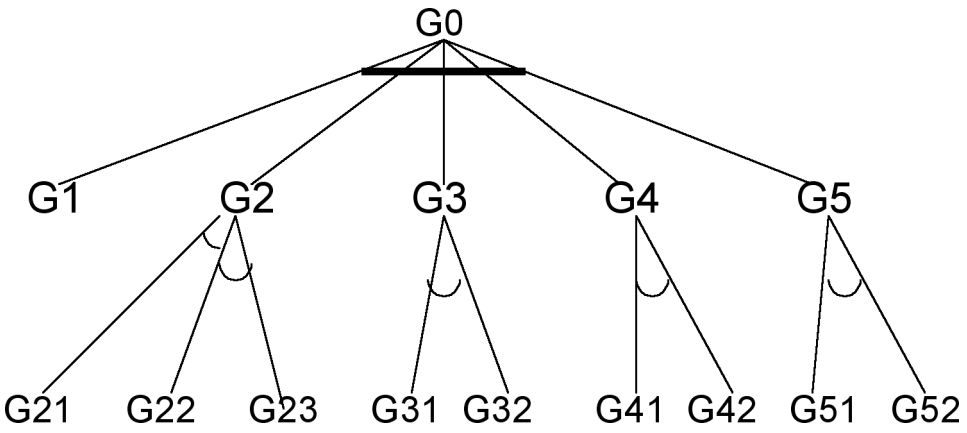

Figure 3. A Graphical Representation of an Attack Tree described in Figure 2
(G0 ≡ G1 ∩ G2 ∩ G3 ∩ G4 ∩ G5); (G2 ≡ G21 ‖ G22 ‖ G23)

The attack tree in Figure 3 can generate 24 attack or intrusion scenarios [10] in Figure 4 below. The 24 possible attack scenarios are:

[G1, G21, G31, G41, G51], [G1, G21, G32, G41, G51],
[G1, G21, G31, G42, G51], [G1, G21, G32, G42, G51],
[G1, G22, G31, G41, G51], [G1, G22, G32, G41, G51],
[G1, G22, G31, G42, G51], [G1, G22, G32, G42, G51],
[G1, G23, G31, G41, G51], [G1, G23, G32, G41, G51],
[G1, G23, G31, G42, G51], [G1, G23, G32, G42, G51]
[G1, G21, G31, G41, G52], [G1, G21, G32, G41, G52],
[G1, G21, G31, G42, G52], [G1, G21, G32, G42, G52],
[G1, G22, G31, G41, G52], [G1, G22, G32, G41, G52],
[G1, G22, G31, G42, G52], [G1, G22, G32, G42, G52],
[G1, G23, G31, G41, G52], [G1, G23, G32, G41, G52],
[G1, G23, G31, G42, G52], [G1, G23, G32, G42, G52]

Figure 4. Twenty-four possible attack scenarios

The realization that a simple Web server attack tree can generate 24 attack scenarios helps the student to understand the importance of developing secure software as a countermeasure against Web server exploits.

3.8 Benefits of Documenting the Threats: From an Attack Tree to an Attack Forest

Figure 3 is one of hundreds of vulnerabilities that make up an organization's attack forest. By creating an attack forest, an organization not only has a roadmap for testing for both known and unknown vulnerabilities, but also a means to document the threats to their systems and assets.

Ultimately, the attack forest document for an organization will include; each threat; a description of the threat; the target of the attack; the risk of the attack; the techniques likely to be used in carrying out the attack; and a risk management strategy. For example, when dealing with a SQL injection attack, the target is the database and the technique is that the attacker types a command into a textbox that is automatically added into a T-SQL command without client-side validation. To counter this threat, one can use regular expressions to validate the user name, and use a parameterized query to access the database [12].

4. CHALLENGES

Teaching threat modeling can be difficult for a number of reasons, each of which is described below.

4.1 From Past to Present

Organizations trying to protect their assets from attackers and current intrusion detection tools face a common problem: Being able to generalize from previously observed behavior to recognize future behavior, either malicious or normal.

Signature-based misuse detection techniques are acutely prone to this problem, as are anomaly-based detection tools that must determine normal behavior that is not identical to past observed behaviors, in order to reduce false alarms [13].

By understanding this problem, students can use both advances in intrusion detection techniques and adversary models as ways to better understand who their attackers are, and ultimately, implement the proper countermeasures to deal with the threats.

4.2 Input Data: Developing Attack Trees

As stated earlier, there are multiple ways students can identify the threats to a given enterprise system. We encourage using a combination of known threats [11] as the basis for developing an enterprise attack forest. When combining known lists of vulnerabilities with attack trees and attack trees' patterns, new vulnerabilities can be identified.

5. FUTURE WORK

If a simple attack tree (like the one outlined in Figure 3) can generate 24 attack scenarios and there are hundreds of attack trees in a medium-sized business, it is feasible to have thousands of possible attack scenarios. Most organizations do not have all the resources that enable their system administrators to exhaustively test all of the thousands of possible scenarios. We are working on developing a framework that will generate a sampling scheme that will provide a degree of confidence metric to aid systems administrators and Red Teams in selecting scenarios to test.

At the conclusion of their testing and using the sampling scheme, the testing team will be able to claim, with a degree of confidence, that 90% or 95% or 98% of all the attack scenarios have been tested. We have successfully demonstrated the approach to resolve aircraft to aircraft in-flight conflict scenarios and maneuver strategies for the Federal Aviation Administration (FAA) and in testing porosity of graphite composites at the Boeing Aircraft Company in Seattle, and we are optimistic of the potential success in the information assurance domain and its educational value in enhancing student learning.

REFERENCES

1. Shooman, M. L. Probabilistic Reliability: An Engineering Approach. McGraw-Hill Book Company, 1968.
2. Anderson, R., Why Cryptosystems Fail, Proceedings of the 1st ACM Conference on Computer and Communications Security, 1993.
3. SANS Internet Storm Center, http://isc.sans.org
4. Security Focus, http://www.securityfocus.org
5. Schneier, B., Attack Trees, *Dr. Dobb's Journal*, Vol. 24, no. 12, December, 1999, pp. 21-29.
6. Ray, H. T., Vemuri, R., & Kantubhukta, H. R. Toward Automated Attack Model for Red Teams, IEEE Security & Privacy, Vol. 3, No. 4, IEEE Computer Society, July/August 2005, pp. 18-24.
7 Howard, M. and LeBlanc, D., Writing Secure Code, 2nd Edition, Microsoft Press, 2002.
8. Meier, J.D., Mackman, A. & Wastell, B., Walkthrough: Creating a Threat Model for a Web Application. Retrieved August 27, 2005, from http://msdn.microsoft.com/security/securecode/threatmodeling/default.aspx?pull=/library/en-us/dnpag2/html/tmwawalkthrough.asp#tmwawalkthrough_breakingdowntheapplication
9. Salter, C., Saydjari, O.S., Schneier, B., & Wallner, J., Toward a Secure System Engineering Methodology, New Security Paradigms Workshop, September, 1998.
10. Moore, A. P., Ellison, R. J., & Linger, R. C. Attack Modeling for Information Security and Survivability. Technical Note: CMU/SEI-2001-TN-001., March 2001.
11. The SANS Institute, The Twenty Most Critical Internet Security Vulnerabilities (Updated) – The Experts Consensus, http://www.sans.org/top20/
12. Threat Model Your Security Risks, Microsoft Security Developer Center. Retrieved on August 27, 2005, from http://msdn.microsoft.com/security/securecode/threatmodeling/default.aspx?pull=/msdnmag/issues/03/11/resourcefile/default.aspx
13. Ghosh, A.K., Schwartzbard, A., A Study in Using Neural Networks for Anomaly and Misuse Detection, Proceedings of the 8th USENIX Security Symposium, August 23-26, 1999, Washington, D.C.
14. Mackman, A, Meier D.J.Meier,& Wastell B., Threat Modeling Web Applications. Retrieved July 29, 2005, from http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpag2/html/tmwa.asp
15. The TeleManagement Forum www.tmforum.org