A Practical Approach
to
Threat Modeling

Tom Olzak
March 2006

Today's security management efforts are based on risk management principles. In other words, security resources are applied to vulnerabilities that pose the greatest risk to the business. There are several processes for identifying and prioritizing risk. One of the most effective is threat modeling.

There has been much written about threat modeling. But most of the papers and books come at the problem of threat and vulnerability management from an academic perspective. The papers and articles that do take a business management approach typically cover one or two aspects of the process.

This paper is a practical, high-level guide to conducting threat modeling activities within a business environment. It begins by exploring why threat modeling is important. This is followed by a step-by-step process, including some tools you might find helpful.

Why Threat Modeling?

It's common for security teams to receive reports of vulnerabilities with requests for immediate action to eliminate them. One big source of these requests is an organization's internal audit team. Another common source of fix-it-now-because-the-press/vendor-says-it's-critical messages is management, including many IS Directors. But should all vulnerabilities be considered emergencies? Are all vulnerabilities worthy of your security budget dollars?

One of the basic tenets of risk management is that not every vulnerability presents a threat to a network. Only a vulnerability that can be exploited is a threat to business operations and information assets. Threat modeling helps to identify those vulnerabilities that are actually critical in the unique environment that is your network. The threat modeling process should:

1. Identify potential threats and the conditions that must exist for an attack to be successful
2. Provide information about how existing safeguards affect required attack conditions
3. Provide information about which attack condition and vulnerability remediation activities add the most value
4. Help you understand which conditions or vulnerabilities, when eliminated or mitigated, affect multiple threats; this optimizes your security investment

The Process

The description of the threat modeling process varies depending on who's doing the telling. The following process is based on research covering several different approaches. Based on my experience as a security manager, I took what I believe to be best practices and compiled them into a hybrid model. This model consists of six steps, or phases:

1. Identify critical assets
2. Decompose the system to be assessed
3. Identify possible points of attack
4. Identify threats
5. Categorize and prioritize the threats
6. Mitigate

Identify Critical Assets

Before spending time assessing a system, you need to be sure it's important enough to your business to warrant the necessary time and resources. In this first step, you should list all critical assets and the systems on which they reside. Whether an asset is critical to business operations isn't an IS-only decision. The business users must also play a part in determining which assets can't be compromised without serious negative consequences.

Decompose the System

Once you identify your critical assets, select a system for which you'll create a threat model. A system is defined as an environment within your network that provides a specific set of related functions. Your human resources application, with all related servers, routers, switches, operating systems, user workstations, etc. is an example of a system. System decomposition produces two deliverables: a network diagram and a functionality (interaction) diagram. Figure 1 is an example of a network diagram.

The format of the diagram is a variation of the UML, or Unified Modeling Language standard. Each component in the ESI Financial System (a fictitious entity) is represented by a box. Each Workstation and server box includes information about the corresponding real-world device's hardware and software configuration. In addition to the actual hardware connectivity, logical flow of data is also indicated. Finally, the network diagram should include interfaces to outside entities. In this case, the connection to the Internet is depicted.

It's a common mistake when putting a network diagram together to omit pieces that aren't considered critical to the system's operation. Make sure you include EVERY component, interface, and user access point that touches the system in any way. Also identify any interdependencies with other systems.

Figure 2 is an example of a simplified Functionality or interaction diagram.

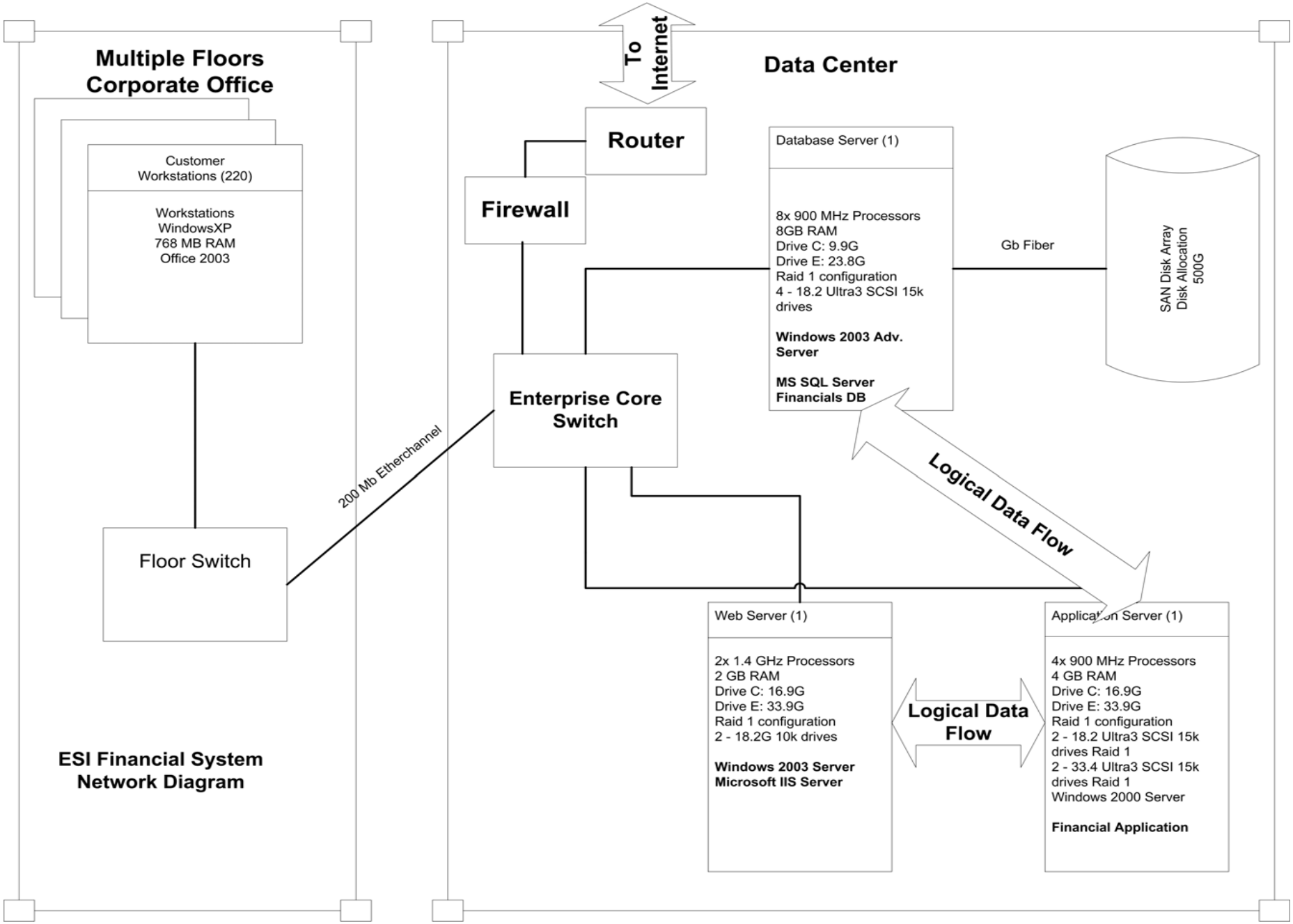

Figure 1: Network Diagram

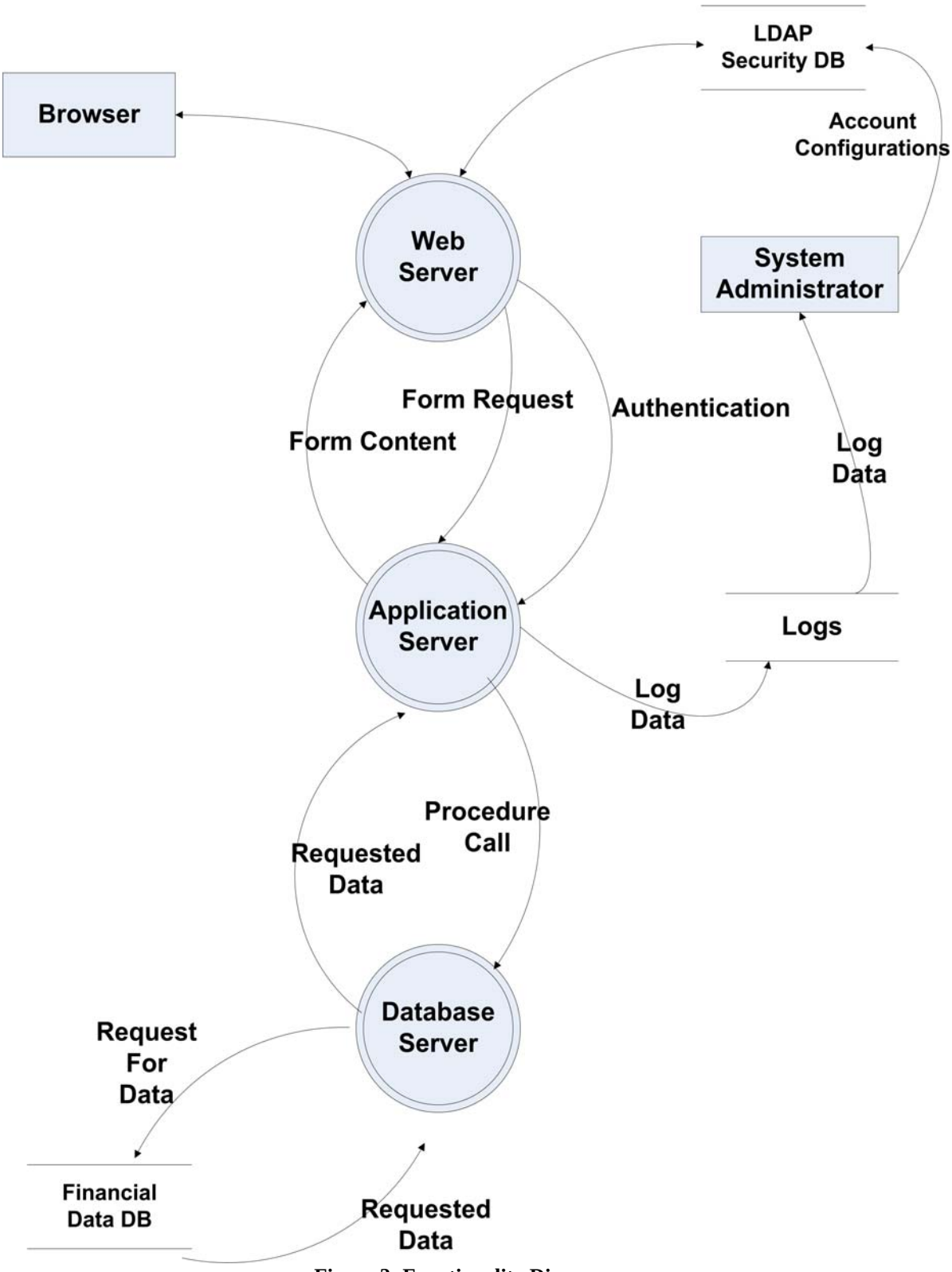

Figure 2: Functionality Diagram

This functionality diagram uses a DFD (Data Flow Diagram) approach to show the functional relationships between the various system components. Although I used device names in the circular component symbols, analysts often use the names of software components instead.

The level of detail in both the network and functionality diagrams is up to you. Just be sure to include enough information to ensure the threat modeling results are accurate.

Identify Possible Points of Attack

The first step in the identification of attack points is designating trust boundaries. A trust boundary separates processes, system components, and other elements that have different trust levels. Figure 3 shows the ESI Network Diagram with trust boundaries added.

Trust boundaries also exist at all entry points into the system. Classify each entry point based on the classification of the data exchanged. Table 1 lists example data classifications. If the highest classification for data moving across an entry point is Restricted, then the entry point must be classified Restricted. Examples of entry points include sockets, interfaces between application components, and user workstations.

At each trust boundary, identify the types of safeguards that provide access controls. This information is required when completing attack trees.

Table 1: Data Classifications

Classification	Description
Restricted	Applies to the most sensitive business information. The unauthorized disclosure of this information could result in a serious negative impact on the company, its customers, its business partners, and its suppliers.
Confidential	Applies to less sensitive business information. The unauthorized disclosure of this information could result in a negative impact on the company, its customers, its business partners, and its suppliers.
Public	Applies to information approved for public disclosure.

Identify Threats

The next step is to list any critical activities that take place at each trust boundary. Using this list, determine what an attacker might do to damage, destroy, or otherwise

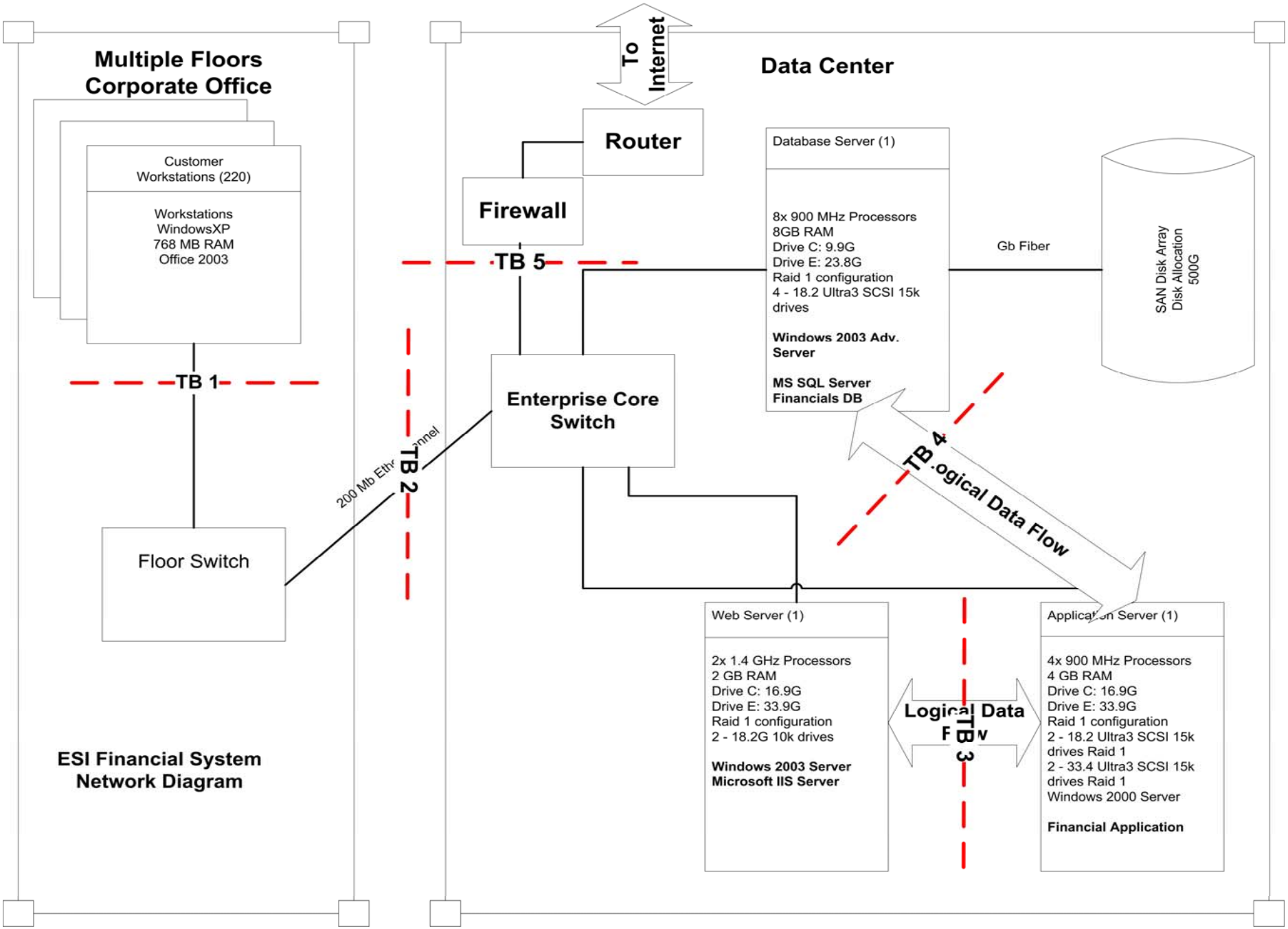

Figure 3: Network Diagram with Trust Boundaries

compromise your system and its data. There are two ways to approach threat identification: use of the STRIDE method and a step-by-step analysis.

STRIDE

STRIDE is an acronym. The terms/phrases it represents, along with an explanation of each, are listed in Table 2. At each trust boundary (TB), apply the STRIDE model by asking whether one or more of the threat types represented apply. If so, include it on your list of potential attack goals.

Step-by-step analysis

STRIDE is a very simple approach to threat identification. Because of its simplicity, its use tends to result in one or missed threats per TB. Using a step-by-step analysis typically produces a more complete threat list. One step-by-step method is a review of specific threats organized into three categories: network threats, host threats, and application threats (Chidambaram, 2004).

Table 2: STRIDE

STRIDE (MSDN)	
Letter	**Stands for…**
S	**Spoofing Identity** - Impersonating someone else to the computer
T	**Tampering with Data** - The malicious modification of data
R	**Repudiation** - Involves users who can deny performing an action without other parties having any way to prove otherwise
I	**Information Disclosure** - Involves the exposure of informaion to individuals who are not supposed to have access to it
D	**Denial of Service**
E	**Elevation of Privilege** - An unprivileged user gains privileged access and thereby has enough access to compromise or destroy the system

The following are examples of the threats included in each of the categories.

Network Threats

- Web services subjected to a denial of service attack
- IP spoofing
- Faulty configuration of firewall rules, allowing outsiders to get access to a database and change the data
- Errors in ACLs
- Sensitive data that flows unencrypted through the network

Host Threats

- Using un-patched servers allows crackers to exploit known vulnerabilities
- Lack of clearly defined trust boundaries
- Improper server hardening guidelines resulting in a mismatch between the server configuration and the security context in which it's placed

Application Threats

- Code that's prone to buffer overflows, SQL injection, or cross-site scripting
- Defective or missing data encryption resulting in password compromise

Once you complete your list of threats, it's time to build the system's attack trees. Attack trees are useful when capturing attack patterns that require events to occur in sequence. They add less value when analyzing attacks comprised of parallel events (Ellison, 2005). Figure 4 is an example of an attack tree.

An attack tree is a tree structure with the attacker's objective placed in the root node. In this example, the objective is to obtain sensitive information from the database server in Figure 3. Working down the branches of the tree, the analyst decomposes the attack into its various options and required conditions. At the first layer under the primary objective, our tree lists potential entry points to obtain server access. Notice that the relationship between these elements is OR; only one entry point has to be successfully exploited to obtain information from the server.

In an actual attack tree, the analyst would drill down into each of the top level nodes. For our example, we'll use *Gain access via Internet*. To successfully exploit this vulnerability, port 1434 must be open on the firewall for general access AND the server's subnet must be open to general traffic rather than protected by an access control list. If ESI has very stringent policies and standards for opening this port, then this attack path might be already impossible to travel. So a recommendation from an auditor to implement a restrictive ACL in the core switch might be a best practice, but it probably wouldn't be critical to

the protection of the database server from Internet attack. This is a very simple, incomplete example. But you should get the idea.

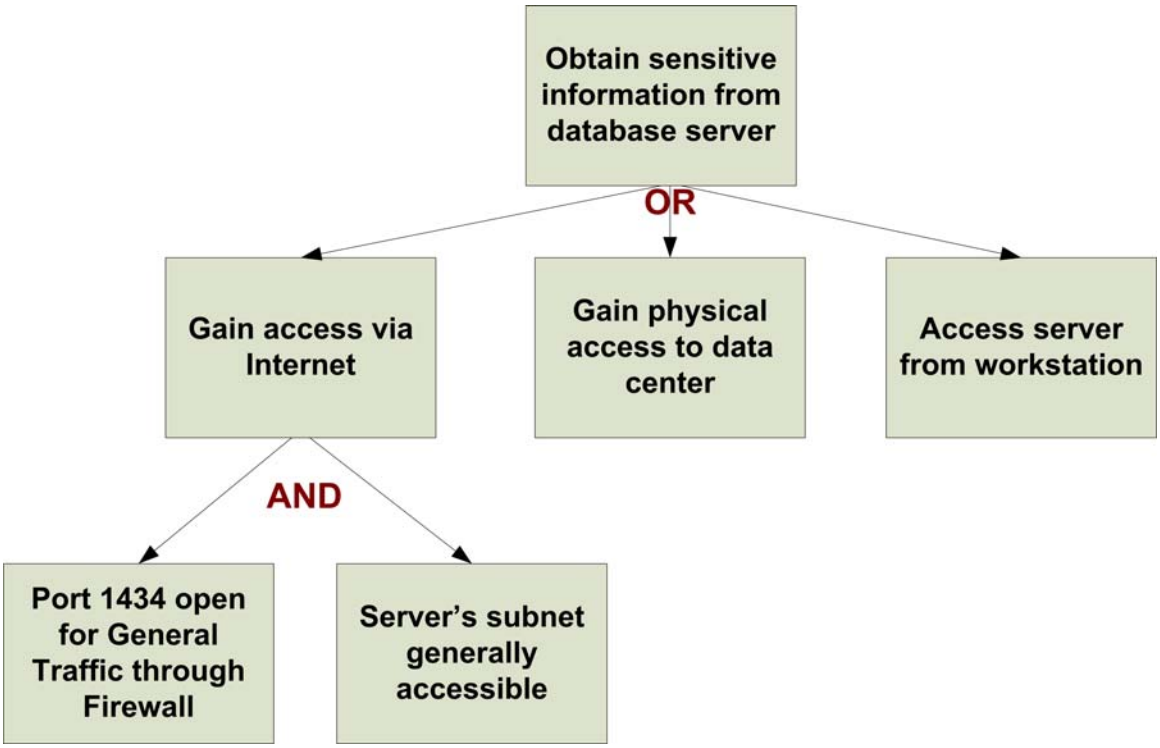

Figure 4: Attack Tree

The attack tree in Figure 4 can be used in at least three ways to help determine which threats and vulnerabilities should be addressed and in what order: probability of occurrence, cost/effort of mitigation, and whether one or more vulnerabilities are mitigated. Activities designed to address these risk management areas occur in the next step in the threat modeling process.

Categorize and Prioritize Threats

In an organization where threat and vulnerability management is governed by solid risk management principles, the following formula is typically used to assign a risk score to a threat:

Risk = Probability of Occurrence x Business Impact

There are a number of ways, both qualitative and quantitative, to apply this formula. For the purposes of our threat assessment model, I'm going to use DREAD. DREAD (yes, another acronym) is a collection of five areas with which to assess both probability of occurrence (PO) and business impact (BI). Table 3 lists these areas.

Each one of the areas is given a score of 1, 2, or 3, with 3 being the highest level of potential risk to the business. To map the DREAD areas to the risk formula, I created a tool in which to enter the scores. The tool, an Excel spreadsheet, automatically calculates the risk score for the threat analyzed. See Figure 5.

Table 3: DREAD

DREAD (Meier et al, 2003)				
Letter	Rating	High (3)	Medium (2)	Low (1)
D	Damage Potential	The attacker can subvert the security system; get full trust authroization; run as administrator; upload content	Leaking sensitive information	Leaking trivial information
R	Reproducibility	The Attack can be reproduced every time and does not require a timing window	The attack can be reproduced, but only with a timing window and a particular race situation	The attack is very difficult to reproduce, even with knowledge of the security hole
E	Exploitability	A novice programmer could make the attack in a short time	A skilled programmer could make the attack, then repeat the steps	The attack requires an extremely skilled plerson and in-depth knowledge every time to exploit
A	Affected Users	All users, default configuration, key customers	Some users, non-default configuration	Very small percentage of users, obscure feature; affects anonymous users
D	Discoverability	Published information explains the attack. The vulnerability is found in the most commonly used feature and is very noticeable	The vulnerability is in a seldom-used part of the product, and only a few users should come across it. It would take some thinking to see malicious use	The bug is obscure; and it is unlikely that users will work out damage potential

Risk Calculation						
	Probability		Business Impact			
Description	Reprod.	Exploit.	Damage Pot.	Affected Users	Disc.	Risk
Obtain senstive information from database server - Internet Path with Port blocked at firewall	1	1	1	3	3	**14**
						0
Obtain sensitive information from database server - Internet Path with port open at firewall	3	2	2	3	3	**40**

Figure 5: Risk Calculation Tool
(Available for free download at http://adventuresinsecurity.com/blog)

I grouped reproducibility and exploitability under PO. The other DREAD areas fall under BI. I've seen Discoverability (Disc.) attributed to both PO and BI. In my opinion, the ability to quickly discover a potential or current attack is critical to mitigating impact. The risk score is calculated as follows:

Risk = PO x BI
Risk = (Reprod. + Exploit.) x (Damage Pot. + Affected Users + Disc.)

Risk can be calculated just for the root node. But this doesn't take into account the risk levels associated with the various attack paths. I prefer to score each potential attack vector. The risk score for the threat is equal to the highest attack path risk score. In this way, I get a good picture of which attack paths contribute most to the overall criticality of the threat. Deciding if or how to apply resources is more effective.

In the example in Figure 5, I calculated the risk score for two scenarios. In the first scenario, Port 1434 is blocked from general access. The resulting risk score is 14. In the second scenario, the port is wide open, increasing the risk score to 40. Table 4 is a recommended score translation.

Table 4: Score Translation

Score	What it means…
1-18	**Low Risk -** Consider accepting this risk.
19-36	**Medium Risk -** There is the potential for moderate damage to the business information assets, finances, or reputation. A plan should be put into place to mitigate this risk as soon as possible.
37-54	**High Risk -** There is the potential for serious damage to the business information assets, finances, or reputation. A plan should be put into place to mitigate this risk immediately.

After calculating the risk scores, you can add them to the attack tree as depicted in Figure 6. In this example, assume that one or more of the conditions required to obtain physical access to the data center or to access the server from a corporate workstation were already mitigated.

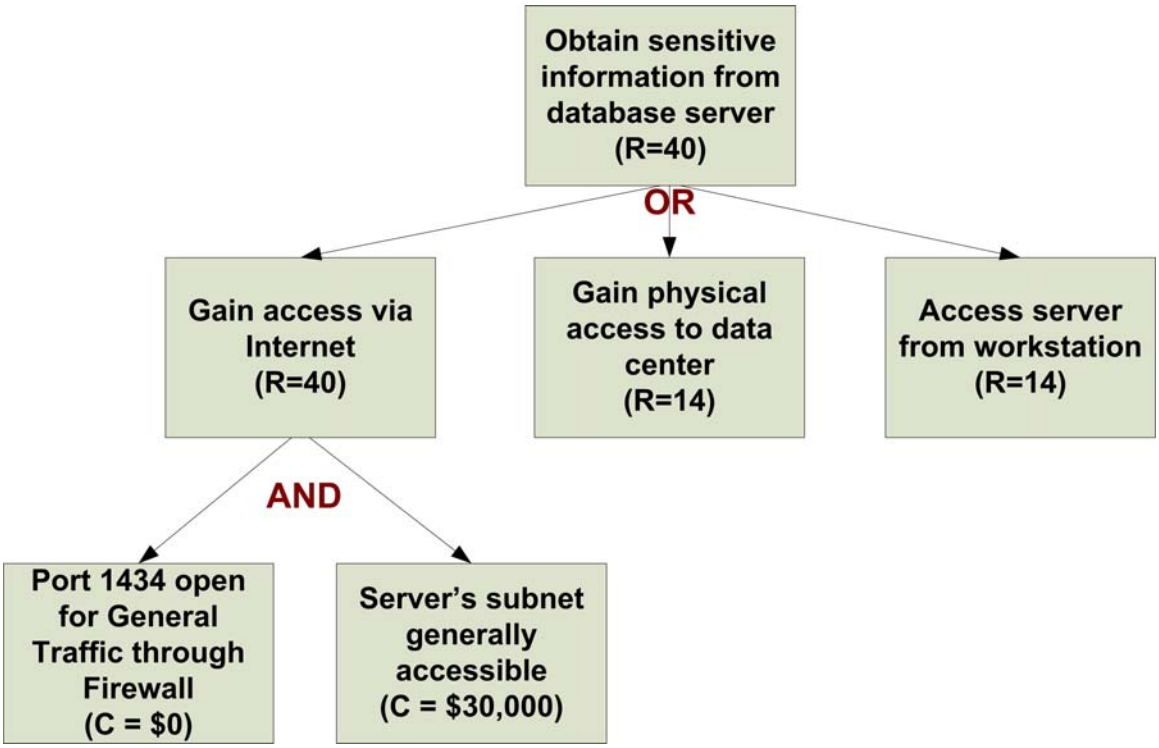

Figure 6: Attack Tree with Risk Scores

Mitigate

The information gathered in the previous step is used as input into the "do-something" mitigation step. What action to take, if any, is based on the severity of the risk scores. If management is evaluating how to apply resources to mitigating risk to multiple systems, the threat risk scores play a large role.

Again, the overall risk score for the threat is the same as the Internet attack vector. Other information that can be applied to the attack tree at this point is the cost of eliminating the conditions necessary for an attack to follow a specific path. In our example, the cost for reconfiguration of the firewall is simply a very low opportunity cost. However, ESI's core switch doesn't support VLAN configuration to segregate the database server onto a more security network segment. So the cost of eliminating this condition is much higher.

We already know from the scenario scoring in Figure 5 that removing this vulnerability moves the risk score into the Low Risk category. The desired outcome at the right cost makes this an easy decision for ESI's management. The final attack tree is shown in Figure 7.

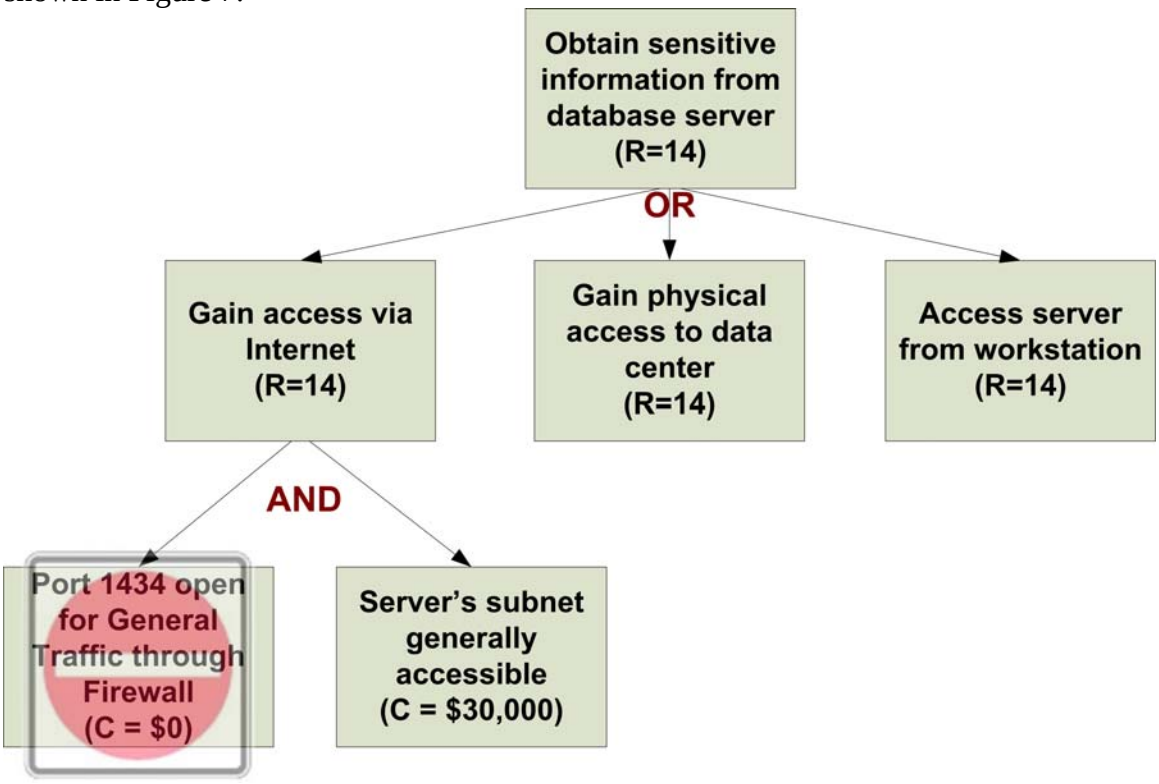

Figure 7: Attack Tree – Final

The final attack tree should be filed with the other business continuity documentation maintained for this system. It helps provide a view of existing vulnerabilities to known threats.

Conclusion

In this paper, I've stepped through a simple, practical approach to threat modeling. When viewed through the risk management lens, this is effectively a qualitative approach. But it allows security analysts to develop documentation necessary to make the right choices when bombarded with scores of recommendations and demands for vulnerability mitigation.

I have one final comment. Don't just accept this as the final word on threat modeling. This is just a high-level methodology that can be a good starting point. Use it to develop your own processes. Expand, bend, or throw out the ideas in this paper as necessary. The objective isn't adherence to some process articulated one Saturday afternoon by a seasoned (translated old) IT professional. Instead, it's the secure operation of your network.

 Tom Olzak, MBA, CISSP, MCSE, is President and CEO of Erudio Security, LLC. He can be reached at tom.olzak@erudiosecurity.com or by visiting http://adventuresinsecurity.com

---

Works Cited

Chidambaram, V. (2004, December). *Threat modeling in enterprise architecture integration.* Retrieved March 2, 2006 from http://www.infosys.com/services/systemintegration/ThreatModelingin.pdf

Ellison, R. J. (2005, September). *Attack trees*. Retrieved March 1, 2006 from https://buildsecurityin.us-cert.gov/portal/article/bestpractices/requirements_engineering/attack-trees.xml

Meier, J. D., Mackman, A., Dunner, M., Vasiereddy, S., Escamilla, R., & Murukan, A. (2003, June). Improving web application security: threats and countermeasures. *MSDN*. Retrieved February 25, 2006 from http://msdn.microsoft.com/library/en-us/dnnetsec/html/THCMCh03.asp?frame=true

MSDN (n. d.). *Evaluating security threats.* Retrieved March 1, 2006 from http://msdn2.microsoft.com/en-us/library(d=robot)/ms172104.aspx