Understanding Risk Through Attack Tree Analysis

Amenaza Technologies Limited
Suite 550, 1000 8th Ave SW
Calgary, Alberta
Canada T2P 3M7

1-888-949-9797

info@amenaza.com
www.amenaza.com

Table of Contents

List of Figures

Understanding Risk Through Attack Tree Analysis

Attack Tree Theory

Introduction

Background

Each and every day we all make decisions that involve the assessment of risk. In fact, it is hard to think of any choice that doesn't involve some risk. Should I have the tuna sandwich for lunch or the cheeseburger? The fish might be more prone to spoilage if not handled correctly, but the burger has a higher fat content that may harm my heart. Price is also a factor in my decision. Most of our risk assessment decisions are informal. Through experience we intuitively come to understand which choices result in more risk than we are willing to accept and take steps to avoid, reduce or share it.

The world has become very complex. Modern technology makes it possible to feed, clothe, shelter and amuse millions of people worldwide. The effectiveness of these tools simultaneously enhances our lives and makes us vulnerable. In 2003, a massive power outage (of uncertain cause) left millions of people in the US and Canada without electricity for days. Not only was the problem not predicted, but weeks after the event, experts were still trying to comprehend it. There is no question that complexity creates situations beyond most people's experience. This makes it difficult for them to gauge risk intuitively.

Historically, we have addressed this challenge through the use of statistics. Although no single individual might have the necessary experience with certain events to properly judge their frequency, society as a whole may. Collecting data on random events makes it possible to use society's collective experience to make rational, educated predictions. Without understanding the technical details associated with an anticipated incident, statistics make it possible to predict the likelihood of an event occurring and take appropriate action. For example, the knowledge that power failures occur every few years suggests to business that critical computer systems should have backup power systems. This type of thinking allowed contingency plans to be implemented that made the long-term damage caused by the 2003 power grid failure surprisingly light, considering the extent of the outage.

Unfortunately, using past events as a basis for risk decisions is not adequate for all of today's situations. This is particularly true when considering the risk associated with deliberate, hostile attacks. The lever of modern technology allows an individual, or small group of individuals to inflict damage disproportionate to their numbers. In 2001, a few dozens of terrorists, with an estimated budget of less than $500,000, used a human guided missile to kill thousands of people and destroy a $40B skyscraper that took thousands of person years to build. When the cost of the US led retaliation is included in the calculation, the terrorists cost their victim over $200B, or 40,000 times their investment!

At the more mundane level, adolescent computer hackers regularly, with a few hours of work and zero cash outlay, create viruses that cost business tens of thousands of hours (and

millions of dollars) of repair work. These examples demonstrate that people of modest means are now capable of causing unprecedented damage. This situation is unique to our era.

Dealing with these new threats is beyond the experience of most people. The resources available for defenses are finite. Although it is possible to implement protective measures against almost anything, it is not possible to protect against everything. No matter what we do, it is possible that we will be criticized for our actions. If something bad happens (and we didn't prepare for it), how do we show that our preparations were not unreasonably (and miserly) optimistic? When the bad things we predicted do not come to pass, how do we demonstrate that we did not squander resources through paranoia? Answering these questions is what risk analysis is all about.

Risk Theory and Definitions

The definitions of risk related terms vary slightly from author to author. Here are the meanings that we will use throughout this document.

Risk – Traditional Definition

Traditionally, the **risk** associated with a particular event can be defined as:

$$Risk_{Incident} \equiv (Probability\ of\ the\ incident) \times (Impact\ caused\ by\ the\ incident)$$

While it is satisfying to write a formula describing risk, in many situations this formula is not useful. Although it is usually straightforward (if tedious) to estimate the potential damage caused by a hypothetical incident, it is not always obvious how to find a value for the *Probability of the incident* term. That term's value (usually expressed as a number between 0 and 1) is a result of many factors, some of which may not be easily quantified.

System

Whenever we consider risk, we have to establish what is included in the scope of our analysis. Philosophers would probably argue that an injurious event ultimately affects everyone in the world, both today and down through eternity. Most other people limit their concern to things for which they have a direct responsibility or that affect them directly. The area of consideration is usually called, the **system**.

Webster's dictionary defines the word **system** as a *regularly interacting or interdependent group of items forming a unified whole*. Although a *system* almost certainly contains a variety of physical components (such as computers, buildings), systems may also include the people that interact with the components and the processes they use to do so. An early step in risk analysis is to decide which components make up the **system** being studied.

Vulnerability

All systems suffer from one or more **vulnerabilities**. A **vulnerability** is a weakness in a system. It is a mechanism by which a system could be damaged, its resources used in an unauthorized way or caused to enter an undesirable state. For instance, a computer system that authenticates users via passwords is vulnerable to password guessing. The classic Greek hero, Achilles, was only vulnerable to injury in his heel.

Threat

A **threat** is a potential source of danger to a system. A **threat** is something that is capable of acting on a specific vulnerability or set of vulnerabilities. The presence of a threat does not guarantee that the threat will act on a vulnerability. Rather, it shows the potential for this to occur. A threat may originate from a hostile, intelligent agent that desires to inflict damage or it may be a product of random conditions in nature. For example, the possibility of metal stakes being driven into trees is considered a threat to safe logging operations. The potential of lightning strikes are a threat to people who enjoy walking outdoors during thunderstorms.

Threat Agent

A class or group that constitutes a threat to a system is known as a **threat agent.** Using the example given earlier, a **threat agent** that might drive metal rods into trees to interfere with logging could be radical environmentalists determined to stop the felling of trees. However, a disgruntled ex-employee hoping to cause financial damage to his former employer would also be a valid **threat agent**. If you are protecting a computer system that contains trade secrets then both industrial spies and adolescent script kiddies are plausible threat agents.

Some authors consider every class of individuals that have the potential to carry out an attack on a system as legitimate threat agents. Strictly speaking, this is true. However, throughout this document, we will restrict our use of the term to groups that have some conceivable reason or desire to harm a system. This means that, by our definition, the US Army is not a plausible threat agent against a New York bank. Although the US Army certainly has the potential to invade the bank and steal the money, we can think of no reason why it would be motivated to do so. To emphasize that our usage of the term includes the concept of motivation, we will often speak of plausible or viable threat agents.

Exploit

Whereas a threat is an abstract way to take advantage of a vulnerability, an **exploit**(n) is the detailed procedure for doing so. The term is frequently used in connection with attacks on computer systems. For example, it may be known that an application suffers from a *buffer overflow* vulnerability. That is, if excessive data is supplied to the program's input, the program may behave in a way that is inconsistent with its design. An **exploit** that makes use of this vulnerability would consist of the exact procedure needed to cause the program to misbehave. It would include the method by which the data would be transmitted, the sequence of characters that would be used and any other details needed to make use of the vulnerability. When used as a verb, **exploit**(v) means the act of carrying out the malicious procedure.

Incidents and Events

Again citing Webster's, an **incident** is *an action likely to lead to grave consequences*. In essence, when a threat ceases to be a merely hypothetical possibility and a vulnerability is acted upon, it becomes an **incident**.

In many cases, an **incident** occurs as a result of a sequence or combination of other, contributing **events**. For example, a car may have an accident as a result of a flat tire. At first glance, the flat tire is the **incident** that caused the accident. However, the flat tire was the result of a puncture by a nail. The nail fell from a passing construction truck. This occurred because a construction worker neglected to put the box of nails in the toolbox. Each of these events is an **incident** that caused a subsequent **incident.** Some of the **incidents** caused a degree of harm immediately while others only led to unpleasant consequences.

We tend to use the term **event** for lower level actions which may, or may not, have immediate consequences. The term **incident** is usually used to describe the event or events higher in the causal chain that are more directly associated with the resulting *grave conditions*. This distinction is largely artificial since almost any event can be decomposed into more detailed

events. **Incident** and **event** are practically synonyms.

Attacks and Mishaps

There are two types of **incidents.** An **incident** that is caused by a conscious, deliberate application of an exploit is called an **attack. Incidents** that result from unintentional or random events are called **mishaps.**

Victim Impact and Attacker Benefit

Incidents and events generally cause damage to the system involved. This is called **victim impact** or, more simply, the **impact.**

When the incident occurs as the result of a deliberate attack, there may also be a positive impact or benefit to the attacker. This is called the **attacker benefit.**

In some cases, the **victim impact** of an attack is equal to the **attacker benefit.** For example, if an attacker steals $1000 then one party loses and the other gains $1000. However, this is the exception rather than the rule. Vandalism illustrates this point. The victim may suffer significant financial damage while the attacker gains nothing of monetary value.

Impact can be measured in many different units. Although money is the most common metric, almost anything that has value can be used. For example, the number of casualties, or a low/medium/high loss of reputation are valid measures of impact. Depending on which vulnerability in a system is targeted, and which exploit is used, the **impact** will vary.

Fault Trees

An incident is the result of other underlying incidents and events. The relationship between the incident and the contributing events needs to be captured in some form. A graphical representation that has proven valuable in the study of industrial system failures is called a **fault tree** (also known as a *failure tree*).

Fault trees[1] are extensively used to model how problems occur in critical systems. For example, chemical plants use fault tree analysis to determine the consequences of a ruptured pipe. Fault tree analysis was used in the investigation[2] of the space shuttle Challenger disaster and again in the loss of the Columbia.

While **fault trees** do not describe all of the concepts that need to be considered in a complete threat-risk model, they are a good foundation to build on. Consider a simple example involving **fault trees.**

Traffic Light Example

A small town is considering converting its traffic light system from manual to computer control. The town is very small and has only one traffic light, located on main street. The mayor is progressive (some would say overreaching) and wants the hamlet to have a centrally controlled traffic light, similar to that used in major centers. He has proposed a system that involves a central computer system which sends instructions via a network link to the lone traffic pole. The Mayor touts cost savings as the justification for his system. He claims that it will no longer be necessary to send a worker to the pole to make adjustments. Since the system can be programmed to automatically change signal durations based on time of day this will improve traffic flow.

The mayor's critics have expressed concerns about the cost of the system. Seeking to appease the naysayers, the mayor has trimmed back the design such that the proposed traffic pole has very little intelligence – it cannot function without communication from the central office. The pole does have a battery backup that will keep the lights functioning for a time if utility power is lost, but almost any other fault will result in total system failure. His opponents now insist that a study be done to examine the reliability of the proposed system.

After some consideration the following three major conditions are identified that may result in a traffic light failure:

1. Power failure.
2. Hardware failure.
3. Software failure.

Additional study reveals that each of these states can result from one or more incidents or

---

[1] Fault Tree Development, 3rd Edition March 2002, Sutton Books, http://www.swbooks.com/#monographs

[2] http://www.hq.nasa.gov/office/codeq/doctree/fthb.pdf

subevents. These are shown in the list below. The list shows the various conditions that will result in a failure of the traffic control system. Increasing levels of detail are shown by lower levels (greater indentation) in the list. Within a given level the condition is preceded by ▲ if it can be reached by satisfying any one of its subconditions or by ❖ if all of its subconditions must be true.

Traffic Light Failure Description

1 ▲ ***Power failure.***

1.1 ▲ Power to central computer system fails.

1.1.1 Failure at power utility (blackout).

1.1.2 Power line severed.

1.1.3 Problem with power infrastructure inside of building.

1.2 ❖ Pole power fails

1.2.1 Traffic pole backup battery fails

1.2.2 ▲ Utility power to traffic pole fails.

1.2.2.1 Failure at power utility (blackout).

1.2.2.2 Power line severed.

2 ▲ ***Hardware failure***.

2.1 Computer components fail at central office.

2.2 Network components fail between central office and intersection.

2.3 Components on traffic pole fail.

3 ▲ ***Software*** failure.

3.1 ▲ Central Traffic Control System is unavailable (crashed).

3.1.1 Operating system failed.

3.1.2 Real-time application failed.

3.2 Central Traffic Control System sending incorrect information to traffic pole microprocessor (up, but confused).

To illustrate, consider the major fault *1 Power Failure,* and its associated subfaults. There are two major ways in which the power failure fault might happen. Either the *1.1 Power to central computer system fails* condition could occur, or item *1.2 Pole power fails* might take place. The *OR* relationship is denoted by the ▲ symbol on item *1 Power Failure*. Focusing, for the moment, on item *1.1 Power to central computer system fails*, we see that this condition could happen in any of three ways:

- Utility power might fail (blackout) (fault *1.1.1*)
- The power line leading to the central office might be cut (fault *1.1.2*)
- Some part of the central office infrastructure (e.g., a circuit breaker) might fail (fault *1.2.3*)

Any one of these conditions would result in the central computer going down which would, in turn, knock out the traffic light.

A power failure at the traffic pole might also cause the system to go down, but due to the batteries located at the pole, both of conditions *1.2.1 Traffic pole backup battery fails* and *1.2.2 Utility Power to traffic pole fails* would need to be true. The *AND* requirement is indicated by the ❖ symbol which precedes the *1.2 Pole power fails* condition statement.

The enumeration of incidents, conditions and events could be continued at ever greater levels of detail. Unfortunately, as the list becomes larger and more detailed it becomes more difficult to comprehend. The same information can be depicted more clearly in a graphical format called a **fault tree**. Each possible incident or event is depicted as a bubble or *node* in a tree structured graph. Unlike the trees that occur in nature, **fault trees** grow downward. Within the **fault tree**, the topmost node is called the *root* node and represents the state or condition that we wish to avoid. Nodes immediately below the *root* node are the states, conditions or events that contribute to reaching the *root*. These nodes are called the *children* of the *root.* That is, the *root* is their *parent*. Nodes which share a common parent are known as *siblings.*

The level of detail continues to increase moving downward through the intermediate nodes in the tree. The *child*ren of each node provide more precise descriptions of the states, conditions or threats leading to the *parent*.

The lowest nodes in the tree are known as *leaf* nodes. *Leaf* nodes are sufficiently detailed so as to require no further decomposition. They represent not only a condition, but an action or event that may lead (possibly in conjunction with other incidents) to the high level incident represented by the root node.

It is convenient to establish a graphical convention for distinguishing the logical relationships between different types of nodes[3]. *AND* nodes represent states that can only be reached if all of the states represented by their children are achieved. Throughout this document we have adopted the convention that *AND* nodes are depicted as 8-sided, blue polygons. *OR* nodes symbolize states that occur if any of the node's children are attained. They are shown as green rectangles with rounded corners. *Leaf* nodes are drawn as grey, sharp cornered rectangles.

Figure 1 is a **fault tree** that uses these symbols to graphically represent the *Traffic Light Failure Description* shown earlier. Notice how the **fault tree** makes it much easier to identify the three high level potential causes of traffic light failure *(Power failure, Hardware failure, Software failure*).

[3] Other authors use slightly different graphical symbols to create *fault trees*. Our representation is chosen to lead into the more sophisticated graphical structures that will appear later.

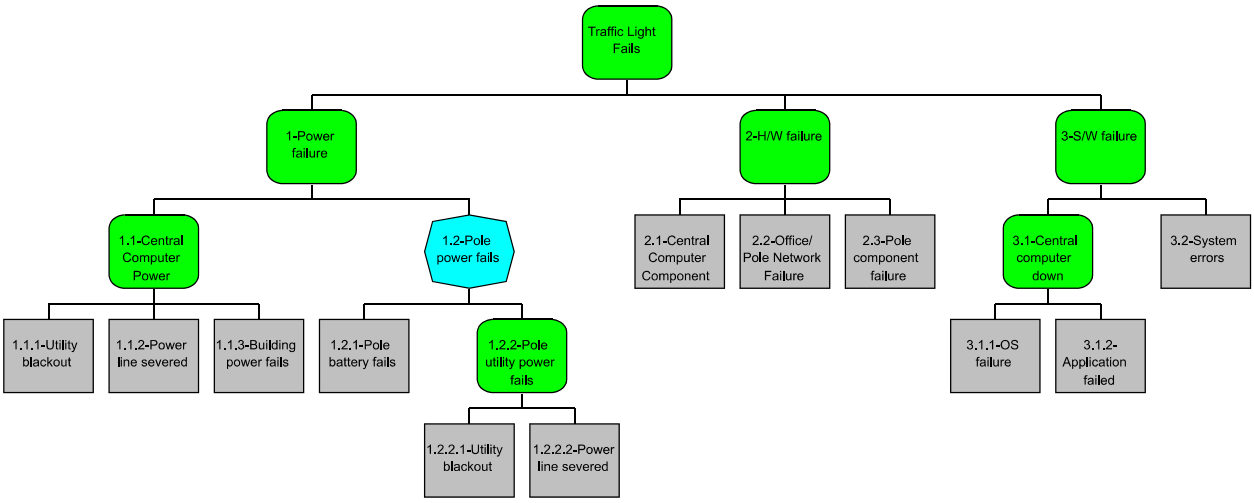

Figure 1 – Traffic Light Fault Tree

This **fault tree** describes **how** an event can occur, but not **whether** it is likely. This is a major limitation of the basic **fault tree**.

Enhanced Fault Trees

The potential ways in which the proposed traffic light system can fail are easily identified through simple inspection of the fault tree. Unfortunately, this knowledge is not enough. The town council needs to know which, if any, of the theoretical defects will have a significant impact on traffic signal operations. Before approving the Mayor's scheme, council wants an indication of the reliability of the overall system. They also need to know which defects will cause the majority of the failures and how they might be prevented. Fortunately, this can be calculated.

Most manufacturers of industrial equipment provide data on the reliability of their gear. The reliability of their components is expressed as a *Mean Time Between Failure (MTBF)* and *Mean Time to Repair (MTTR)*. These values can be used to compute the probability that a subsystem or component will be down[4]. Values for the traffic system are shown in Table 1.

[4] For example, suppose a system is known to have a MTBF of 8760 hours (1 year). When it breaks it has a MTTR of 4 hours. We can estimate that, on average, the system will be down 4 hours out of 8760 hours or 0.0004566 of the time (about 0.045%).

Probability of Failure of Traffic Light Components						
Component	MTBF (hrs)	Failures per year	MTTR (hrs)	Down time (hr/yr)	Downtime ÷ Total Time	% Down time
Utility Power	8760	1.00	1	1	0.00011416	0.0114%
Power Line	17520	0.50	4	2	0.00022831	0.0228%
Building Power	4380	2.00	0.25	0.5	0.00005708	0.0057%
Pole Battery	35040	0.25	4	1	0.00011416	0.0114%
Central Office Hardware	2190	4.00	4	16	0.00182648	0.1826%
Network	1460	6.00	2	12	0.00136986	0.1370%
Pole Hardware	8760	1.00	2	2	0.00022831	0.0228%
Operating System	8760	1.00	1	1	0.00011416	0.0114%
Application Software	4380	2.00	1	2	0.00022831	0.0228%
System Degraded	8760	1.00	4	4	0.00045662	0.0457%

Table 1 – Traffic Light Component Reliability

Knowing the probability that a subsystem is in failure mode allows us to calculate the probability that other components that depend on the subsystem will be affected[5]. **Figure 2** shows the traffic light fault tree with added information about the probability of failure.

The enhanced fault tree acts as a model to help us understand the system under study. As with all models, the fault tree simplifies the situation enough to enhance understanding while still including the characteristics that affect the system's behavior[6].

[5] For *n* independent events, the probability that all will occur simultaneously is the product of the individual event probabilities, $P = a \times b \times \ldots \times n$. This formula can be used to calculate the probability of achieving an *AND* node's state if the probabilities of the *AND* node's children are considered to be the individual probabilities. Similarly, since the probability of one or more independent events occurring is given by, $P = 1 - [(1 - a)(1 - b)...(1 - n)]$, that formula can be used to calculate values for *OR* nodes.

[6] As presented, the model only describes the very simple situation where the traffic light control system consists of a single traffic light on main street. If the town had multiple traffic light poles then nodes *1.2.1 Pole battery fails, 1.2.2.2 Power line severed* and *2.3 Pole component failure* would need to be repeated. If the number of poles were large the model would quickly become unreadable. In this situation it might be better to approximate reality by calculating an aggregate probability value for failures on the set of poles.

The enhanced traffic light fault tree indicates that the traffic light will experience about 40 hours of downtime per year[7]. This is clearly unacceptable. The biggest source of downtime seems to be the hardware subtrees which contributes 0.34% of the total or almost 30 hours. Roughly ⅔ of the outages are induced by failures at the central office. The remaining ⅓ of the failures are mostly due to network issues.

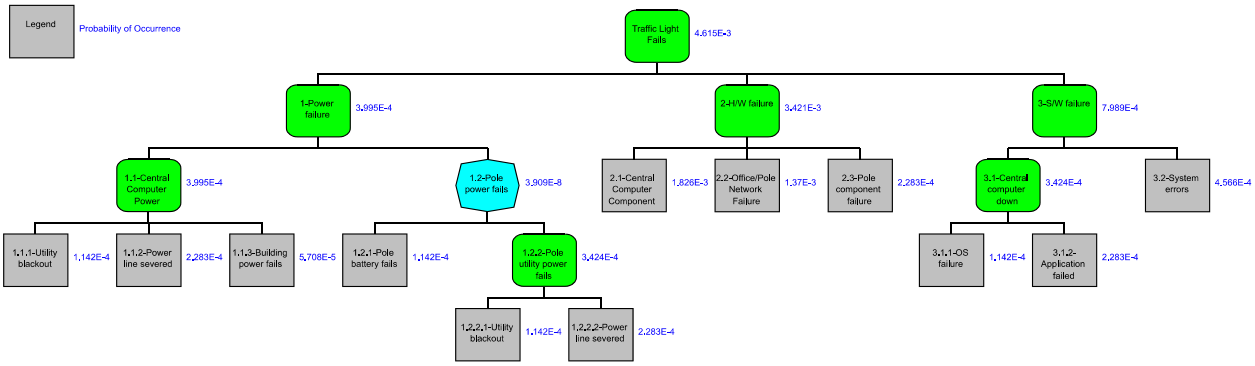

Figure 2 – Traffic Light Probability of Failure

The system can be made more reliable by reducing the number of outages at the central office and by improving the reliability of the network. Alternately, deluxe hardware can be installed at the pole that will allow it to operate autonomously without instructions from the central office. Since there is only one traffic pole, it is decided to improve the hardware at the pole. If the town had many poles, a different decision might have been made. The system could then be remodeled to see what effect the changes might have.

In this example the potential causes of system outages were all non-malicious, random component failures. Because we were able to obtain reliability statistics for the components, the enhanced fault tree could be used to calculate the expected reliability of the entire system. This made it clear which portions of the system would be problematic and provided clues as to how the system might be improved.

The enhanced fault tree provides the likelihood of a failure in the traffic signal system. Failures presumably lead to accidents, congestion (causing loss of productivity) and overtime for the police to direct traffic in the intersection. With some thought and research it would be possible to estimate the impact of a traffic signal failure. The risk equation

$$\boldsymbol{Risk_{Incident} \equiv (Likelihood\ of\ the\ incident)\ x\ (Impact\ caused\ by\ the\ incident)}$$

can then be used to calculate an expected loss in a given period of time (typically yearly). This value is called the **annual loss expectancy (ALE).** Generally, if the cost of improving the system (and operating it) is lower than the ALE, standard risk management practices recommend proceeding with the upgrades.

Fault trees make it easy to model the risk of systems that have good statistical information

[7] The traffic light being analyzed will be down 0.46% of the time. 0.0046 x 8760 ≈ 40 hours/year.

for subcomponents. While this is certainly valuable, the requirement for statistics greatly limits the applicability of fault trees. Moreover, the very fact that statistics are available may mean that we have sufficient experience in the area to depend on intuition – allowing us to dispense with formal analysis entirely.

Unfortunately, many incidents are induced by events that are not random or happenstance. These events are driven by hostile attackers with malicious intent. For situations where the attacks occur frequently, statistics may exist. For example, the automobile theft rate is well known. All too often, however, statistics are unavailable. In these cases, other techniques for estimating probability have been devised. For example, checklists are sometimes used to identify factors that increase the probability of an incident. Computer security experts often use checklists that assign a certain number of points to configurations that include risk-increasing components such as modems, Internet connections, poor physical security. The points are then entered into some empirically derived formula[8] to provide an estimate of how likely it is the system will experience an incident. It is our belief that these techniques generally yield poor results.

The next section further enhances the tree model so that it is capable of evaluating risk even when statistics do not exist.

[8] Since the formula is almost certainly based on experience, it is fair to say that the checklist approach is essentially an informal kind of statistics.

Capabilities-based Attack Trees

In 1994 Amoroso[9] discussed a concept called *threat trees.* More recently, Bruce Schneier[10] (a noted cryptographer) refined and popularized the idea through what he called *Attack Tree Modeling.* Other researchers have continued to develop the idea of tree-based, threat analysis models[11].

Amenaza Technologies Limited has taken inspiration from all of these tree-based security models and enhanced them by creating procedures and software. Amenaza refers to its approach as *capabilities-based attack[12] tree analysis.* The associated software is known as Secur*I*Tree®. Capabilities-based attack tree analysis has been especially well received by the information technology security community. In response to this interest, Amenaza created a library of predefined Secur*I*Tree® models for analyzing systems built around popular computer technologies. However, **attack tree analysis works for almost any type of system.**

Earlier, we pointed out that statistics frequently do not exist for threats originating from an intelligent, malicious adversary. While this is true, it is still possible to identify factors which affect an attacker's behavior. These factors influence whether and how a threat agent will attack. **Even a highly motivated threat agent can only carry out a given attack if the resources at their disposal meet or exceed the resources required to perform the exploit.** The scarcity of resources forms a constraint on the threat agent's behavior. Resources include money, technical skill, time and a willingness to pay the price for one's actions. Capabilities-based attack tree analysis uses these constraints to determine the likelihood of attacks.

---

[9] Edward G. Amoroso, Fundamentals of Computer Security Technology, Prentice-Hall, ISBN0131089293

[10] B. Schneier, Attack Trees, Dr. Dobb's Journal, v. 24, n. 12, December 1999, pp. 21-29.

B. Schneier, Attack Trees: Modeling Actual Threats, SANS Network Security 99 – The Fifth Annual Conference on UNIX and NT Network Security, New Orleans, Louisiana. Wednesday, October 6th, 1999, Session Two, Track One - Invited Talks

B. Schneier, Seminar session given at a Computer Security Institute conference in November, 1997. See also http://www.counterpane.com/attacktrees.pdf

[11] Moore, A., Ellison, R. and R. Linger, "Attack Modeling for Information Security and Survivability", March 2001, http://www.cert.org/archive/pdf/01tn001.pdf

[12] We debated whether **threat tree** or **attack tree** was the correct term. According to the definitions given earlier, a **threat tree** model would encompass both the malicious, intentional attacks associated with intelligent adversaries and threats originating from random, environmental conditions. The analysis technique we are about to describe is definitely capable of modeling both hostile and environmental threats, so **threat tree** is a perfect name. On the other hand, an **attack tree** would (by our definitions) restrict itself to incidents that were caused deliberately. Notwithstanding the apparent violation of our definitions we have chosen to use the term **attack tree.** There are two reasons for doing so.

First, B. Schneier has popularized the term **attack tree** through his well known publications on the subject. We felt it would introduce confusion to use another term, even if it was more correct. Second, while the **attack tree** models we will study are quite capable of modeling random mishaps, the focus is strongly on understanding deliberate attacks. Hence, our use of the term **attack tree** instead of **threat tree.**

A Sample Attack Tree

To illustrate the concept of a **capabilities-based attack tree**, we will again imagine a hypothetical situation. This time, we will consider the home security challenge faced by the residents of a typical, suburban home. While few middle-class home owners would do a formal security risk assessment on their home, the subject was chosen as being one to which all readers could relate.

The house we have in mind is a middle-class dwelling, complete with attached garage. The incident that concerns us is the possibility of the house being burglarized. This is depicted in the topmost node of the tree, labeled *Burgle House*. After some consideration, we can think of seven possible ways in which a thief might enter the house to commit burglary:

1. Passage doors (i.e., the front and back doors normally used for entry).
2. Windows.
3. Through the attached garage.
4. Walls (including the roof – it is essentially an angled wall).
5. Chimney.
6. Floor (attacking from beneath).
7. Social engineering (convince the resident to allow entry).

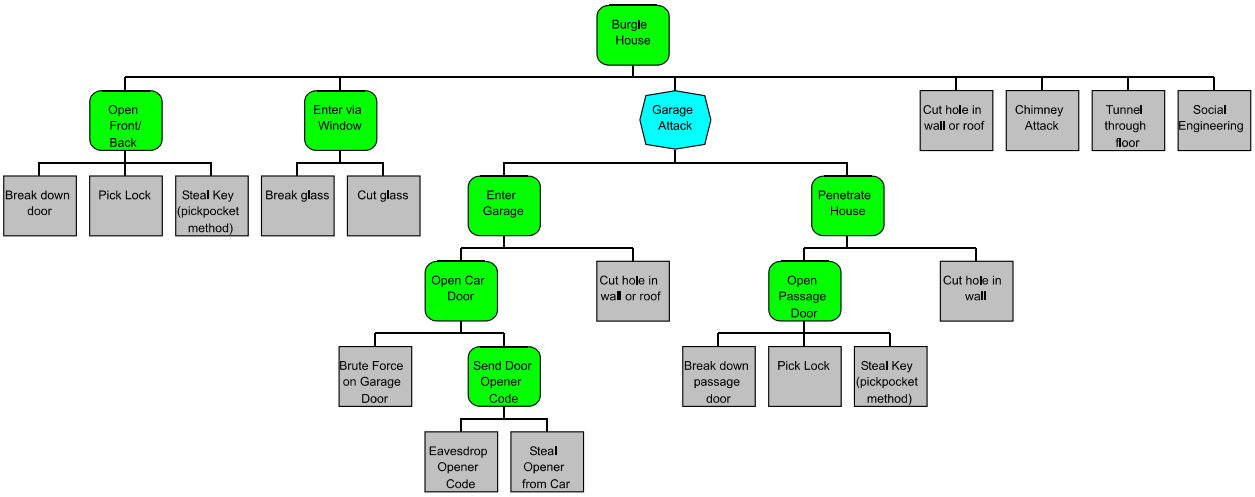

Figure 3 – Attack Tree Showing Ways to Burglarize House

These attacks, which have been partially decomposed into more detailed steps, are shown in **Figure 3**. To simplify our example, we have restricted the decomposition to the *Open Front/Back Door*, *Enter via Window* and *Garage* attack vectors. Obviously, greater detail could also be added to the *Cut Hole in Wall or Roof, Chimney Attack, Tunnel through Floor* and *Social Engineering* attacks.

As can be seen in the diagram, there are three types of passage door attacks. The doors can be physically broken, the locks can be picked or the key can be obtained through theft. Similarly, an intruder can either cut or break the glass in the windows. To enter via the garage,

the burglar must first gain entry to the garage and then enter the house (either through the wall or by penetrating the passage door leading from the garage to the house).

Decomposition of events into smaller, more precisely defined events could continue almost indefinitely. For our purposes, it is only necessary to continue to the point where further decomposition will not increase the understanding of the intended viewers of the model. For example, the *Break glass* leaf node could be decomposed into the steps of picking up a rock and throwing it at the window. This is unnecessary since almost everyone knows how to break a window. Moreover, it doesn't really matter much whether the burglar throws a rock or swings a stick. On the other hand, the leaf node that deals with *Eavesdrop opener code* could, and probably should, be decomposed into smaller steps. This would enhance the analysts understanding of the actions that would be required of the burglar. We have not described this attack to that level for purposes of brevity.

To this point, our attack tree looks very much like the basic fault tree studied earlier. We will now build on the fault tree structure by introducing *behavioral impact indicators*.

Behavioral Impact Indicators

In attack tree models, incidents begin with the events represented by the leaf nodes. If a suitable combination of leaf node events occur (as determined by the *AND/OR* structure of the tree), then the intermediate states or events represented by the parents of the leaf nodes will also be achieved. The event(s) will cascade upward, satisfying higher nodes, and ultimately resulting in the root node being achieved (meaning that the incident has occurred).

Whether or not the leaf node events actually occur is dependent on the presence of a threat that is both willing and able to bring sufficient resources to bear to overwhelm the system's defenses. For non-intentional incidents (such as the traffic light example discussed earlier), suitable environmental conditions must exist to cause one or more leaf node events to occur. The root causes of natural events are rarely understood[13] completely, but their frequency is given by statistics. For example, floods may be known to occur in a particular area once every *x* years. The Mean Time Between Failure (MTBF) and Mean Time To Repair (MTTR) figures in the traffic light example are another statistical example. Typographical mistakes[14] are an example of a non-malicious human threat that can be described statistically. So, *probability of occurrence* is a metric of how an environmental threat agent will behave[15]. This is the simplest example of a

---

[13] Chaos theorists may speculate whether a hurricane in the Gulf of Mexico might have begun with the flapping of an African butterfly's wings. Realistically, the knowledge of all contributing events cannot be determined. This means the event represented by the leaf node cannot be decomposed further but is instead represented by a statistic.

[14] Typos may sound innocuous, but a one character error caused an early space probe to miss Mars!

[15] The term *threat agent* is a bit strained in the case of environmental threats. Who, for example, is responsible for a flood? As will be seen shortly, this terminology fits better for hostile, intelligent adversaries. If it helps, consider the threat agent for environmental threats to be *Mother Nature*, or perhaps *Murphy*, oft referred to in *Murphy's Law*.

behavioral indicator.

In the case of hostile attacks, the threat agent must expend sufficient resources to carry out the appropriate leaf node exploits. Malicious threat agent behavior is strongly influenced by factors such as the *cost of performing the attack, technical expertise required, availability of special materials* and the *probability of getting apprehended and punished*. Since these factors affect the threat agent's behavior, they are also called **behavioral indicators.**

Whether the threats are deliberate or unintentional, the **behavioral indicators** can be used to determine whether an incident will occur.

Behavioral Indicator Functions

In order to determine whether or not a particular goal in the tree model can be reached, we must examine whether a particular type of threat agent has the resources required to reach that state[16]. **By incorporating these resource requirements into the threat model, accurate predictions about the system become possible.**

Fortunately, it is not necessary for the analyst to explicitly specify the resource requirements values for every node in the tree. As with the previous traffic light example, it is possible to automatically calculate the resources required to reach a node from the resource values associated with its children. This works for all but the bottommost (leaf) nodes in the tree (which have no children). Leaf node values must be entered explicitly by the analyst.

Figure 4 and **Figure 5** are two simple attack trees that model the costs that a threat agent would expend in order to achieve various goals. In the case of **Figure 4**, the parent of goals #1 and #2 is an AND node, meaning that both children must be accomplished if the parent is to be fulfilled. It makes sense that, if the attacker must spend $500 accomplishing Goal #1 plus $200 to perform Goal #2 then it will cost the sum of $500 + $200 = $700 to reach the AND node's

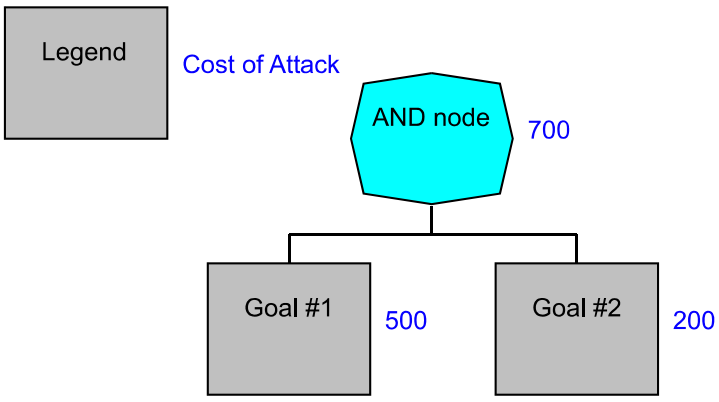

Figure 4 – *AND* Cost of Attack

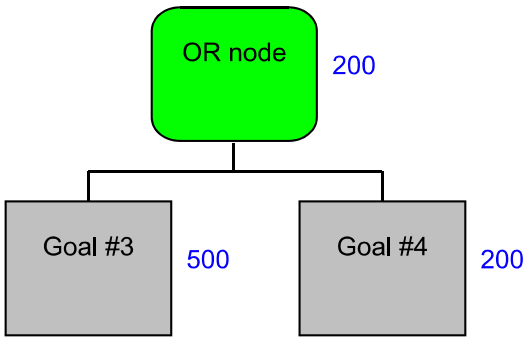

Figure 5 – *OR* Cost of Attack

state. The function associated with the *Cost of Attack* behavioral indicator for AND nodes is

[16] As discussed earlier, implicit in our definition of *threat agents* is the requirement that they are motivated to harm the system.

therefore the *sum of vertices.* We call this the *Cost of Attack Behavioral Indicator AND Function* or simply, the *Cost of Attack AND Function.*

Similarly, in **Figure 5** the OR node can be reached if either Goal #3 or Goal #4 is attained. If we assume that the attacker will choose the cheapest of the two goals then the cost of arriving at the OR node will be the *minimum of vertices.* We call this the *Cost of Attack Behavioral Indicator OR Function,* or simply, the *Cost of Attack OR Function.* It is worth remembering that there are two separate paths by which the attacker can achieve the OR goal – the $200 value shown by the OR node corresponds to the lowest cost path – which may not be the path chosen by the attacker if they have the resources to use the more expensive route.

Indicators are assigned to the tree by the analyst as seems relevant to the problem at hand. Typically three or four indicators are used. Too few indicators leads to a flat, one-dimensional understanding of the forces that drive incidents. An excessive number of indicators may lead to such complexity that the forest is lost in the trees[17].

Ideally, indicators should be **orthogonal.** This means that the behavioral influence of one indicator is independent of another. For example, an attacker's bank balance is largely unrelated to their willingness to be apprehended in an attack. Therefore, *Cost of Attack* and *Probability of Apprehension* are orthogonal indicators. Sometimes, however, there are unavoidable dependencies between indicators. For instance, in many cases it is possible to use money to buy technical skill. Thus, *Cost of Attack* and *Technical Skill* are not completely orthogonal. Complete independence is not always achievable. Indicators should be chosen that minimize dependencies as far as possible.

When the focus is on hostile attackers the behavioral indicators should represent factors that will influence the behavior of the broadest possible spectrum of adversaries. Most classes of attackers are influenced by similar things: cost, difficulty, danger of being caught, etc.

The indicators used for environmental threats are usually probability-based (as in the traffic light example). However, it is possible to think of other things that would also reflect environmental situations. For instance, an indicator might be *Height of Flood Water*. A particular node's *Height of Flood Water* indicator value could reflect the level required to overflow a dike and cause damage.

Indicators may be chosen differently for each project or standardized across the organization. There are pros and cons to each approach. Using a standard set of indicators makes it easier to integrate or compare different projects. A customized set of indicators may better reflect a project specific situation. Perhaps the best idea is to use a standard set of indicators and augment these if they do not seem to influence the threat agents.

Choosing the appropriate indicator functions requires some understanding of mathematics and statistics. Fortunately, the popular indicators have well-known functions.

---

[17] Surely you realized that, somewhere in this document, that pun would be used?

Behavioral Indicators and Recommended Functions		
Indicator	**AND**	**OR**
Cost of Attack	a + b + c + ... + n	Minimum (a, b, c, ..., n)
Probability of Apprehension	1-[(1-a) + (1-b) + (1-c) + ... + (1-n)]	Minimum (a, b, c, ..., n)
Probability of Success	a × b × c × ... × n	1-[(1-a) + (1-b) + (1-c) + ... + (1-n)]
Technical Difficulty	Maximum (a, b, c, ..., n)	Minimum (a, b, c, ..., n)

After selecting the appropriate indicators the analyst must define behavioral indicator functions for each indicator and supply the resource requirement values for the leaf nodes[18]. The leaf node indicator values are obtained through *expert opinion* based on a sound understanding of the activities associated with exploiting the vulnerability in question. Then, the resource values for all non-leaf nodes in the tree can be calculated using the selected formulas.

A problem arises when a model attempts to incorporate both environmental and malicious threats. It has already been noted that the probability figures commonly used for accidental threats are not available for deliberate attacks. Similarly, what is the monetary cost to Mother Nature to launch a flood? There are no perfect answers to these questions but some strategies for dealing with the situation will be shown later (on page 29).

Paths Used to Calculate Node Indicator Values

It is important to understand the significance of the calculated values assigned to the intermediate nodes and the root node. Although it ultimately depends on which indicator functions are used in the calculations, typically **the value at a given node represents the resource requirements needed to reach that point through the lowest cost path or paths.** When multiple different indicators are used within a particular tree, each node will have one value for each indicator. **The calculations for a particular indicator have no bearing on the calculations for another indicator.** Each indicator value represents a specific path from a leaf node or nodes to that point in the tree. The lowest cost path or paths to a specific node are, in general, different for each indicator.

Paths of Influential Nodes

The indicator values at a particular node in the tree are calculated from the values of nodes below. The nodes below have varying degrees of influence depending on the AND/OR structure of the tree, the indicator functions and the node values. It would be convenient to identify the descendent nodes that are most influential on a parent node's indicator values.

The problem is that "influential" is a somewhat subjective term. Is a node influential if it

[18] Although there is no reason why leaf values cannot be added at the time of node definition, most people find it easier to build the tree first and revisit the leaf nodes to add this information.

affects the parent's value somewhat? By 50%? By 10%? There is no pure, mathematically correct answer to this question. It depends on what the person asking the question is trying to achieve.

Critical Path

To illustrate, we define a path that we will call *critical.* The *critical path* will contain all nodes that have some effect on the values calculated above. More precisely, a node is said to be on the *critical path* if its deletion (and the deletion of all sibling nodes with the same value) would change the parent's value or cause it to become undefined for that indicator function. There is nothing sacrosanct about our definition – many others are possible.

Computing and viewing the *critical path* in a tree is helpful to see which leaf nodes provide the lowest cost exposures for each specific resource. It is particularly significant when the critical paths of multiple resource types overlap. **This hints strongly that the vulnerability or vulnerabilities in the overlapping critical paths are a weak point in the system.**

In general, it is quite difficult to interpret the meaning of the critical path. Other analysis mechanisms will be illustrated shortly that are easier to use.

Tree Pruning – Capabilities-based Analysis

Earlier, an attack tree model was created showing the various ways in which home security could be compromised and the resources that would be required of any attacker wishing to enter the house. Although interesting, this is not sufficient to predict who will attack and how. *Capabilities-based analysis* allows us to use the attack tree model to gain exactly these insights and predict attacker behavior!

Capabilities-based analysis of attack trees is based on a very simple premise about attackers' behavior:

IF *they want to* AND *they have the capability to do so* THEN *they will*

In other words, if there exist adversaries that have the motivation to harm the system, the resources needed to carry out the exploits and a willingness to accept the consequences[19] of their actions, then, sooner or later, they will carry out a successful attack on the system.

There are very few situations in which this is not true. The adversaries might not attack if they are unaware that the system exists[20]. They might not attack immediately if there are so many other systems with similar defects that they simply haven't gotten around[21] to hitting this system.

---

[19] Our definition of *capability* includes the attacker's tolerance for embarrassment, financial loss, personal harm or even death. We are unable to think of a more appropriate term.

[20] If the secrecy surrounding the system is a significant part of the system's defenses then the ways in which the system could be discovered should be modeled in the attack tree.

[21] More formally, we might state that, attacking the other targets either brings greater benefits to the attacker or their resource costs for attacking the alternate target are lower.

However, unless all of the capable adversaries are caught before they can get to you, your number will come up.

System Vulnerabilities

The traditional definition of risk (see page 3) includes a term for incident probability. Statistics are derived by observing events that result due to the aggregation of the underlying factors that determine whether or not the events will occur. If those factors only rarely combine to produce an event, then it is impossible to gather statistics. Even if incidents occur frequently, the aggregation inherent in statistics still causes a loss of information about the underlying events. **This impedes understanding of ways to reduce the likelihood of an incident occurring.**

Instead of waiting for incidents to occur so that we can gather statistics, what if we look at the fundamental drivers that generate incidents? Assuming for the moment the adversaries want to attack the system, what determines whether or not they can? It is generally accepted that

$$\text{Probability of Incident} = \text{Threat} \times \text{Vulnerability}$$

The attack tree model we created earlier has been updated (see **Figure 6**) to show the resources required to perform the leaf node exploits and to subsequently reach higher states in the tree. The effort required to carry out an exploit is a measure of the magnitude of a particular vulnerability in the system. So, the tree shown in **Figure 6** represents the *Vulnerability* term of the equation above. If we can determine the magnitude of the *Threat* faced by the system, we will complete the equation.

Threat Agent Capabilities

Earlier we stated that threats originate from *threat agents*. A *threat agent* is a group of adversaries that shares common characteristics and that is motivated to damage a system[22]. Hostile threat agents attempt to create the conditions that will exploit a vulnerability. *Threat* might therefore be written as

$$\text{Threat} = \text{Capability} \times \text{Motivation}$$

In our definition of *threat agent* we assume that they are motivated. Therefore, the level of threat is dependent solely on the attacker's capability[23]. That is,

$$\text{Threat} \propto \text{Capability}$$

Capability is a measure of the resources available to the *threat agent*. Resources include money, skill and knowledge, time and a willingness to suffer adverse consequences.

---

[22] By stretching our definition to include Mother Nature this definition also covers environmental threats.

[23] Later we will show how the motivation level of the attacker can be incorporated into the tree model. This will make use of the fact that: *Threat Agent's Motivation ≈ Possible Benefits for the Threat Agent from the Attack.*

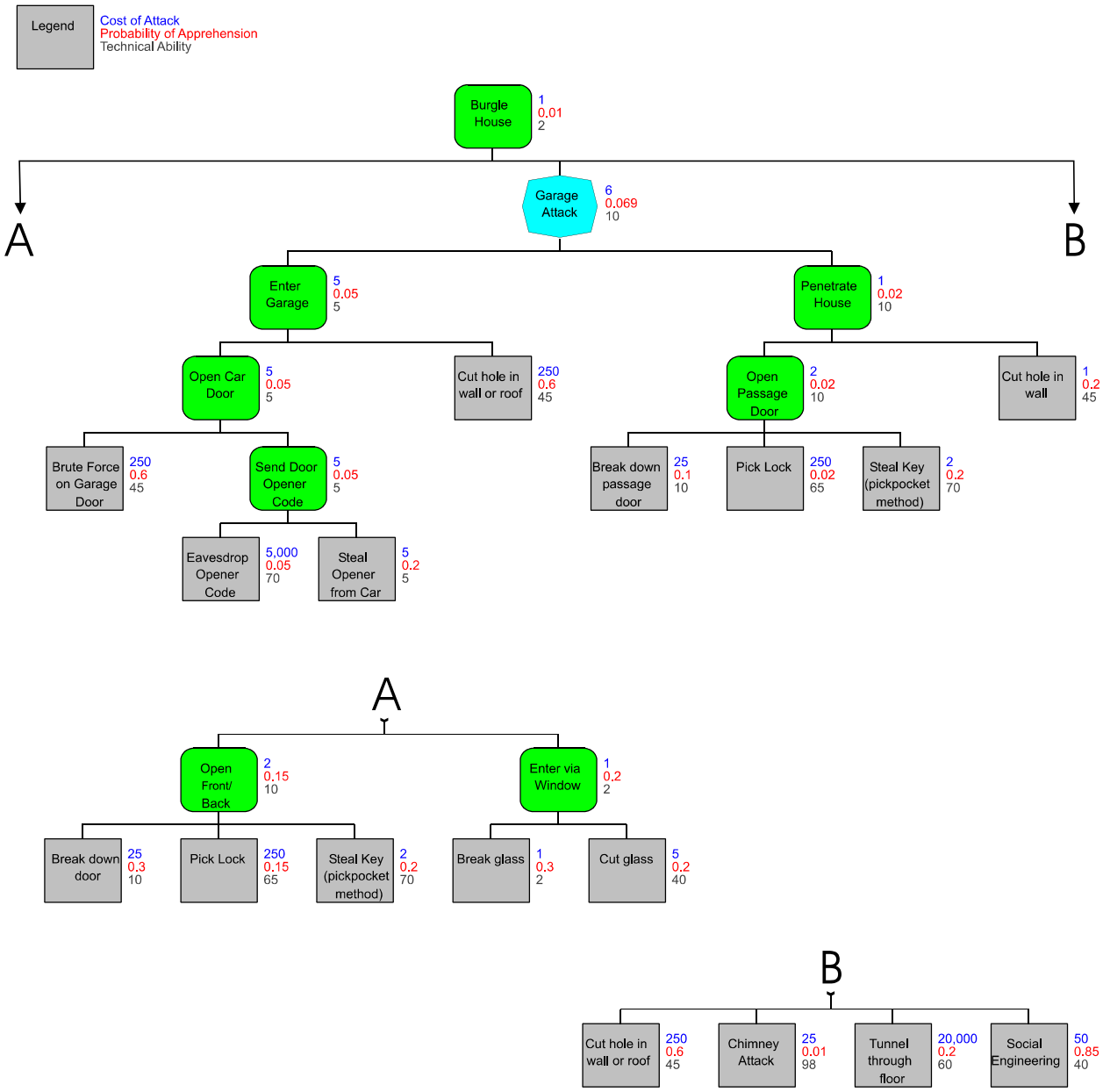

Figure 6 – House Burglary Attack Tree with Exploit Resource Requirements

Identifying threat agents that are motivated to harm a system requires human insight. Most of the time it is a fairly obvious process. For example, banks worry about bank robbers, embezzlers and hackers trying to commit computer fraud. They may have to worry about jewel thieves, depending on the contents of customers' safety deposit boxes. They normally don't worry about an army attacking them. At the same time it is worth considering a few implausible adversaries that may have extraordinary resources just to see how a system will perform when faced with an extraordinary enemy.

The analyst should create a profile for each type of threat agent under consideration. This

profile describes the threat agent's resources corresponding to each indicator in the attack tree model. The next step in analysis will use this information to generate predictions for the behavior of each type of threat agent included in the study. **Failure to include a threat agent with motivation to harm the system means that risks from that agent will not be considered.** It is better to consider too broad a range of threat agents than too narrow.

For example, consider two plausible threat agents for our home security model.

Threat Agent	**Budget**	**Acceptable Probability of Apprehension**	**Technical Capability** 1 - 100 scale
Juvenile Delinquent	$50	50%	25
Cat Burglar	$5,000	10%	70

The Juvenile Delinquent is an angry youth who doesn't have much money to spend on burglarizing houses. He or she is not worried about getting caught. A misspent youth has prevented our miscreant from developing technical skills.

The Cat Burglar, on the other hand, is a pro. Our felonious feline filcher views burglary as a type of employment. He is willing to spend money to make money. He is prepared to spend up to $5,000 on tools. Like any professional he has studied his subject well and is quite capable of picking locks, deactivating simple alarms and jimmying windows. The one thing he is not willing to do is go to jail.

These profiles are assumptions based on the information available to us as well as expert opinion. The accuracy of our predictions depends on the correctness of our assumptions about the threat agents.

Pruning (Eliminating) Non-achievable Goals

The attack tree in **Figure 6** describes the resources required to perform leaf level exploits and, by extension, to achieve goals and intermediate states within the tree. The threat agent profile describes the resources available to a potential attacker. By comparing the two it is possible to prune[24] or delete from the tree all the nodes which are beyond a threat agent's ability to achieve. For example, the possible attacks against the home security system by a Juvenile Delinquent are shown in **Figure 7**. Compare these attacks to those shown for the Cat Burglar (**Figure 8**). The behavior of the two types of attackers will be quite different. In both cases, the available attacks are what we would intuitively expect[25]. Juvenile delinquents tend to smash things a lot. Cat burglars are stealthy and subtle.

[24] Do not confuse a pruned tree with a prune tree. A pruned tree is a threat tree that has had nodes eliminated through the application of filtering criteria. A prune tree is a plum tree after a drought.

[25] As initially stated, in situations where we have significant experience, our intuition is amazingly accurate.

Given that the model of the system is accurate and the assumptions about the threat agents are valid, a pruned tree describes those attacks that are possible. If the threat agents have been selected as being motivated to attack the system, it stands to reason that the attacks remaining after pruning are probable. There isn't enough information available to quantify the probability,

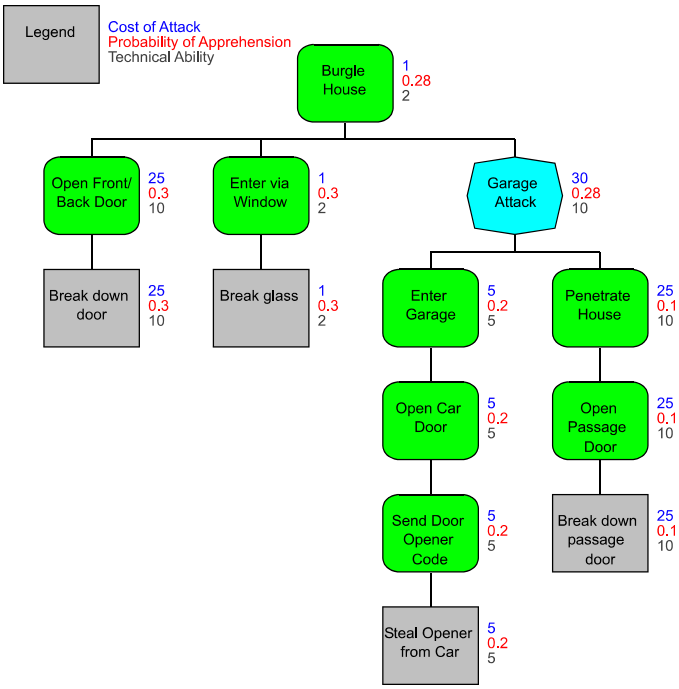

Figure 7 – House Burglary Tree Pruned on Constraints of Juvenile Delinquent

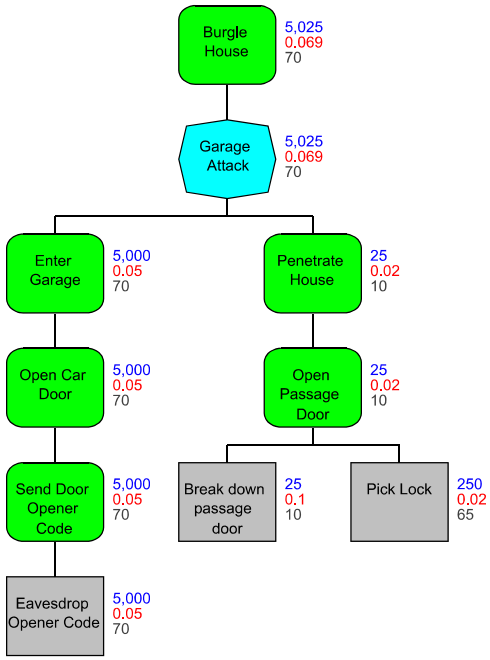

Figure 8 – House Burglary Tree Pruned on Constraints of Cat Burglar

but it is significant.

A major limitation of the pruned tree is that, like the original tree, the indicator values for a given node may reflect a series of different paths that reach that point. The pruned tree shows the set of nodes attainable by the threat agent, but does not show the specific paths they can use to achieve their goal.

Attack Scenarios

An *attack scenario* is a particular path, or set of paths, through an attack tree that leads from a minimal set of one or more leaf nodes to the root. It is minimal in the sense that, if any of the leaf events are removed from the path, then the root cannot be achieved.

Attack scenarios can be found for an entire attack tree. The complete set of *attack scenarios* for an attack tree shows all of the attacks that are available to an attacker who possesses infinite resources and capabilities.

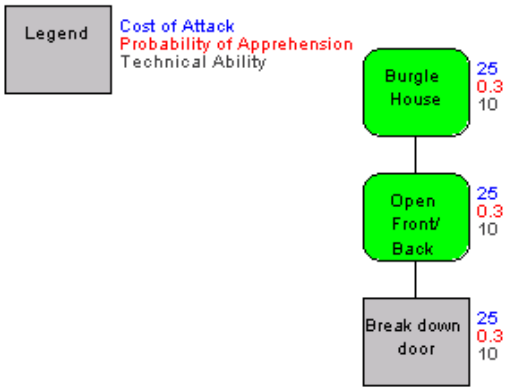

Figure 9 – Juvenile Delinquent Attack #1

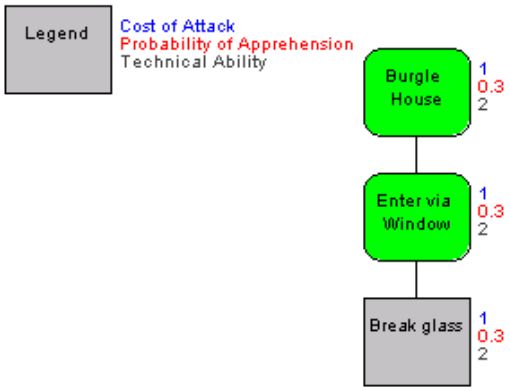

Figure 10 – Juvenile Delinquent Attack #2

It is generally more useful to compute the set of *attack scenarios* for a pruned tree, as this shows which attacks are possible by a particular type of threat agent. For example, the Juvenile Delinquent can choose three different house burglary attack scenarios from the pruned tree (**Figure 7**).

1. Break down a passage (front or back) door (**Figure 9**).
2. Break the glass in a window (**Figure 10**).
3. Steal the garage door opener from the home owner's car (to gain access to the garage) AND break through the passage door leading from the garage to the house (**Figure 11**).

Because an attack scenario shows only a minimal path, this means that the **values computed for the nodes of the tree are the specific resources the attacker must expend along the way.**

The leaf-level events associated with a specific attack scenario can also be used to detect when a particular attack is being

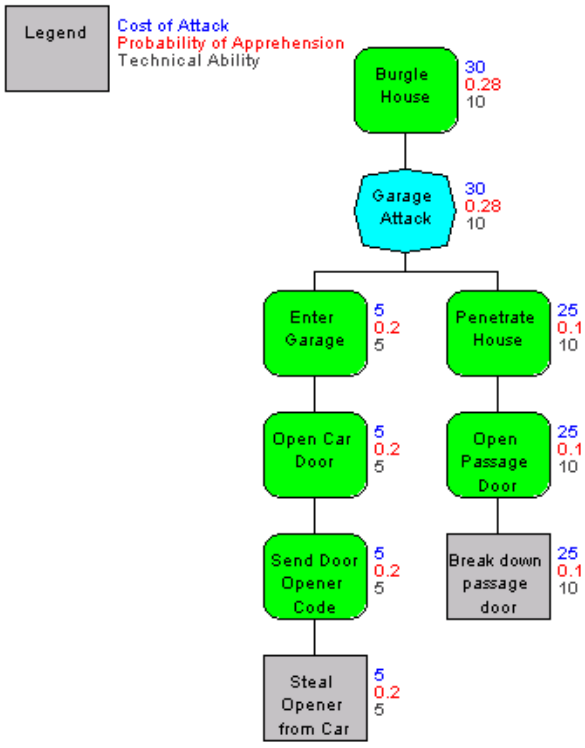

Figure 11 – Juvenile Delinquent Attack #3

employed. Although this may seem somewhat obvious in this example, this can be quite valuable when the situation is complex – with hundreds or thousands of attack scenarios. **It would be quite possible to build an attack detection system to compare leaf level events with attack scenarios and sound an alarm when an attack is underway.**

Determining Risk through Capabilities-based Analysis

Earlier, we defined risk to be,

$$Risk_{Incident} = (Probability\ of\ the\ incident)\ x\ (Impact\ caused\ by\ the\ incident)$$

We then showed that if,

$$Probability\ of\ Incident = Threat \times Vulnerability$$

and

$$Threat = Capability \times Motivation$$

which for a motivated attacker reduces to

$$Threat = Capability$$

then

$$Probability\ of\ Incident = Threat \times Vulnerability$$

Therefore,

$$Risk_{Incident} = (Threat \times Vulnerability)\ x\ (Impact\ caused\ by\ the\ incident)$$

Since a pruned attack tree models the *Threat* ×*Vulnerability* term, if we can incorporate *impact* into our model we will be able to determine risk.

Impact Indicators

Just as we earlier created *behavioral indicators* to model factors that influence attackers' behavior, we will now introduce the concept of *impact indicators. Impact indicators* are primarily used to model the effect or impact that an attack will have on the victim. However, they can also be used to represent the benefits that an attacker derives from attacks.

The *behavioral indicator* values are computed from input at the leaf level. Once the analyst has selected the correct formulas and the filled in the starting values, everything is automatic from that point.

Impact indicators require greater analyst intervention. While some values can be calculated, many more can only be determined by examining the business processes affected by specific attacks. Even though all successful attacks reach the same root goal, different paths through the tree (attack scenarios) have varying levels of impact on the victim.

For instance, suppose an attack tree is created showing the various ways of disrupting the operation of a computer system. One attack scenario might involve compromising an Internet firewall, gaining access to the local network and sending malformed network packets at the computer in question until it locks and requires a reboot. Another approach would be to park a truck filled with diesel fuel and fertilizer next to the building and detonate it, leveling the building and incinerating the server. Both attacks achieve the goal of crashing the computer but one is many times more damaging than the other.

This also applies when modeling benefits to an attacker. It is well known in the intelligence community that it is much more valuable for an adversary to steal a secret without being discovered than to perform a noticeable attack. Attack scenarios involving stealthy attacks therefore have a higher benefit to the attacker. This may influence the threat agent's choice of attack.

In order to model impact accurately, the model must allow the analyst to choose whether to input external business information to influence a node's impact indicator values or whether they can be calculated from entirely from the node's children's values. Impact values can be in a variety of different units. The most common metric is money, but they can represent other things such as the number of casualties suffered.

Consider the now familiar home security model (**Figure 3** on page14). The risk analyst estimates that, if an attacker gets into the house, they will steal or damage $15,000 worth of goods. In this particular home, the owner has a lot of valuable tools and sporting goods in the garage. The analyst believes that the attacker will overlook these goods if they break directly into the house. However, if the attack takes the burglar through the garage, the burglar will surely notice the valuables and steal them. In addition, the model must reflect any collateral damage that occurs in the attack. This includes broken windows, damaged doors, stolen door openers, etc. To achieve this detail in modeling, the analyst selectively chooses whether to use a summing formula to tally the damage along a path or to "inject" values into the nodes.

This is most easily demonstrated by examining the attack scenarios of the home security tree which has been pruned to show the attacks possible by a juvenile delinquent. These

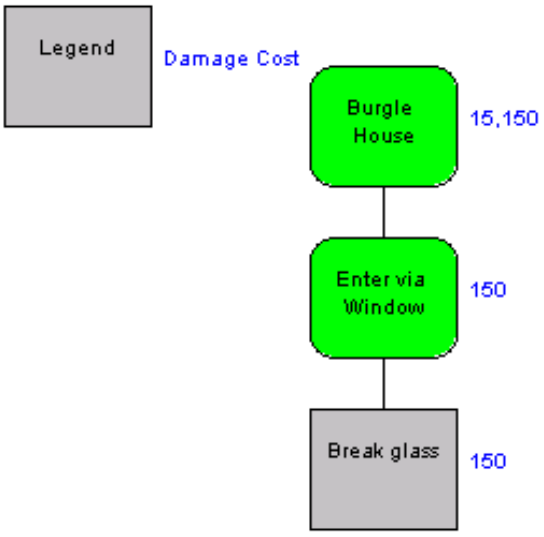

Figure 12 – Juvenile Delinquent Monetary Impact on Victim - Attack #1

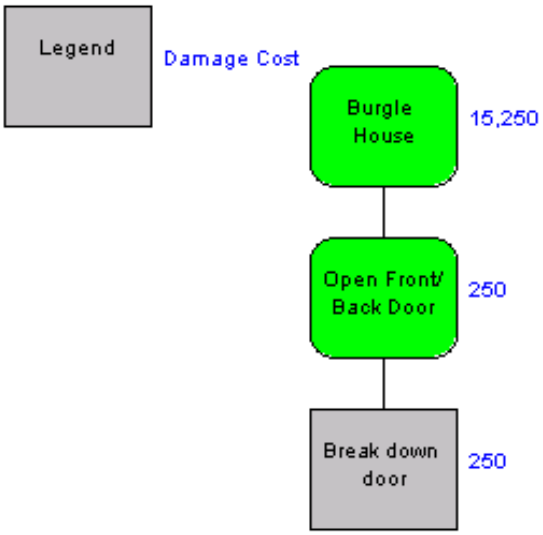

Figure 13 – Juvenile Delinquent Monetary Impact on Victim - Attack #2

scenarios, complete with monetary impact values, are shown in **Figure 1**2, **Figure 13** and **Figure 14**.

Figure 12 is the lowest impact attack scenario for the home owner. The damage is $150 to fix a broken window plus $15,000 in stolen goods. The total is $15,150. **Figure 13**'s attack is only marginally more expensive. The $250 of repairs to the door are slightly more expensive than replacing a pane of glass. The big jump in victim impact comes in **Figure 14**. In that scenario, the juvenile delinquent breaks a car window ($200) to steal the garage door opener ($50). Once inside the garage he discovers the gold mine of sporting goods and makes off with an additional $3,000. Finally, he forces the passage door leading to the house ($250) and steals $15,000 of household items. The total bill comes to $18,500.

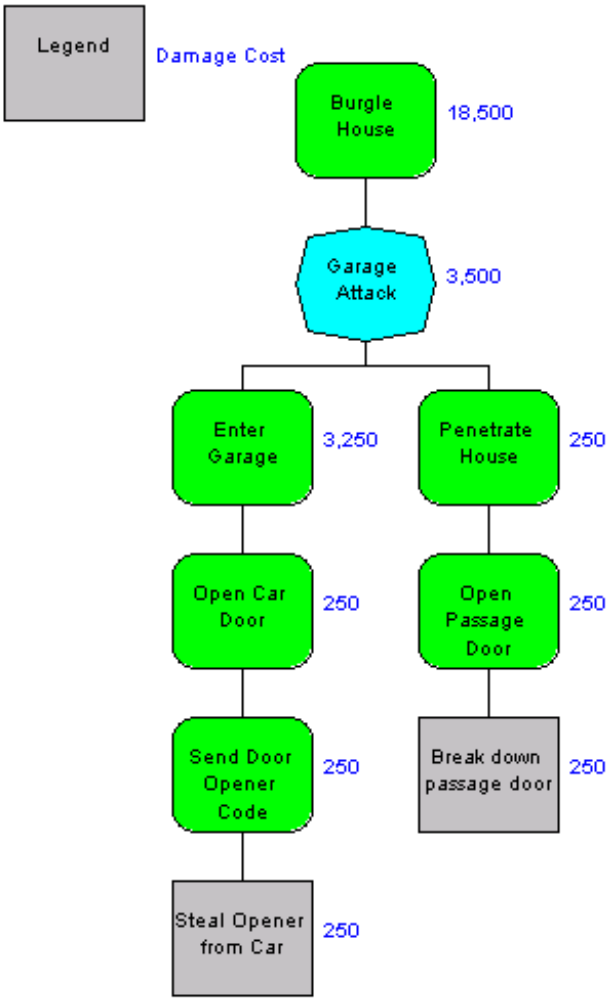

Figure 14 – Juvenile Delinquent Monetary Impact on victim - Attack #3

With no further information about the likes and dislikes of juvenile delinquents, it is not unreasonable to assume that the probability of all three attacks is roughly equal. This means that the attack with the highest impact (attack scenario #3) has the highest risk.

It is important to understand why this is so. The three attacks shown are probable because they are possible by the threat agent under consideration who is motivated to carry out an attack. This is the first term in the risk equation. The impact then completes the equation and shows relative risk.

Threat Agent Motivation

To this point, our analysis has made the simplification that all threat agents were equally motivated to harm the system. Clearly, this is not generally true. Fortunately, if the attacker's psychology is understood, it is possible to include attacker motivation in an attack tree model.

Attackers can be motivated by many things. The home security model we have discussed assumes that the burglars want to enter the house and steal valuables. This can easily be modeled by creating an *impact indicator* that measures the monetary benefit to the attacker. It will be very similar to the victim's damage cost

indicator, except that the attacker will not gain any benefit from the breakage of security components (such as locks, windows) whereas the victim most certainly will feel an impact. By creating a list of attack scenarios it is possible to see which attacks are most attractive to the threat agent. In other words, it is possible to see which attacks are most likely to motivate them.

By comparing the resources expended by the threat agent in a particular attack scenario with the benefits obtained it is even possible to calculate a *return on investment* for the attacker. This may provide an even more sophisticated view of threat agent motivation. However, caution is urged in placing too much confidence on these types of analyses[26]. They depend heavily on an understanding of threat agent psychology. Deranged people, or people from cultures with values that differ greatly from ours, may think in ways we cannot predict. By assuming that threat agent behavior is constrained largely by capability we err on the side of caution. Even madmen can only achieve what their resources allow. This is the power of capabilities-based analysis.

Mixing Probabilities and Capabilities in Behavioral Indicators

Some analysis models are purely probabilistic. Others consider only hostile threats. In these situations there is no confusion about how to set up behavioral indicators. Problems arise where a model must handle both types of threats. This can occur in two ways.

The simplest case is when the probabilistic and hostile threats against a system have little or no interaction. For example, a manufacturing facility may be concerned about hostile threats of break-ins leading to the theft of merchandise or equipment. At the same time, they may be located in a flood plain and be worried that the river will rise and cause water damage to their goods. Fire from a nearby forest could also cross their perimeter. The easiest way of handling this type of situation is to make two separate attack trees and perform the analysis separately.

In some cases, however, the threats are interrelated (**Figure 15**). Suppose that the manufacturing plant has a perimeter consisting of a fence with alarms and guard posts. This defense might normally be difficult to penetrate. However, a flood may weaken the perimeter by interfering with alarms and changing guard procedures. The flood lowers the bar sufficiently so that certain threat agents (that would otherwise be thwarted in their endeavors) can penetrate the facility.

[26] In the house burglary example it is tempting to use an impact indicator showing the amount of loot the attacker would get in a given attack as an indication of the attacker's motivation. However, our model was based on the fact that the attacker is unaware that there are valuables in the garage, and only discovers this accidentally when certain attack scenarios are used. This is not uncommon. The results of attacks frequently surprise their perpetrators.

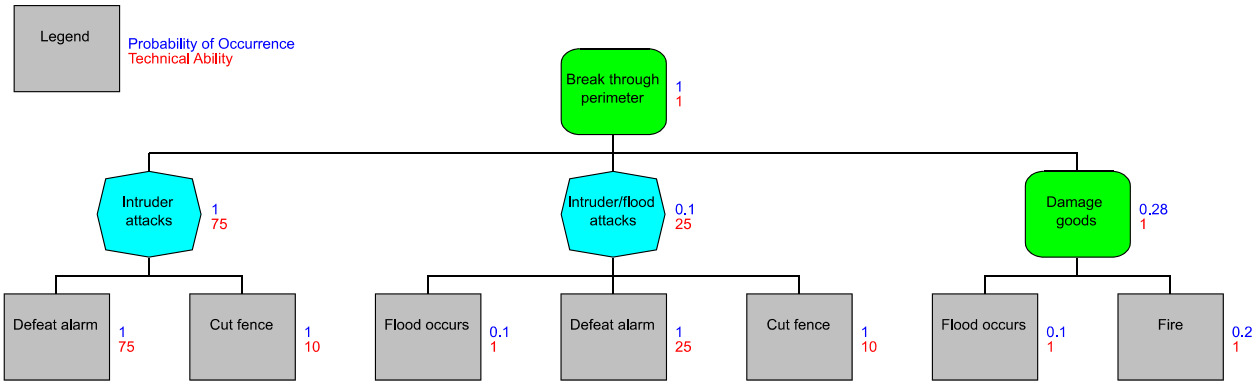

Figure 15 – Mixed Threats Against a Perimeter

While there are no perfect techniques for merging dissimilar behavioral indicators, there is a useful trick for working around the problem. Behavioral indicators for the tree should be divided into two categories: environmental and causal. The environmental indicators reflect the *probability of occurrence*. The causal set of indicators relate to the capabilities of malicious threat agents (e.g., *cost of attack, technical skill*). Leaf nodes representing environmental events should set their environmental behavioral indicators to values found from statistical sources. The causal behavioral indicators for the environmental nodes should be set to "easy" values. I.e., the thresholds should allow any threat agent to achieve them. This indicates that the environmental events are totally controlled by statistical events.

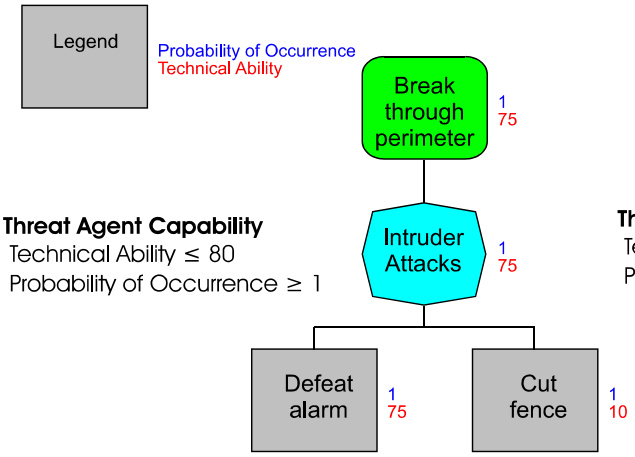

Figure 16 – Skilled Attacker only

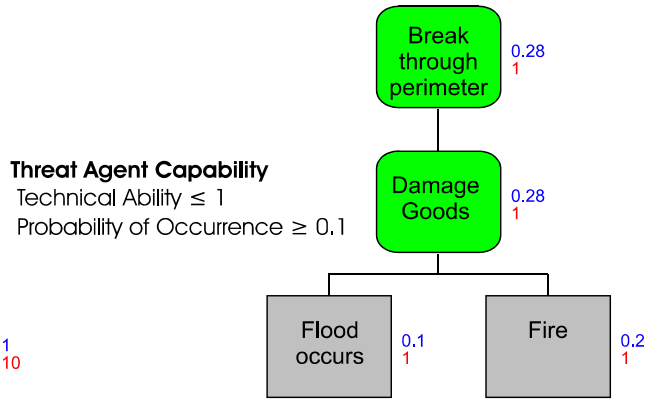

Figure 17 – Mother Nature only

Conversely, leaf nodes relating to causal activities should have their environmental thresholds set to unity (1). This means that their likelihood is totally controlled by attacker behavior. Then, the tree can be pruned to examine three situations. Which incidents will occur based solely on adversarial capability (**Figure 16**), which incidents will occur strictly based on probability (**Figure 17**), and which incidents will occur given a certain level of adversarial capability and a certain probability (**Figure 18**). Care must be used in setting up and interpreting mixed indicator situations.

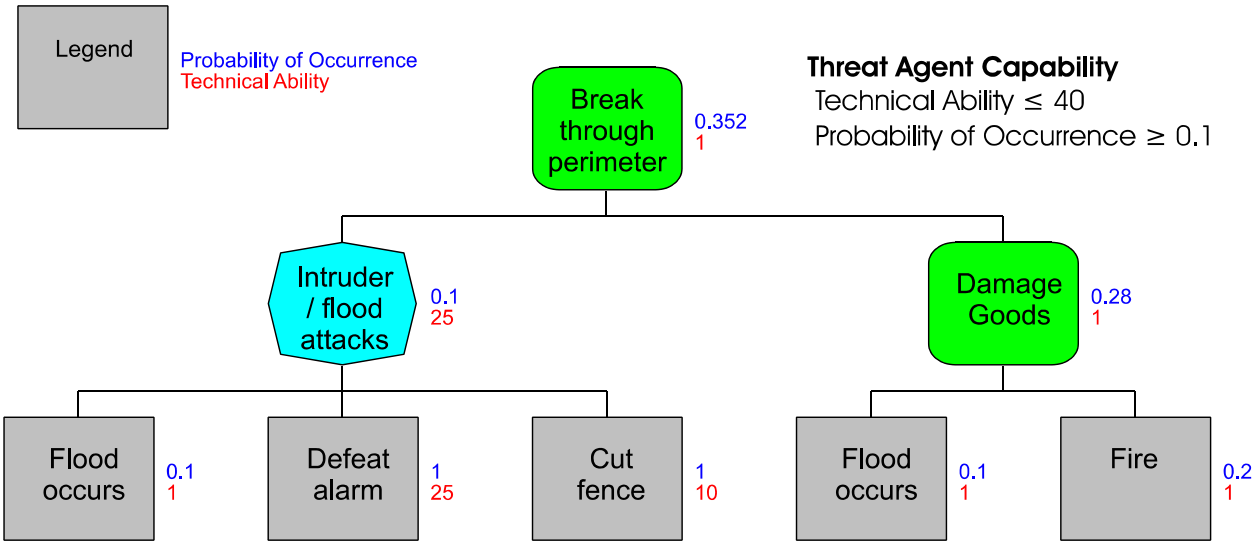

Figure 18 – Combined Threat of Mother Nature and Semi-skilled Attacker

The Need for Analysis Tools

The techniques described earlier are based on simple concepts. The examples given have, for the most part, been small enough that they could be carried out by a person with pencil and paper. However, these operations quickly become unwieldy when applied to attack trees of sufficient complexity to describe meaningful problems. All desire to experiment with the model by changing the tree, altering the assumptions about the threat agents would quickly vanish once the analyst realizes the effort required to prune or calculate attack scenarios.

The answer to this problem is to provide a software tool capable of performing these operations with the click of a mouse. Just as a spreadsheet program removes the tedium of performing the cascading calculations caused by updating a spreadsheet cell, an attack tree analysis tool can free the analyst to use his or her insight in understanding the system.

Amenaza Technologies has produced such a tool. Secur*I*Tree® is the world's first commercially available attack tree modeling and analysis tool. With Secur*I*Tree you really can see the forest through the trees!

Advantages of Attack Tree Analysis over Traditional Risk Analysis Methodologies

A conventional, statistics-based risk assessment might tell you how likely you were to have your house burglarized and what damage you might expect to suffer. Unfortunately, it would not provide anything more than general guidelines (so called, "best practices") on how to be more secure.

The attack tree analysis allows us to see the underlying forces that channel the attacker's behavior. For example, the home owner might notice that, simply by removing their garage door opener from the car in the driveway, the riskiest attack from juvenile delinquents is eliminated.

Although statistics might be available for house robberies, this is not true of many other types of illicit activity. Without statistics, the conventional risk analysis process is unable to

convincingly predict which attacks are likely to occur. This leaves the analyst to rely on guesses which, even if correct, are not supported by evidence. If the analyst chooses to alter the system to mitigate certain risks, the conventional risk assessment methodology provides no suggestions as to which changes will be most effective. The result is a series of highly subjective decisions for which the reasoning process is undocumented. Sooner or later, problems arise – and no one can remember the rationale behind the recommendations. This leads to indefensible due diligence positions and legal exposure.

Attack tree models are largely self documenting. The assumptions about the systems vulnerabilities are captured in the tree itself. The assumptions about the threat agents are stated in the table of threat agent capabilities. The conclusions are reached through the mathematical operation of applying the threat agent's profile to the model (pruning). This is much more reliable than depending on an analyst's memory.

Conventional approaches to risk analysis are also very time consuming. This makes analysts reluctant to update them when the system or environment changes. As a result, only a snapshot of the system at some point in time is considered. By the time the analysis is complete it is no longer relevant because the system being studied has changed. Attack tree models can (with the proper tools) be updated and reevaluated in minutes.

Methodology

Attack Tree-based Risk Assessment Deliverables

The previous section described the theoretical underpinnings for *capabilities-based attack tree* analysis. To be useful, these principles must be applied in a structured way. This is known as a *methodology*. Organizations are generally seeking three things from a risk analysis methodology:

A Credible *Due Diligence* Defense

We are not lawyers. Generic legal advice is of limited value anyway since laws vary by jurisdiction, market sector and the specific situation. However, we are told that due diligence laws generally require that an organization be able to demonstrate that they have studied their potential risks and carried out the actions expected of a prudent man. Attack tree analysis is a formal way for identifying risks and making decisions about those risks.

All too often organizations fail to document the reasoning process they use in deciding which risks to address or how they choose mitigation strategies. Attack tree analysis captures this information and logic in a mathematical model which can be exhibited as required – even if the original analysts are no longer available!

Attack tree analysis documents which threats, vulnerabilities and projected impacts of attacks were considered during analysis. An attack tree model is able to

a. reproduce the reasoning used to reach decisions
b. eliminate or reduce the dependency on human memory
c. demonstrate the reasonableness of decisions.

Even if your analysts are geniuses that do not require tools and techniques to make the right decisions, if you can't defend your actions you may still be found wanting in due diligence situations!

Identify effective security solutions

Management is tired of solutions that never seem to deliver on their promises. This is particularly true in the area of information technology. All too often management views security people as "Emperor's Tailors." As you will recall, a certain emperor was once victimized by a pair of fraudulent tailors who purported to sew clothes out of a magical fabric that possessed all sorts of wonderful properties. The emperor couldn't quite see what the proposed garments would deliver, but gave the tailors the benefit of the doubt. After being caught naked in public he vowed never again to be taken in by charlatans. The emperor is now your boss and won't approve anything unless it is absolutely certain to bring results. Attack tree analysis allows you to show what proposed solutions will deliver before you buy or implement.

Ensure cost effective security solutions

Not only must the solution work, but the cost of the solution must be justified by the risk. Demonstrating return on investment (ROI) for security tools has always been difficult. Heisenberg's Uncertainty Principle states that the act of observing a system changes its behavior.

Although Heisenberg was referring to physical systems when he made this statement, nowhere is this more true than with a security study. Risk mitigation measures prevent risky incidents from occurring. By so doing, they eliminate the evidence that would have proved their value.

Attack tree models and scenario analysis show which situations are likely to arise and how much damage will result. The analysis can then incorporate various mitigation mechanisms and demonstrate the costs and savings that will occur. This is as close to an ROI calculation as is practical in the circumstances.

Universal Methodology

The exact details of a methodology must be tailored to meet the culture and procedures of the organization concerned. Amenaza generally recommends a four step risk assessment process.

1. **Produce an attack tree model that shows how an incident can occur.** This includes a representation of the vulnerabilities and the resources required to exploit them as well as the impact to the victim of the various attacks. The model provides insight into techniques that could be used to harden a system or improve its design.
2. **Identify which vulnerabilities will be exploited by a given type of threat agent**. Identify who your enemies are. Predict where and how they will strike.
3. **Produce a prioritized list of the risks associated with each type of attack.** Few organizations are capable of correcting every vulnerability. The prioritized list provides all the information needed to justify which problems need to be resolved and demonstrate that the proposed corrective action is cost justified.
4. **Propose effective mitigation strategies.** Show how changes to the system will provide effective (and cost effective) improvements.

Sample Information Technology Methodology

As mentioned above, the detailed risk analysis activities will vary depending on the type of system being studied. The steps discussed in this section are specific to information technology applications. It is hoped that people in other disciplines will see how similar steps might be applied in their discipline.

Step One – Model the Information System in an Attack Tree

Scope Definition

Before we can evaluate the risks associated with a particular information system we must first identify what makes up the system. In the vernacular of risk assessors the system is often

called the *system under scrutiny*[27]. Clearly, the *system under scrutiny* includes all of the computing devices that process and transmit the system's information. However, the system may also include computers which provide infrastructure support. Compromising the security of the network infrastructure may have a serious effect on the computers that process the information.

The people who work with the information are also part of the system. Identifying them is trickier than it sounds. While it is possible to follow cables and find out which computers are interconnected it isn't always as obvious how people communicate.

There are practical limits as to how much can be included in one system. It is tempting to say that a thorough risk assessment should consider all components that are on the same network. Since most corporate networks now have a connection to the Internet (which has millions of hosts) this simply isn't practical! Instead, we need to include in our study those computers whose function will most likely affect the information provided by the system under scrutiny.

Questions to Help Identify Major Components in the System

Answer the following questions to identify which computers are major or primary components of the *system under scrutiny*.

- What is the name of the application being analyzed? _____
- What computing platform does it run on? _____
- Is this an end-user application or is it infrastructure? _____
- Does this system meet the minimum corporate security standard? I.e., has it been inspected and found to meet corporate security practices? _____
- Whose primary job function depends on this system? _____
- Does the system deal with process control? _____
- Does the system deal with financial data? _____
- Does the system deal with human resource information? _____
- Is customer data involved? If so, what data? _____
- Does the system process shareholder data? _____
- Which departments directly use the system? _____
- Who, outside of the organization, uses the system? (E.g., business partners, wholly or partly-owned subsidiaries.) _____
- System access is:

[27] We will generally just use the term *system* unless this is ambiguous. It is important that you are familiar with the term since it is often used by risk assessors.

- Via the corporate campus network? _____
- Using a dial-up connection? _____
- By way of a Virtual Private Network link? _____
- What form of authentication is used? _____
- Via the Internet web portal? _____
- Using direct leased line (WAN) connections? _____

Questions to Identify Supporting and Dependent Components

The system's major devices normally require infrastructure (network) services in order to operate properly. Technical people can usually supply this information. The application may require access to name servers, file servers, authentication servers, time servers. In addition, the major components may interact with other applications and databases. The following questions will help identify supporting and dependent components.

- Which in-house computers exchange data with the system under scrutiny? _____

  _____
- Which systems outside of the organization exchange data with the system under scrutiny? _____
- Does the system under scrutiny exchange information with hosts that are accessible from the Internet? _____
- Are any of the information exchanges transmitted using communication channels that are not part of the corporate network (LAN)? _____
  - Which systems transmit data to the system under scrutiny? (Don't forget to include network file servers in this list.) _____
  - What will happen to this information if the system under scrutiny is unable to receive the data? _____
  - Will this adversely affect the sender's system? _____
- Which databases are accessed by the system under scrutiny? _____

  _____
- Which group is responsible for supporting the system under scrutiny? Be sure to include all aspects of support: hardware, OS, application, data. _____

  _____

Determining which supporting components to include in the study requires a great deal of judgement from the analyst. If you include too many irrelevant components in your assessment then the task may become intractable. Eliminating supporting components that sustain the major components will cause your analysis to ignore important vulnerabilities. It is probably better to

err **slightly** on the side of including unrelated components in the system scope.

We are interested in interested in facilities that sustain the system under scrutiny because these represent potential areas of vulnerability – attack points. Neither can we ignore other systems that depend on information or services from the system under scrutiny. The impact of a failure of the system under scrutiny on dependent systems must be included in the estimate of an incident's impact.

Unrelated Systems

Is it even relevant to consider the role of unrelated systems in the risk assessment? Usually, the answer is no. However, the analyst must understand the expectations of the people who have commissioned the risk assessment. The typical high level assignment is to "find the risks associated with a failure of the system under scrutiny. Most people interpret this to mean, "What can happen that will affect the confidentiality, availability or integrity of the information in the system being analyzed and what impact with this have on business?" Answering this question only requires that the major, supporting and dependent components be included in the study.

Sometimes, however, it is expected that the analyst will identify situations where a successful attack against the system under scrutiny may place the attacker in a position to harm other, unrelated systems. Systems that are unrelated from a business point of view may not be unrelated from an attacker's point of view.

The analyst should make certain what is expected. If management expects that attacks using the system under scrutiny as a springboard are part of the mandate, then this will greatly expand the scope of the assessment. It could conceivably include every other system in the organization! That would require that an attack tree be constructed for every system[28] on the network. These attack trees would focus on the exposures resulting from a compromised system under scrutiny, but this still might involve a tremendous amount of work. There is a good chance that the assessment might never be completed.

A possible way out of this conundrum is to note that the secondary effects of an attack against the system under scrutiny would be avoided if the system was secure. If this is not possible or economical, then automated intrusion detection facilities would at least allow administrators to shut down the compromised system before it could be used as a springboard. These efforts only require analysis of the major and supporting components.

[28] The attack trees for unrelated systems would only need to consider vulnerabilities resulting from a compromise of the system under scrutiny.

Identification of the People in the System

Any group that makes direct use of the system[29] should be included in the risk analysis discussions. If the system is even indirectly related to finance, human resources or other areas where conduct is regulated, they should also be included. Groups that cannot participate directly should have their interests represented by the group managing the infrastructure and by the team that provides support for their computer system.

Most organizations are extremely concerned about the possibility of an incident affecting external customers. However, it is not always possible to invite customers to participate directly but their needs and opinions should be incorporated. The analyst must identify their requirements in the study.

Identifying Business Processes Dependent on the System

Key decision making processes may depend on the system. Managers and executives base their actions on the information at their disposal. These senior level people often do not use the computer systems directly. Rather, they rely on reports that were generated indirectly using system information. Identifying the people involved in these information flows can be difficult,[30] however it is essential that the impact of total and partial system failures are identified. The following questions may be helpful.

- What decision making processes would suffer if the system under scrutiny were unavailable or the information it supplied was inaccurate? _____

  _____
- Who would be embarrassed if they could not supply information due to a failure of the system? _____
- What opportunities might not be realized due to a failure of the system under scrutiny? I.e., what is the opportunity impact of a failure? _____

  _____
- How long could you survive without this system and what sorts of things would fail if the outage were longer? _____

  _____

Communicate with the Stakeholders

At this point, a number of stakeholders will have been identified who ought to be involved in the risk analysis process. Contact each group. Explain that a risk assessment of a system they

[29] Sometimes the system being analyzed is infrastructure. In this case it probably affects a large number of other systems (and departments). It may not be feasible to include all of these groups in the analysis process so a representative sample should be chosen.

[30] Actually, it's easy. Just turn the system off and wait. Your successor will find the results very useful!

use is being performed. Explain the importance of their perspective and invite them to participate. If the organization declines the invitation then the IT department will have to bear the responsibility of representing them alone. IT Security may become involved to ensure that the department's (and corporation's) interests are considered. The analyst should create a table for the system under scrutiny. In can be similar to the one shown below for the hypothetical inventory system:

Risk Assessment Participants – ***Widgets-R-Us Inventory System***		
Business Unit	**Relationship to System**	**Representative**
IT Services Department	Operational support for hardware and OS	Self represented
Database Support	Operational support for database	Self represented
Manufacturing	Direct user of application	Self represented
Market Planning	Indirect user of application	IT Security
Vice-President of Marketing	Makes decisions based on daily reports received from the system	Corporate audit

Their Dream, Your Nightmare – Identifying the Attackers' Goal

A key step in attack tree analysis is in identifying the top level, root goal. This represents what the attacker wants to achieve (the thing that the defender of the system does not want to happen).

Usually it is possible to identify one high level goal that makes sense for a wide variety of attackers and situations. For example, *Compromise Inventory System.* Sometimes, however, this is not sufficiently precise.

Information technology security focuses on the confidentiality, integrity, availability and regulatory compliance[31] of information. If the attacker will do more or less the same things to compromise each of these then a single attack tree will probably suffice. If the attacks or the impact differ significantly, it may be necessary to create several goals, one for each facet of security.

Create Attack Tree Model(s)

Using the information that has been gathered and the root goal(s) that have been identified,

[31] For example, privacy regulations may require that certain personal information be encrypted regardless of whether it is protected in other ways.

create an attack tree model or models. The model should define the appropriate behavioral and impact indicators, and populate them with the information discovered in the information gathering phase. Remember to add abundant documentation since the model may eventually be used in a due diligence defense.

Knowledge Reuse

Most commercial information systems are based on a small set of core technologies. There are only a limited number of operating systems and databases in popular use. Once these components have been analyzed and modeled in an attack tree the information can be stored in a library and reused.

Amenaza Technologies produces a set of attack tree libraries for popular information technology components. Organizations are encouraged to build up libraries of their own technologies. Regardless of the source of the libraries, it is essential that they be reviewed for applicability and tweaked if necessary. Not everyone deploys the same components in exactly the same way.

When creating a technology library the analyst should focus on the product's architecture. One or more of the numerous vulnerability databases should be consulted to get a list of vulnerabilities and known exploits. These weaknesses will reveal architectural deficiencies. A sample of specific vulnerabilities should be included in the library, but it is usually unnecessary to find every known defect.

Step Two – Identify Exploitable Vulnerabilities

The attack tree model shows all the attacks that could be used by an infinitely wealthy, powerful, daring and intelligent enemy. This set must be reduced to those attacks that are probable by a more limited adversary.

Threat Agent Selection and Definition

Anyone or anything that wants to harm the system is a threat agent. It may be helpful to break the threat agents into two categories: deliberate and unintentional. Deliberate threat agents include hackers, competitors, employees and special interest groups (e.g., radical environmentalists). They attack systems with viruses, Trojan Horses, unauthorized access, denial of service attacks and social engineering.

The primary unintentional threat agent is Mother Nature. She can hit us with environmental mishaps such as floods, fires, earthquakes, ice storms and tornadoes. We also include accidental human errors in the category of unintentional threats. If we need a name for the threat agent responsible for silly errors we can use *Murphy*, of Murphy's Law[32] infamy.

Some time ago we saw a case where a server with multiple network interfaces had its network connections accidentally swapped. Without going into the details, a number of network switches became confused. The problem was difficult to isolate and brought network

[32] If anything can go wrong, it will.

communications to its knees. People were unable to use the systems for days. Although the error was accidental, it was no less harmful than an intentional attack.

In many cases, an organization has a good understanding of potential natural disasters. They may be confident and satisfied in the steps they have taken to mitigate these risks. In that case, the analyst can document that the analysis will only consider deliberate attacks. Obviously this decision will require the support of the other participants in the study as well as the backing of whoever commissioned it.

Otherwise, both environmental and deliberate threats will need to be included in the study. If the two types of threats are not interdependent, then it is probably best to create two separate attack trees. The tree representing deliberate threats will use indicators showing the resources required to carry out the attacks. The accidental or environmental tree will be statistics based.

If the two threats are interdependent, or if one threat type greatly predominates, then a single tree should be used. This may require some of the tricks shown earlier on analyzing mixed threats (shown on page 29).

Some people have difficulty understanding the difference between threat agents and threats. Giving them examples of threat agents and the attacks they use may help clarify the concept.

Threat Agents and Attacks		
Threat Agent	**Type**	**Sample Threat**
Hacker	Deliberate	Malware – viruses, worms, trojan horses
Employee	Deliberate	Altering information in database for their own interests
Competitor	Deliberate	Stealing information to help their own product
Mother Nature	Unintentional	Flood, fire, tornado
Murphy	Unintentional	Trip on power cord. Type wrong command.

The best way to agree upon the threat agents is in a brainstorming session with all of the risk assessment participants. This is workable when the number of participants is small, but it may not be realistic to try and gather together a larger group. If that is the case, the risk assessor should interview each of the participants to solicit their input as to which threat agents need be considered.

Ask the following questions to identify possible threat agents.

- What are your worst nightmares about what might cause this system to fail? ____

  _____

- What is the most unusual or bizarre thing you could contemplate happening to the system? _____

- Who would be pleased if you could not access your system or could not depend on the information it provided? _____

  _____

In general, if you aren't certain whether or not to include a threat agent it is better to include it.

Create a table listing the various threat agents that have been identified. The table should list the estimates of the threat agents' capabilities for each indicator in the tree model. Be sure to document the reasons for selecting the capability values.

Threat Agents' Capabilities			
Threat Agent	**Money to Spend on Attack**	**Skill (1 - 100)**	**Tolerance of Apprehension**
Hacker	$50	35	50%
Employee	$50	25	5%
Competitor	$10,000	75	2%

Discard Attacks Beyond Threat Agents' Capabilities

For each threat agent, compare the resources required to exploit each vulnerability in the attack tree model. If the exploit exceeds the capability of the threat agent, remove (prune) it from the tree. Those vulnerabilities which remain are those which are probable (for each type of threat agent).

Confidence Estimation

Several techniques can be used to estimate the reliability of the prediction as to which attacks are probable. The first is known as *sensitivity analysis*. To perform sensitivity analysis, increase the capabilities assigned to each threat agent slightly and repeat the tree pruning. If the results do not change (or change insignificantly) with adjustments, then it is safe to say that the prediction about the threat agent is insensitive to errors in that indicator.

Another helpful activity is to create a list of all nodes which have been pruned (for a given threat agent). The list should detail which resource constraints caused the node to be removed. Nodes which are pruned on multiple indicators are less likely to be exploitable for the chosen threat agent.

Step Three – Create a Risk-prioritized List of Attack Scenarios

Knowing which attacks will be performed by each threat agent is only half the battle. To understand the risk associated with an attack it is necessary to know its impact.

Generate Attack Scenarios for each Threat Agent

Create a set of attack scenarios for each threat agent being considered. This will contain all possible ways that a threat agent could use to attack the system.

Prioritize Attack Scenarios by Impact

The attack tree model should contain one or more indicators of the harm or loss that the victim will suffer through an incident. Sort the attack scenarios by impact level. This will have to be done once for each impact indicator in the model. The results are risk prioritized lists to the victim for a particular type of damage by the specified threat agent.

Step Four – Identify Effective Mitigation Strategies

In certain (rare) cases, the risk analysis will show that the risks associated with a system are all acceptable to the corporation. Generally, the analyst will need to make a recommendation that deals with excessive risk. This can be achieved by assigning the risk to others (waivers, insurance). More commonly, changes must be made to the system to reduce the probability of an attack or its impact. Attack tree analysis is ideal for investigating proposed solutions to security problems. Solutions include changes to policies, procedures or technologies.

Examination of the attack scenarios generated in the previous step will often lead to insight about defects in the security architecture. The analyst can imagine a variety of solutions and alter the attack tree model to reflect these changes. That is, the analyst begins again at step one (*Model the Information System in an Attack Tree*) and proceeds through steps two and three as before. Solutions that "work" will result in the removal of attack scenarios. By comparing the impact cost of an attack scenario with the implementation cost of the proposed solution it is possible to identify worthwhile solutions. By comparing the cost and effectiveness of a variety of proposed solutions, it is possible to choose the best solution.

A similar, iterative process should also be applied whenever changes will be made to a system. This allows risk assessment to be proactive instead of reactive.

Future Directions

The four steps identified in this methodology are tried and true. They deliver excellent results in a fast, defensible way. However, as research continues, new steps may be added to further increase the benefits of attack tree analysis.

One idea that looks promising is the integration of attack tree analysis with intrusion detection systems (IDSs). Current IDSs operate through pattern matching against a list of "signatures." At some point in the future, it may be possible to use attack tree models for comparison and detection.

Conclusions

This document has presented the basic concepts of capabilities-based attack tree analysis. A sample methodology suitable for use in an information technology environment has been demonstrated. Hopefully people in other disciplines will take inspiration from this example and create methodologies appropriate for their own environments.

The best way to learn about attack tree analysis is to take the concepts that have been presented and tackle some of the problems that concern you. Amenaza Technologies Limited is a world leader in this field. We would be delighted to help you by providing training, software tools or consulting services. We also appreciate hearing your experiences with attack tree analysis.

Amenaza Technologies Limited