Attack Tree-based Threat Risk Analysis

by Terrance R Ingoldsby
Amenaza Technologies Limited

Amenaza Technologies Limited
406 – 917 85th St SW, m/s 125
Calgary, Alberta T3H 5Z9
Canada

1-888-949-9797 toll free US & Canada
+01 403 263 7737 International

E-mail: Terry.Ingoldsby@amenaza.com
Web: www.amenaza.com

Table of Contents

List of Figures

Foreword by the Author

I originally heard of attack trees through presentations given by the noted cryptographer and security researcher Bruce Schneier in the late 1990s at computer security conferences. I had been working in the cybersecurity field for some time and was generally dissatisfied with the then popular threat and risk modeling methodologies. Schneier's talk electrified me. My first degree is in the field of physics and I like to evaluate problems in a methodical, scientific way. Until learning about attack trees, I had observed remarkably little science in the field of cybersecurity. Attack trees promised to bring greater rigor and objectivity to hostile risk analysis.

Unfortunately, when I attempted to learn more about attack trees I discovered that there were very few references on the subject. The few papers I found were either elementary or assumed that the reader already knew all about attack trees. Not dissuaded, I determined to learn more about attack trees and to create software to support the analysis process. Late in 1999 I began to assemble a team to design and create a software tool. Early in 2001 my colleagues and I incorporated Amenaza Technologies Limited – a company dedicated to the creation of state of the art threat modeling software. Amenaza inherited and carried on the work done prior to Amenaza's incorporation.

Having organized a company the only problem was that none of us had more than a vague idea of what said software was supposed to do! Much of the initial research and development work involved trial and error. A rapid development approach was used to create the initial versions of the Secur*IT*ree® software – new versions emerged every few days!

With each new version test models would be created of real world systems and analysis performed. As chief technical architect of Secur*IT*ree, all too often my feedback to the development team would be, "Great job, you gave me exactly what I asked for. Unfortunately, I was wrong in my requirements – change the software to do {something quite different}." It is a tribute to Amenaza's head of development, Christine M McLellan, that she became expert in delivering what I should have asked for, instead of what I actually requested!

Through this process of trial and error, I learned what worked and what didn't. The software evolved and improved. Then, around 2003, I had the incredible good fortune of making contact with some of the people who originally conceived of attack trees. These individuals worked for highly respected government agencies, prestigious academic/research institutions and high end consulting organizations. As is often the case with individuals working in these settings, they have not received the recognition they deserve. Discussions with these people largely confirmed the findings of Amenaza's independent research. When large organizations with brilliant people (and very large budgets) and a small team (with an extremely small budget) work independently to arrive at similar solutions this validates the techniques.

The discussions with the originators of the attack tree methodology also revealed analytic techniques that were only feasible because their originators had extensive mathematical backgrounds and could quickly whip up a custom program to implement their mathematical concepts. It then became Amenaza's task to capture the essence of these ideas in Secur*IT*ree in ways that mere mortals could use.

I have written white papers on attack tree analysis since about 2006. In this 2021 paper, I hope to collect the information that was previously scattered across several previous papers and also to include some of the new techniques that have been developed more recently.

Terrance R Ingoldsby

Attack Tree-based Threat Risk Analysis

Introduction

Risk analysis is as old as civilization itself. People quickly learn that there are pros and cons to every choice they make. Repeated observations and experiences lead to an intuitive sense of risk in a given situation. Unfortunately, there are limits to intuition's ability to cope with changing variables and unfamiliar situations. Additionally, the human mind seeks to find patterns, even where none exist (for example in the behavior of a slot machine). Using intuition to predict outcomes in these situations generally leads to poor decisions.

Nowhere is this truer than in the field of hostile risk analysis. Modern civilizations provide an unprecedented level of safety and security to their citizens. Even people working in the security field often have limited first hand experience in fending off attacks. As a result, a variety of methodologies have been developed to help analyze the risks from hostile threats. Unfortunately, many of these systems are based on simple checklists which are overly general in nature. Other approaches are highly subjective and fail to capture the logic behind the analysis. **Attack tree-based threat models provide a more rigorous, engineering-like approach to hostile risk analysis.**

The techniques of attack tree analysis have been known by expert practitioners for almost thirty years. A number of papers have been published on the subject. However, there seem to be few publicly available documents that provide comprehensive coverage from basic principles to advanced techniques. This paper attempts to fill that gap.

One glaring problem with many existing hostile risk analysis strategies is that they focus exclusively on the *system* the defender is trying to protect. A knowledge of the characteristics of both the defender's system, and the adversaries that threaten it, is required to understand how the two will interact. This enhanced understanding is essential in estimating risk. Accordingly, the techniques described in this document emphasize the roles of both defenders and adversaries.

Attack trees are models of reality. That is, they are a simplified representation of complex real world objects and forces. The accuracy with which the underlying drivers are known depends on many factors including the time and effort spent studying them. In some cases it becomes necessary to make assumptions based on the best information available. Of course, the accuracy of the analysis will be limited by the correctness of the assumptions. **All models, including attack trees, will break down if they are used beyond their limits.** The conclusions reached by any risk estimation scheme (including attack trees) should be subjected to a reality check and compared to the results from other methodologies.

Despite this note of caution, it should be noted that <u>all</u> predictive mechanisms depend on assumptions. It is a serious problem when analysts begin to treat their assumptions as facts and are surprised (sometimes disastrously) when their conclusions are proven wrong. Attack trees provide a discipline for declaring and understanding assumptions. Exposing assumptions to review and critique makes unpleasant surprises less likely.

Hostile risk analysis is not the first risk discipline to use tree structures. Fault (or failure) trees have long[1] been used to understand how component failures affect overall system function. Fault trees are useful for understanding the risk associated with random (stochastic) events, including incidents caused by Mother Nature, human error and equipment failure. This paper explores mechanisms for merging both hostile and random risks into an integrated tree-based model.

One of the most significant differences between attack tree analysis and some other hostile risk analysis methods is that attack trees are built largely from the point of view of the attacker (instead of the defender). Attack tree models excel at estimating the risk for situations where events happen infrequently or have never happened before.

Security practitioners have always found it challenging to provide convincing evidence that the countermeasures they deploy were responsible for preventing attacks. It is fundamentally difficult to provide conclusive proof of why an event doesn't happen. In fact, management often uses the absence of an event as evidence that they needn't have spent any money on preventive measures[2].

The challenge of justifying controls is further exacerbated when dealing with unprecedented or infrequent events. In this case, statistics (which are based on a representative sample of events) cannot demonstrate that any risk exists. Nonetheless, as the 9/11 tragedy sadly demonstrated, the absence of a statistical precedent does not provide any assurance that the event will not occur. The adversary may choose to create novel events precisely because the victim is unprepared for them – leading to exceptionally devastating results.

In theory it should be possible to compare the results predicted by the attack tree methodology to the frequency of common hostile events (for which statistics are readily available). Such comparisons are not as easy as they might appear. Statistics are a generalization and may not be relevant to a particular situation.

Attack tree analysis incorporates information about a specific defender's adversaries and the benefits they will realize from carrying out an attack against a particular defender. This precision is a virtue because it offers the hope that predictions will be more accurate for a given situation than statistics. This specificity makes it difficult to compare defender-specific predictions with statistics that are generalized over a wide variety of defenders and attackers.

Consider two technically identical information systems. Although the systems may use the same hardware, operating systems, data bases and so forth, the actual information stored within them might differ greatly. One system may store the plans for an advanced military weapon and the

---

[1] Fault trees were invented in the early 1960s for use in the Minuteman Missile System. Clifton A Ericson II; Fault Tree Analysis – A History from the Proceedings of the 17th International System Safety Conference, 1999.

[2] The author recalls spending the night of 31 December 1999 at a computer operations center due to management concerns that staff be on stand-by in case of any Y2K glitches – only to later overhear management complaining that they didn't know why they spent so much money on Y2K remediation efforts since "nothing happened." Of course the reason why nothing happened was because of the extensive preventive work that had been done!

other a particularly good pizza recipe from the local mom and pop restaurant. Clearly the two systems will attract different types of adversaries who possess different levels of skill, resources and motivations. Additionally, the impact of the theft of the pizza recipe would presumably be far less than the disclosure of the secret military design.

These differences make it difficult for general purpose statistics to provide meaningful estimates of probability (and risk) for specific cases. Fortunately, attack tree-based threat models are capable of providing risk estimates for specific situations. Admittedly, creating attack models does require more effort and expertise than simply referring to some table of statistics. However, the advantage is that system stewards are able to construct defenses that more effectively (and cost effectively) address their most critical risks.

Despite the caveats mentioned above, the widespread and increasing usage of attack trees by aerospace, electric power, defense and intelligence organizations demonstrates the confidence that they place in the technique. Attack trees encourage a rational thinking process. They are just as applicable in less esoteric applications, and are becoming more commonly used in commercial, medical and critical infrastructure fields.

Many of the diagrams in this paper are screen shots from a commercial attack tree software tool called Secur*IT*ree®. Secur*IT*ree, a commercial product of Amenaza Technologies Limited, implements the modeling functions described in this paper[3].

Basic Attack Tree Concepts

Attack Tree Origins

Attacks can be modeled using a graphical, mathematical, decision tree structure called an *attack tree.* There is evidence to suggest that *attack trees* originated in the intelligence community[4]. At least one intelligence agency is believed to have used tree-based attack modeling techniques in the late 1980s. In 1991 Weiss published a paper[5] describing *threat logic trees.* In 1994 Amoroso[6] detailed a modeling concept he called *threat trees.* It appears that the different groups may have worked independently. Bruce Schneier[7] (a noted cryptographer and security expert)

---

[3] The author has been the chief technical architect of the Secur*IT*ree software.

[4] In the 1998 paper by C.Salter, O.S. Saydjari, B. Schneier and J. Wallner, Toward a secure system engineering methodology, Proceedings of the New Security Paradigms Workshop, ACM Press, September 1998, two of the authors are shown as working for the National Security Agency and a third for DARPA.

[5] J.D. Weiss, A System Security Engineering Process, Proceedings of the 14th National Computer Security Conference, 1991.

[6] Edward G. Amoroso, Fundamentals of Computer Security Technology, pp 15-29, Prentice-Hall, 1994, ISBN0131089293

[7] 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

popularized the idea and the term *attack trees*. Other researchers have continued to develop the idea of tree-based, threat analysis models [8, 9].

Prerequisites of an Attack

Three conditions must be present in order for an attacker (also known as a *threat agent*) to carry out a successful attack against a defender's system.

1. The defender must have **vulnerabilities** or weaknesses in their system. Of course different resources are required to exploit different vulnerabilities.
2. The threat agent must have sufficient **resources** available to exploit the defender's vulnerabilities. This is known as **capability.**
3. The threat agent must believe they will **benefit** by performing the attack. The expectation of benefit drives **motivation.**

Condition 1 is completely dependent on the defender.

Condition 2 involves the interplay between the defender and the attacker. Whether condition 2 is satisfied depends on both entities. The defender has some control over which vulnerabilities exist in their systems (and the level of resources required to exploit them). Different threat agents have different capabilities. If an attacker possesses sufficient resources to exploit all of the vulnerabilities associated with a given attack it means they find the attack feasible.

Condition 3 mostly involves the nature of the attacker. If the attacker finds the results of a successful attack beneficial, they will be motivated to carry out the attack. Conceivably, the defender could contribute to an attacker's motivation if they do something to provoke the threat agent.

The threat agent and the defender interact to jointly determine whether an attack will occur. It is the combination of *feasibility* (as determined by conditions 1 and 2) and *desirability* (determined by condition 3) that provides insight into the likelihood of an attack. Understanding these factors also provides insight into effective ways of preventing attacks.

Attack Tree Vulnerability Models

Attack trees are best constructed from the point of view of the adversary. Creating good attack trees requires that we *think like an attacker*. We do not focus on how to defend a system when we initially create the model. Instead, we think of what an attacker wants to achieve and ways to accomplish it. Later, we use the understanding we have gained about how a system's

---

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

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

[9] Shelby Evans, David Heinbuch, Elizabeth Kyle, John Piorkowski, James Wallner, "Risk-Based Systems Security Engineering: Stopping Attacks with Intention", November/December 2004, IEEE Security and Privacy

vulnerabilities are likely to be exploited to improve its defenses.

Like most mathematical tree models, attack trees are represented by a diagram with a single *root* node at the top. The root branches downwards via children nodes, which in turn repeatedly fork and branch to their own children. This is similar to the *decision trees* often used to help with business decisions or the *fault trees* used to understand the reliability of machines and machine-like processes.

Significance of the Root Node

In an attack tree model, the topmost (*root*) node represents an objective that would be beneficial to one or more threat agents. The reason why we care about the fulfilment of these objectives is because attaining the root goal usually brings negative consequences to the defender[10].

It is often said that the root node in an attack tree represents what the attacker wants to make happen, and what the defender wishes to prevent. Loosely speaking, that may be true. But this is an over simplification of the situation.

In a zero-sum situation (where the attacker's gain is the defender's loss) then both points of view may be fully captured or represented by a single root node. However, in many cases, the defender's loss is incidental to what the adversary is trying to accomplish. For instance, a mugger may injure or kill a victim in the course of a robbery. That is not their objective but simply a side effect of attaining their objective (obtaining money). The attacker sees the tree's root goal to be *Obtain Money* whereas the defender might show the root goal as *Attacker Injures Victim*. Both points of view may be valid depending on what the analysis is trying to accomplish.

In a classic attack tree, different adversaries might have different goals, requiring distinct attacker-specific attack trees. For instance, one threat agent might be trying to attack an organization's computer systems in order to steal intellectual property to create a competing product while avoiding the development costs. In this case stealth would be a virtue because the defender might not ever realize their design had been stolen. Another type of attacker might be trying to ruin the system owner's reputation by causing a very visible attack (that does not disclose any trade secrets). These would be very different attacks.

Fortunately, by associating attacker and victim impacts with intermediate nodes in a tree, it is usually possible for a single attack tree to represent the different attackers' goals and outcomes. This will become clearer later when we discuss impacts. For the present we will only state that different goals influence the path taken en route to the root node. Only in exceptional situations are multiple attack trees required to carry out the complete analysis of a particular system.

Of greater relevance is the issue of *scope*. The selection of a root node goal implies the scope of the analysis – and greatly affects the accuracy of the results.

Consider the tragic events of September 11th, 2001 wherein terrorists flew commercial aircraft

[10] If the defender suffers no negative consequences from an attack, there is no reason to spend effort to prevent it.

into the World Trade Center skyscrapers in New York City and the Pentagon in Washington DC. What was the attacker's overall goal? Was it to hijack an aircraft? To destroy buildings? To create terror? Was it to use terror to influence geopolitical events? Or, in a more metaphysical sense, was it to gain eternal glory by self sacrifice for what the attackers believed to be a greater cause?

It appears that the attackers (or their leaders) believed that decadent western philosophy was creeping into regions normally under their control. Concepts such as women's rights, freedom of religion and so forth threatened their ability to control their population. They also believed that western societies were easily intimidated and that bloodying the nose of the leader of the free world would lessen western involvement in their region and ensure the continuity of their extremist regime. Accordingly, they created a plan using aircraft to eliminate western meddling in their affairs.

Arguably, it was the defenders' lack of understanding of these goals that resulted in the implementation of inappropriate security controls.

The aviation industry was very familiar with threats against aircraft due to the frequent aircraft hijackings carried out in the late 1960s and early 1970s. Hijackers frequently took control of aircraft bound for Miami and forced the pilots to fly to nearby Cuba. In 1969 alone over 30 such hijackings occurred.

The main outcome of these attacks was to ferry the hijacker to the nearby socialist utopia, Other than some embarrassment, minor financial losses and delays to schedules, there were very few instances in which passengers or flight crew were injured. This was not surprising – no rational attacker would dare harm the flight crew or their own personal safety might be in peril – or so it was assumed.

The industry had also experienced attacks involving bombs planted in the luggage or cargo compartments of aircraft[11] and screening procedures had been implemented. However, at least some of the bomb attack security measures were again predicated on the assumption that no rational attacker would detonate a bomb if they themselves were onboard the aircraft.

This understanding of the potential threats shaped the aviation industry's response. Protocols were implemented ensuring that no baggage was ever placed on a flight unless the passenger who checked the item was also onboard. Metal detectors were used to screen large metal objects (like guns) but protocols were not intended to block a small pocketknife or box cutter, for example[12]. Pilots were instructed that if someone did enter the cockpit and threaten harm they were to comply and fly the plane wherever the intruder demanded.

Focusing for the moment solely on the hijacking threat, the attacker's strategy might have been depicted in an attack tree with the attacker's root node goal being *Hijack Plane to Cuba*. The

[11] The 1985 bombing of Air India flight 182 killed 329 persons. In 1988 Pan Am flight 103 was similarly destroyed killing 270 people.

[12] The author remembers routinely flying with a small Swiss Army knife in his pocket in the 1990s.

tree would describe the various ways in which the hijacker might accomplish this goal. The defender's countermeasures would have attempted to prevent the attacker from attaining this root goal (by preventing passengers from bringing firearms onboard) or to limit the negative effects if they reached the cockpit (by quickly complying and flying to Cuba).

It cannot be denied that these measures were quite effective over a twenty year period. By the 1980s, diversions of American aircraft to Cuba were almost non-existent. Presumably erstwhile hijackers discovered that they could simply travel to Toronto and take a commercial flight to Havana.

Unfortunately, the defenders failed to anticipate the emergence of a type of attacker with a more sinister objective. They left themselves vulnerable to a far more serious type of attack – an attack by someone willing to use an aircraft as a human guided missile. An attacker so philosophically committed that they would willingly give their lives for their cause (and ostensibly earn themselves a pleasant spot in the hereafter).

The attack tree model for this type of attacker might have had as its root objective *Prevent Influx of Liberal Western Ideas into Afghanistan.* Various strategies might be represented as subgoals, including *Create Terror in Populace, Destroy Landmark Buildings* and *Kill 1000s.* Finally, (and various levels below) would appear one of the means of accomplishing these higher goals – *Hijack Airplane*. Due to the lack of understanding of the higher level goals the defenders could not foresee the type of attack their systems would be subjected to. And so their security controls failed.

Of course, it is easy to point out these deficiencies in hindsight. At the time, we could imagine how management and decision makers would have responded if the aircraft security team had brought them an analysis that involved geopolitical strategy. The report would likely have been rejected and the top dogs given cause to wonder how a bunch of security professionals came to involve themselves in affairs of state.

This highlights a dilemma faced by security analysts. If they create models focusing only on their stewardship the analysis is likely to be well received by decision makers – but may overlook attacks that are not obvious without a more complete understanding of potential adversaries' high level goals. Models that attempt to contemplate hypothetical situations well above the stewardship may be disregarded.

There is no perfect answer to this question, and it will depend somewhat on the organization's culture and willingness to think outside the box. In the author's experience, if an analyst places the root node of their model one level higher than what management requested, the analyst will be seen as having initiative and attention will be paid to the study's conclusions. However, if the analyst attempts to place their root node two or more levels higher than requested the analyst will be seen as a dreamer and the results of their study will not be taken seriously. Of course, the failure to place the root node high enough increases the chances that the results will not match reality – and when that happens management will come looking for someone to blame.

Once the root node goal is chosen, the scope of the model is largely defined. The root goal becomes the starting point of the study. By itself, the root goal is so lofty or broadly stated that it

lends little understanding as to how it might be achieved by the adversary. It is necessary to break the high level root goal into smaller, more manageable steps.

Boolean Logic in Attack Trees

A number of different strategies could be pursued by the adversary to achieve their overall goal. These strategies can be expressed as a series of intermediate objectives that singly, or in combination, realize the root goal. This decomposition process continues, breaking the intermediate goals into ever finer grained activities. This is conveniently represented using a graphical format and familiar Boolean algebra symbols (see **Figure 1**).

The topmost symbol in the tree represents the adversary's overall goal. It is referred to as the *root* node of the tree. The *root* in this particular example is depicted by a green, pointy-topped symbol  . The diagram shows how high level goals decompose into increasingly precise subgoals as we descend through the tree.

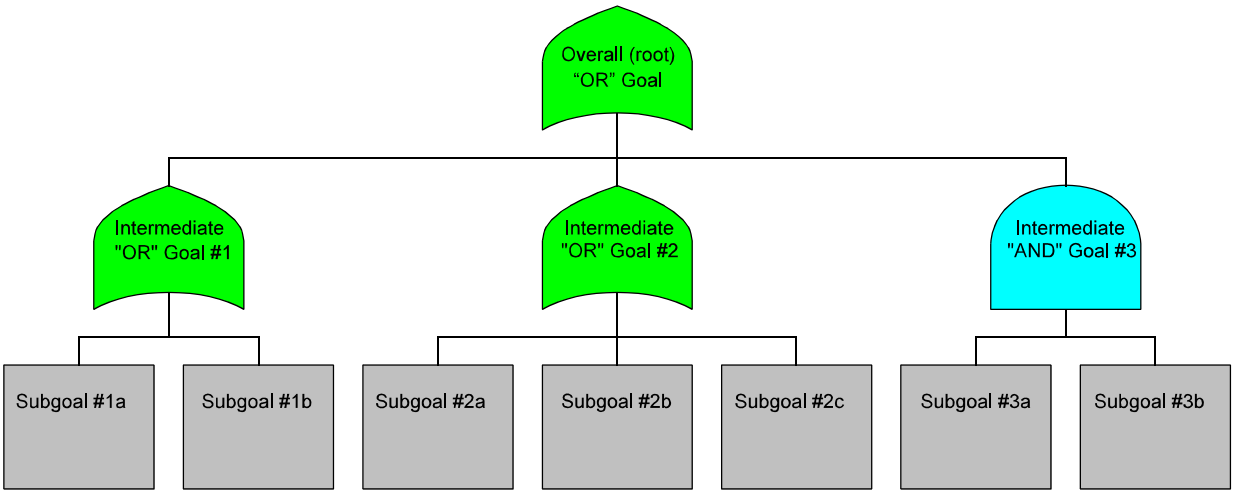


Figure 1 – Goal Oriented Tree

The *OR* symbol  (whose shape should be familiar to readers familiar with Boolean algebra) indicates that the root *Overall "OR" Goal* can be attained by achieving *Intermediate Goal #1 OR Intermediate Goal #2 OR Intermediate Goal #3.* Since there are usually a variety of different approaches to achieving the root goal, most trees' root nodes will be an *OR*. Whether at the root node, or lower in the tree, children of *OR* nodes describe the alternative ways in which an OR subgoal can be realized.

In this example, the *OR* nodes in the figure are further decomposed into rectangular shapes, called *leaf* nodes. For example, *Intermediate "OR" Goal #1* is achievable by attaining *Subgoal #1a OR Subgoal #1b. Leaf* nodes represent atomic activities which require no further decomposition to be understood. They represent exploits that could be performed by an attacker.

Intermediate Goal #3 is represented by a cyan *AND* symbol . This indicates that both *Subgoal #3a AND Subgoal #3b* must be completed in order to attain *Intermediate Goal #3*. The children of *AND* nodes represent a series of steps in a process or procedure that must be performed in order to attain or satisfy the *AND* node. Strictly speaking, the order of the *AND*'s children has no significance. However, a useful convention is that, if the sequence of operations is important to the attainment of the *AND* node, then the children should be arranged in stepwise order from left to right.

Node Labels

Although the labels assigned to a tree have no relevance to the underlying mathematics, they do affect human understanding of what the nodes represent. The following suggestions may provide guidance.

Because the low level *leaf* nodes represent activities performed by the attacker, it is helpful to adopt a *verb-noun* format. For example, *Pick Lock* or *Transmit Selected Sequence of Bytes*.

AND nodes can be thought of as describing processes or procedures, the steps of which are the *AND* node's children. Again, the *verb-noun* format is a good choice. For example, *Elevate Privilege*. In some cases it is also valid to label an *AND* node with a title that describes the state or condition that will be realized if all of the child operations are performed. In this case the node name might be *Privilege Escalation*.

OR nodes represent different ways of arriving at some state or condition and their labels should suggest that – although sometimes the *verb-noun* format is also acceptable.

A Sample Attack Tree

To illustrate the concept of a **capabilities-based attack tree**, let us imagine a hypothetical system we are trying to defend. Consider the home security challenge faced by the residents of a typical, suburban home who are concerned about a rash of home burglaries that have occurred in the neighborhood. The subject of the example was chosen for its simplicity and familiarity to

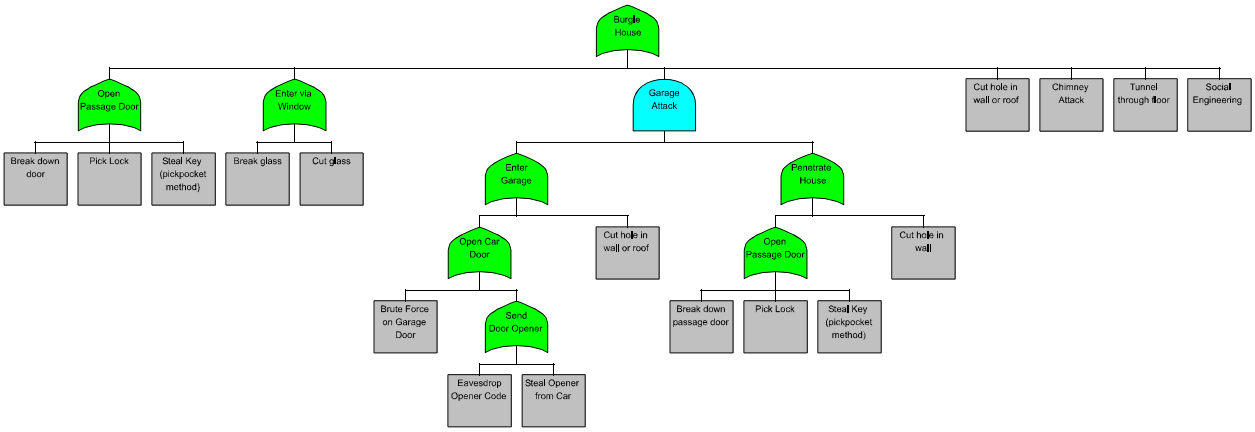


Figure 2 – Approaches to Burglarizing a House

readers. More interesting examples, particularly those involving information systems, are

frequently too large and complex to fit on a single page.

The house we have in mind is a middle-class dwelling, complete with an attached garage. The issue that concerns us is the possibility of the house being burglarized (see **Figure 2**).

Note that the structure of the top levels of an attack tree implicitly define a *taxonomy* or classification scheme for attacks. Different analysts may structure the same problem in different ways. Although the alternate representations may all have validity, some are better than others. Experience has shown that when higher levels representations of the tree reflect the system's architecture that it is easier to understand and extend the model. As will be discussed presently, this is especially important when impacts are added to models.

When first constructing the house burglary tree, the author's lack of expertise in the field of physical security caused him to consult with an expert. He contacted someone in U.S. Special Forces. When confronted with the problem of burglarizing a house, the specialist indicated that there are only three ways into any building: through the roof, through the walls or through the floor. "After that," the operative said, "it is just detail". This illustrates how it is helpful to understand the basics of a system's architecture before creating an attack tree model.

After some consideration, we can think of seven approaches the thief might use to enter the house and commit burglary:

1. Passage doors (i.e., the front and back doors normally used for entry).
2. Windows.
3. 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 to the attacker).

With the possible exception of the *Social engineering* approach, all fall into the basic categories outlined by the special forces person.

These attacks, which have been partially decomposed into more detailed steps, are shown graphically in **Figure 2**. 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 **Figure 2**, 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 higher level events into smaller, more precisely defined events could continue

almost indefinitely. For our purposes, it need only 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 using a rock. On the other hand, the leaf node that deals with *Eavesdrop opener code* ought to be decomposed into smaller steps to enhance the analyst's understanding of the actions to be performed by the burglar. We have not done so for reasons of brevity.

It is important to note that the adversaries' interaction with the system they are attacking takes place entirely at the leaf nodes. For that reason, some people call the leaf nodes *attack stabs* or *exploits*. All of the higher, non-leaf nodes in an attack tree represent logical states that the attacker achieves by performing one or more leaf node operations that satisfy the tree's logic.

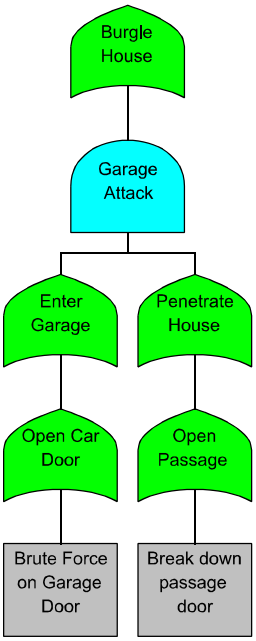


Figure 3 – Attack Scenario Example

Attack Scenarios

An attack tree shows a logical breakdown of the various options available to an adversary. By performing the exploits associated with one or more leaf level events which have been carefully selected to satisfy the tree's *AND/OR* logic, the attacker can achieve the root level goal. Each minimal combination of leaf level events is known as an *attack scenario.* The combination is minimal in the sense that, if any of the leaf events are omitted from the *attack scenario*, then the logic will not be satisfied and the root goal will not be achieved.

Associated with each *attack scenario's* set of leaf nodes is the collection of intermediate nodes that are activated along the path (or paths) to the root goal. Strictly speaking, these intermediate nodes are not part of the *attack scenario,* but it is useful to include them in graphical depictions to illustrate the logical states that will be achieved as the attack takes place. As will be seen shortly, negative impacts are often felt by the victim, and positive rewards by the attacker, when the intermediate *AND/OR* states are achieved.

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, capabilities and motivations. One particular attack scenario from the house burglary tree is shown in **Figure 3**. It consists of two leaf level events: *Brute Force on Garage Door* and *Break down passage door*. Both events are required to satisfy the *AND* node (*Garage Attack*) several levels above.

Combinatoric Explosion

One of the virtues of the attack tree is its ability to capture and depict a potentially large number of attack scenarios in a relatively compact diagram. One of the drawbacks of the attack tree is

that a relatively compact diagram can generate an extremely large number of attack scenarios[13].

The number of scenarios leading to an *OR* node is simply the sum of the number of scenarios leading to each of the *OR's* children (which may be subtrees). The problem comes with the *AND* nodes. The number of scenarios leading to an *AND* node is the product of the number of scenarios of each of the *AND's* children. So, for example, an *AND* node with five child subtrees (each of which having only 15 scenarios) would have 15^5 or 759,375 scenarios. Clearly, the potentially huge number of scenarios poses a significant challenge for analysis – even when the analysis is being performed by a computer. Fortunately, as will be discussed later, there exist heuristic techniques for quickly identifying and eliminating scenarios that do not contribute meaningful risk to the situation.

Behavioral Indicators

To this point, the attack tree shows how attacks could occur, but provides no indication of their likelihood. Intuitively we know that most burglars prefer breaking a window to digging a tunnel underground. We suspect this is because it is less work to break windows than to dig tunnels. It seems reasonable to suppose that the amount and type of resources required to perform an attack affect the behavior and choices of an adversary.

Since all of the direct interaction between the adversaries and the defender's system occurs at the leaf nodes, it is useful to associate values with each leaf node operation describing the resources the leaf nodes require from the adversary. The types of resources examined are chosen to be factors that influence the behavior of the adversary. For instance, the monetary cost, technical ability, time and noticeability of an exploit all affect an adversary's ability to perform the exploit. Values for these parameters are obtained from subject matter experts (SMEs) who provide estimates based on their expert understanding of the activities.

Metrics and Ratings

One of the goals of attack tree analysis is to provide an objective framework for assessing the likelihood of attacks and the risk associated with them. In an ideal world, the *behavioral indicators* chosen in our models would involve *metrics*.

The word *metric* has Greek origins, and means *to measure*. Measurements are (or should be) an objective quantity. That is, two people independently making a measurement should, within an uncertainty value, arrive at the same figure. Of course there are issues associated with measurements (e.g., systemic error in the measuring instrument) but within those bounds measurements are not subjective. A characteristic of a behavioral indicator being a metric is that it will have a numeric value and a unit.

Monetary cost is an example of a behavioral indicator based on a metric. If a leaf node operation requires the attacker to acquire a specialized piece of equipment, then it is possible to assess the cost of the device by visiting several stores and comparing prices for the different models

[13] The author is aware of an actual attack tree used in the field of critical infrastructure that involved tens of billions of attack scenarios!

available. The variations in pricing constitute the uncertainty in the measurement – but the basic assessment of cost will be objective. Two individuals using the same measurement protocol will arrive at similar cost values. In this case our indicator passes the test of being a metric because the cost value has both a number and a unit ("$," or whatever currency is appropriate).

In a perfect world, all of the behavioral indicators in attack tree models would be metrics. Unfortunately, we do not live in a perfect world and not all of the factors that affect adversary behavior are easily measurable. In these situations, we are forced to include *ratings* in our analysis. Ratings are estimates based on subject matter expert opinion.

Subject matter experts accumulate understanding based on decades of experience. This allows them to *rate* things that may be impossible to measure. These ratings are subjective, and the assessed values will differ from expert to expert. Nonetheless, it seems unwise to simply discard these opinions completely simply because of the subjectivity. The experts' experiences and judgement have to count for something.

For instance, some cryptographic systems use keys that are based on a very large number that is itself the product of two large prime numbers. Determining the two large prime factors (given the extremely large product) is a difficult mathematical problem. Mathematicians have been searching for an efficient way of factoring numbers for many years – so far no such algorithm has been identified.[14] The question is, how would a subject matter expert rate the difficulty of such a problem?

The expert would likely first establish a scale. The actual scale is arbitrary. It could range from 1 to 10, 1 to 100 or any other scale that is convenient. Ratings usually have no units associated with them – just raw numeric values.

In constructing a scale, sample values are identified across the range. For example, if an analyst were constructing a *Technical Ability* behavioral indicator rating related to information technology exploits, they might propose a 1 – 100 scale such as

1 - 10	Warm body (no technical skills whatsoever)
10 - 20	Average office employee (computer user)
20 - 30	"Power" user. Also, script kiddie
30 - 40	Professionally trained in IT
40 - 50	Senior IT person. Programmer, senior network administrator
50 - 60	Senior IT person with research facilities
60 - 70	World class expert
70 - 80	Practically impossible, theoretically possible
80 - 90	Believed to be impossible
90 - 100	Demonstrably or provably impossible

[14] Of course, it is possible that some government intelligence agency has discovered an efficient factoring algorithm and kept it secret. Such things have happened in the past. However, in many cases, the persistent efforts of other researchers (operating in unclassified environments) eventually re-discover the algorithm. Such was the case with public key encryption.

Then, if asked to assess a rating value for the task of efficiently deriving the prime factors of a large crypto key, they would refer to the sample table. At a minimum, the task falls in the 70 to 80 range. Many mathematicians believe that no algorithm exists – even though no mathematical proof has been presented to prove it. So, a subject matter expert might even assign a *Technical Ability* rating value of between 80 and 90 to such a leaf node task. Again, there might be some uncertainty in the estimate, but every expert would likely agree that such a task would be far beyond the capability of a *Senior IT Person* (technical ability of 40 to 50 on the scale).

It should also be noted that both metric and rating values often change with time. In the 1990s a network packet scanner cost around $50,000. Today, software to make any laptop into a comparable (or better) scanner is available for free. Similarly, the prime factor technical ability rating value we suggested above is expected to change dramatically when quantum computing devices emerge. Initially a quantum computer capable of efficient factoring may be available only to a world class expert (rating value 60 to 70). Then, as production increases, such computing engines will become increasingly accessible and the technical ability required to use a commercially available device will likely decrease to the 30 to 40 range. This emphasizes how the development of attack trees, and the attack tree analysis process must be ongoing and updated as changes occur in the environment.

In the above examples, both metrics and ratings had numeric values over some range. It is also possible for capabilities to be Boolean in nature. For instance, certain attacks require physical presence. It is impossible to break down a door while sitting at a computer on the other side of the world. Therefore, creating a Boolean indicator called *Physical Presence Required* would help distinguish between leaf node exploits that could be performed remotely and those requiring physical proximity. This indicator would only have two possible values: true or false.

Attack Scenario Costs and Aggregation Functions

The above discussion of behavioral indicators described how indicator values are assigned to the leaf nodes in a tree. In all but degenerate cases (involving very poor security architecture), the execution of a single leaf node exploit does not result in a successful root-level attack. Achieving a root level attack requires that a complete set of leaf level activities in an attack scenario be performed. The overall cost of any given attack scenario can be calculated by aggregating the resource requirements of all of the scenario's leaf level activities. The aggregation function used depends on the nature of the indicator.

If the exploit consumes a resource then the total requirement for that resource is the sum of the scenario's leaf nodes' resource metrics. *(Monetary) Cost* is a good example of this. Consider a scenario that has multiple leaf nodes, two of which involve a task that requires the purchase of a specialized piece of equipment. If it is assumed that the equipment vendor does not permit returns for refund on the equipment, then the overall cost of performing the two leaf nodes is the sum of each individual leaf node's cost.

On the other hand, an indicator such as *Technical Ability* allows the adversary to reuse their skill multiple times. So if the technical ability rating of two leaf nodes in an attack scenario were evaluated it would be the maximum of each of the scenario's leaf node technical ability ratings.

In certain cases the aggregation function or a particular behavioral indicator may vary by circumstances. One interesting capability indicator is *Time*, or, *Time to Perform Exploit*. Depending on the situation, the operations described by the children of an *AND* node might be performed sequentially or concurrently. The overall time to achieve the *AND* node would be the sum of the individual operations for sequential tasks but the maximum of the individual operations for concurrent tasks. Other aggregation functions are possible.

Since the aggregation functions provide a mechanism for estimating the resources required to perform multiple leaf level operations, they implicitly apply only to *AND* nodes. *AND* nodes are the mechanism in an attack tree that allows (and requires) multiple actions below them.

When the set of attack scenarios for an attack tree is generated (showing the combinatoric paths to root) any attack scenario involving *OR* nodes will show only a single child for each of the *OR* nodes. In fact, this is really what it means to compute the set of attack scenarios; to explore each alternative way of achieving *OR* node goals. This means that aggregation functions are only needed for *AND* nodes. In an attack scenario, *OR* nodes are simply passed indicator values from their only active child.[15]

Pruning – Eliminating Attack Scenarios Based on Infeasibility

Whether or not a system's defenses are adequate to thwart an attack depends on whether an attacker has sufficient resources to perform all of the exploits required for a particular attack. If the attacker's resources are sufficient then the scenario is possible. If the adversary also has the desire or motivation to carry out the attack, then the attack is probable.

A simple way of evaluating the feasibility of an attack scenario for a given adversary is to compare the resources available to the attacker with the scenario's behavioral indicator costs. Those scenarios with resource requirements greater than the adversary's capabilities can be safely eliminated from consideration (since it is not possible for that adversary to provide them). The attacks that remain are feasible and, depending whether they are desirable to the *threat agent*, have some, non-zero level of probability. This process is known as *pruning.*[16]

For instance, a typical juvenile delinquent might only have $50 available to spend on attacks, and possess limited technical skills. The cost and technical difficulty of digging a tunnel underground would eliminate the tunneling burglary scenario from consideration by juvenile delinquents.

Pruning provides a defender with a quick estimate of the magnitude of their security problem by eliminating from consideration those attack scenarios that are beyond the capability of a

---

[15] This is different from the mathematics used in *fault trees*. *Fault trees* are used to describe failure conditions resulting from independent stochastic (random) events. Given the appropriate statistical data it is possible to calculate not only the probability of reaching nodes in the tree via a particular scenario, but also the overall probability of reaching a given node from all scenarios leading to it. This is not possible in an attack tree because the leaf level events are highly interdependent. And, as mentioned earlier, even if the statistics are not usually available or applicable.

[16] See also, Computer Security Journal, Volume XX, Number 2, Spring 2004 pp 33 - 59.

particular threat agent. Unfortunately *pruning* is overly simplistic. It treats each of the attacker's resources in isolation whereas it is more realistic that an adversary would consider the combined cost of all of the resources required for an attack.

Pruning is also a binary operation – scenarios are either in or out. Even the slightest amount of uncertainty in the resource cost of a leaf node or the capability of an adversary could result in a scenario of concern being eliminated.

Finally, the amount of resources the adversary is willing to spend depends, in part, to the extent that an attack scenario satisfies their goals and ambitions. The effects of varying degrees of motivation are not captured by pruning.

Still, pruning is a useful "quick and dirty" type of analysis. Even if it does not provide a complete assessment of attack scenario likelihood, it does eliminate scenarios that are infeasible for an adversary regardless of their degree of motivation. It is not unusual for 75% of the hypothetical attacks to be pruned away as impossible. Any defender should find it easier to deal with ¼ of the potential attacks!

Impacts

Attack Scenario Risk Requires Victim Impact

If it is broadly assumed that all feasible attack scenarios (those remaining after pruning) have non-zero probability, that is still only half of the risk equation. Hostile risk is generally accepted to be the combination of two factors:

$$\textit{Attack Risk} \equiv \textit{Attack Probability} \times \textit{Victim Impact}$$

In order to fully understand risk, our model needs to include the impact each attack scenario will have on the defender. This can be achieved by a simple extension to the attack tree model.

Recall that the behavioral or capability values discussed previously are entered at the leaf nodes of the tree, and their overall contributions to the capability costs of the attack scenarios are calculated through the use of aggregation functions. This reflects the fact that all of the adversaries' interactions with the target occur solely at the leaf level.

Impact, however, can occur at any level in the tree[17]. While it is true that the victim may suffer some impact when an attacker performs a leaf level exploit, the leaf level impacts are typically minor compared to the impacts that result at higher levels in the tree. The greatest victim impact is often (but not always) at the root of the tree.

For example, a physical attack may allow an intruder to access a computer server room by breaking down a door (and result in $300 of damage). Once inside, the attacker can easily steal a disk drive, again creating a small financial loss to the victim (perhaps $200). If the attacker were

[17] In a classic attack tree, all of the impact implicitly occurs at the root node. The consequence of this is that all attack scenarios are equally desirable to an attacker and equally undesirable to the victim. Clearly this is an oversimplification.

a common thief and only interested in stealing goods then the total financial impact on the victim would simply be $500.

Unfortunately for the victim, the *Force door* and *Steal disk drive* leaf nodes are children of an *AND* node labeled *Obtain valuable intellectual property*. Performing the two leaf operations results in the attainment of the *AND* node and combine to potentially cause an impact of $1M.

Whereas the value of behavioral capability indicators is calculated mechanically for each scenario using an algorithm or formula to aggregate the scenario's leaf level resource costs, this is not sufficient or appropriate for impacts.

As noted, the *Force door - Steal disk drive* attack scenario does involve $300 + $200 = $500 of property loss and damage. But of much greater importance is the loss of the million dollars of intellectual property (which occurs when the *AND* state is realized). To paraphrase a quotation attributed to Aristotle, the *AND* is greater than the sum of its leaf nodes! In order to properly capture the magnitude of this loss it is necessary for the analyst to associate an impact value with the *AND* node that reflects the anticipated business loss. This inserted value might be used in conjunction with a calculated value (e.g., $1M + $500) or it might simply replace the computed value (which is insignificant in comparison). Inserting these values is only possible because of the business or situational knowledge of the analyst. The participation of a skilled analyst and the incorporation of a system's business knowledge are essential to the creation of high fidelity attack tree models!

The pruning technique can incorporate impact. If you accept the simplification that all attacks remaining after pruning are of comparable non-zero probability, then for those scenarios the risk equation can be simplified to

$$Attack\ Risk \propto Attack\ Impact$$

However, even if you accept this questionable approximation, there is no obvious process for deriving a single value for an attack scenario's impact term if the model incorporates multiple victim impacts (e.g., loss of money, deaths, damage to the environment).

Clearly, our model must become more sophisticated to properly deal with the shortcomings we have identified.

Attack Scenario Probability and Attacker Psychology

The pruning technique shown earlier provided only the coarsest estimate of likelihood, only distinguishing between completely infeasible (and therefore impossible) and feasible. Surely different attack scenarios have a range of probabilities.

Attack tree modeling is essentially an attempt to predict and model human behavior – certainly an ambitious undertaking. Many factors might be incorporated into such a model. The analysis discussed in this paper will attempt to predict an attacker's behavior by focusing on two factors: *capability* and *motivation*. It is our hypothesis that

IF they want to AND they can THEN they will

In other words, if an adversary believes they will gain something by carrying out an attack scenario (the *want to* factor) and the attack is within their capability (*they can*) then the likelihood of the attack will be considered to be high.

The pruning technique described earlier is a primitive mechanism for assessing capability. As we continue, a better approach will be presented that will provide a scale of feasibility.

The *want to* factor represents the attacker's level of desire – their motivation for carrying out the attack. Attacks that are highly desirable (from an attacker's perspective) will convince them to spend more of their resources than attacks that are less desirable. Highly desirable attacks are more likely.

In the same way that victim impacts were inserted at strategic points in an attack tree to reflect the amount of loss that would be suffered by a victim, a different set of impacts can be associated with attack tree nodes representing the benefits (and possible detriments) that an attacker may obtain depending on whether their chosen attack scenario traverses the beneficial nodes.

Our model of human (attacker) behavior assumes that, given two attack scenarios of similar or equal feasibility, the attacker will choose the attack that brings the greatest reward.

Objective Analysis of Subjective Human Traits

Every day, people (good and bad) are faced with choices and consequences. It is our hypothesis that people generally select one activity over another because they believe that it has a superior cost-benefit[18] ratio to the competing alternatives. However, it is not enough to analyze the raw costs associated with their choices. Our models must reflect the fact that different people, and particularly different classes of people (whether attackers or defenders), perceive the same things as having different values.

Attacker Behavior

The simplistic binary pruning operation described earlier needs to be refined to provide a range of feasibility values for each adversary and the set of attack scenarios under consideration. In order to understand attacker behavior we need to see attacks from their perspective.

Feasibility of Attack

Every attack requires the adversary to expend a set of resources. The analyst selects specific types of resources to be included in their attack tree model based on the degree to which they influence the adversary's ability to perform the various attack scenarios. These resources might include money, raw materials, talent, time and a willingness to be noticed.

Even though everyone might be forced to spend the same amount of a resource to perform a specific attack scenario, that does not mean that they are equally willing or able to do so. The

[18] Although common vernacular speaks of the cost-benefit ratio, generally it is calculated as $\frac{Benefits}{Costs}$. The greater the benefits (and the lower the costs) the higher the value. Costs and benefits do not have to be monetary.

availability of resources varies. For instance, a relatively poor juvenile computer hacker (i.e., a script kiddie) might consider $100 to be a lot of money, and be strongly disinclined (or unable) to part with it. On the other hand, a busy executive in a large company might regard $100 as pocket change. However, the time-crunched white collar worker would be far less willing to part with 25 hours of his or her precious time than the bored adolescent who is happy to while away the wee hours trying to crack a computer system. In terms of time, the teenage script kiddie is

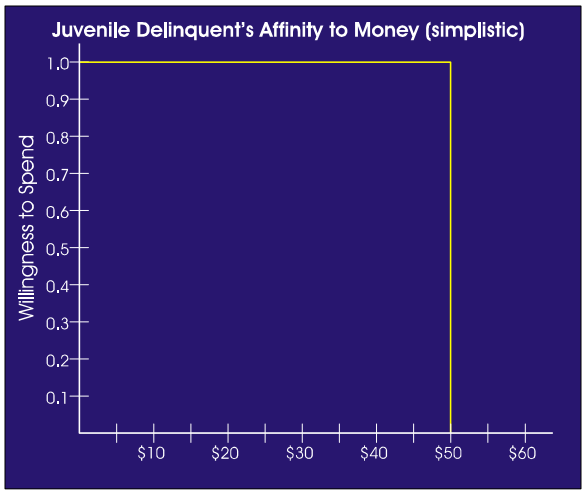


Figure 4 – Pruning-based agent profile

wealthier than the six figure executive.

The simple pruning mechanism described earlier provided a crude representation of the affinity of threat agents to their resources. For example, suppose the analyst created a profile of the juvenile delinquent threat agent that specified a financial limit of $50. This simplistic profile asserts that the juvenile delinquent is completely and equally willing to spend any sum between $0 and $50, but that they would be utterly unable or unwilling to spend $51 (as shown in **Figure 4**).

While we are unaware of any research into the spending habits of juvenile delinquents that would

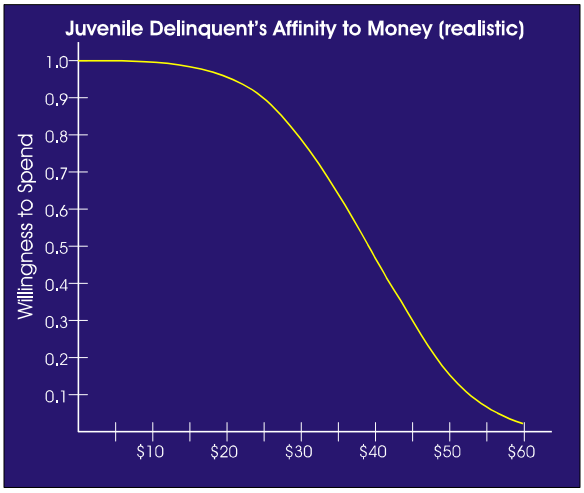


Figure 5 – Value-based agent profile

support the exact curve shown[19] in **Figure 5**, it is more plausible than **Figure 4**. Basic economics dictates that there is some scarcity to almost every asset, and a corresponding reluctance to spend all of it. We find numerous examples where people's willingness to spend decreases gradually (and monotonically), but very few situations where willingness is binary.

In general, people are always well disposed to spend none of a resource to acquire a desirable commodity. This is shown in **Figure 5** by the y-intercept point (1.0) on the graph. No one is able to spend more than they possess (no matter how attractive the goal may be). The limit of their resource is the x-intercept. The y-intercept point (1.0) is true for all adversaries and all types of resources. The x-intercept is established by research and intelligence data about the type of attacker in question.

We call functions that map a threat agent's attachment to quantities of the commodities required to perform an attack, *resource affinity utility functions*. The domain (x-axis value) of these functions is the resource under consideration. While we could choose any range for the y-axis values, for convenience we establish the convention that the range will be from 0 to 1.

If we simply drew a straight line between the y-intercept and the x-intercept it would likely be a better representation of the attacker's behavior than the step function shown in **Figure 4**. At least it would reflect the general principle of human nature that people prefer to spend less than more. However, the shape of the curve can be used to capture our understanding of the nature or psychology of the adversary.

A region in the curve that is near horizontal represents a zone over which the adversary is bold with respect to the resource. That is, they are only slightly less willing to part with the greater amount of the resource. Said another way, it won't take much extra reward to encourage them to spend the higher amount.

A curve region that is steep represents a zone in which the attacker is timid with respect to the resource. It means they value that amount of the resource highly and will require significant inducement to convince them to part with more of it.

Generally speaking, an adversary will be bold when their resources greatly exceed an attack's requirements, or they can easily replenish the resource. As the attack's requirements approach the limit of what the adversary is able to provide they become reticent to spend the final units.

Resource affinity utility functions can map the perceived value of any commodity for any adversary. For example, a utility function could be created to map the domain of raw technical ability rating values (arbitrarily chosen to span from 1-100) to an output range of 0 to 1 (**Figure 6**) for a juvenile delinquent (and similarly for their willingness to commit acts that might be noticed (**Figure 7**).

[19] Indeed, a perfectly accurate curve may be unattainable given that there will be variations in the behavior of individuals within the threat agent class of *juvenile delinquent* and variations in a particular individual's behavior from one day to the next..

In general, it is the combination of costs associated with a task that creates an overall perception

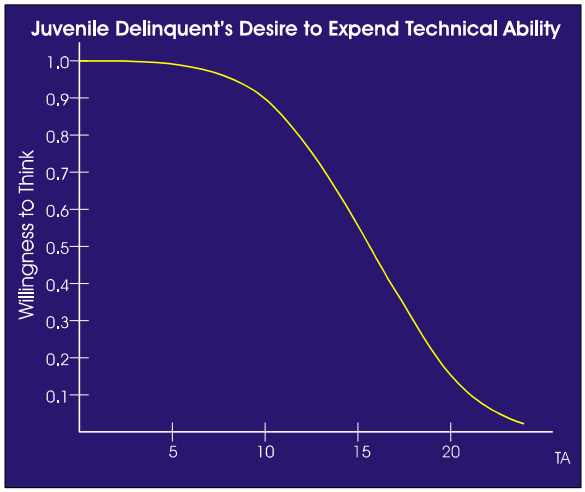


Figure 6 – Juvenile Delinquent's Technical Ability Utility Function

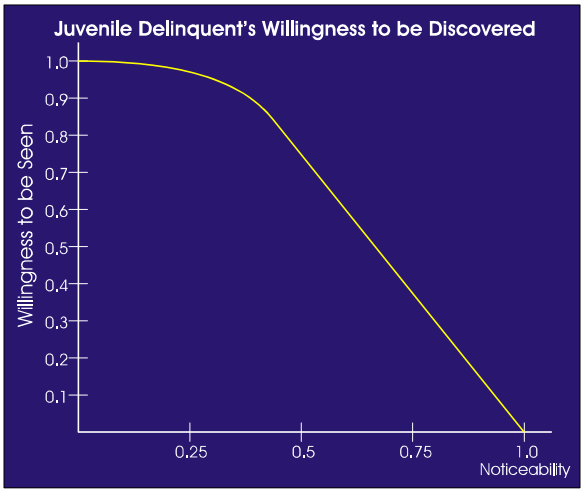


Figure 7 – Juvenile Delinquent Noticeability Utility Function

of feasibility. A good way[20] to estimate the overall difficulty of a specific attack scenario (as perceived by a particular threat agent) is to compute the product of the outputs of the threat agent's utility functions. Using a product formula to compute overall attack feasibility is a good choice because it correctly shows that the lack of even one of the resources needed to perform the attack will prevent it from occurring. That is, if the output for the utility function of even one of the resources in an attack scenario is near zero, then the overall product will also be near zero.

Consider the Juvenile Delinquent's *Cost of Attack*, *Technical Ability* and *Noticeability* utility functions (shown in **Figure 5**, **Figure 6** and **Figure 7**). Suppose that we wish to compare the desirability of various attack scenarios for burglarizing a house. Using the familiar *Burglehouse* attack tree model, we find that the *Break Down Door* attack scenario will cost the adversary $25 (to purchase a steel battering ram), require a *Technical Ability* rating of 10 (out of 100), and expose the miscreant to a 0.3 *Noticeability*. Using the utility functions shown, we discover that

$$f_{cost}(25) = 0.9$$
$$f_{tech\ ability}(10) = 0.9$$
$$f_{noticeabiity}(0.3) = 0.95$$

and therefore

$$\textit{Attack Feasibility} = 0.9 \times 0.9 \times 0.95 = 0.7695$$

---

[20] Initially, the use of a weighted sum was considered. For instance, if there exist three behavioral indicators, and the output of the utility functions for each indicator are *A*, *B* and *C*, then compute the overall value as

$$aA + bB + cC = \textit{Overall Difficulty} \quad \text{where } a + b + c = 1$$

The problem with this approach is that it is does not reflect the fact that the lack of a single resource is enough to prevent an adversary from carrying out an attack. I.e., the attacker's decision is constrained by *AND* node logic.

By comparison, the *Steal Opener from Car, Break Down Passage Door* attack scenario requires $30, a *Technical Ability* requirement of 10, and has a *Noticeability* of 0.28. So, the attack is slightly more expensive than the previous case, slightly less noticeable and requires the same technical ability. This yields:

$$f_{cost}\ (30) = 0.79$$
$$f_{\text{tech ability}}\ (10) = 0.9$$
$$f_{\text{noticeabiity}}\ (0.28) = 0.97$$

and therefore

$$\textit{Attack Feasibility} = 0.79 \times 0.9 \times 0.97 = 0.6897$$

If our assumptions are correct, this attack is slightly harder for a juvenile delinquent than simply *Breaking Down Door*.

One problem with assessing overall feasibility via a simple product formula is that the *Attack Feasibility* values for the scenarios decrease as additional indicators are added to the model. Each adversary under consideration will have a corresponding utility function for the new indicators. Even if a particular adversary has plenty of a every resource needed for a specific attack scenario, the output of each of the utility functions will be high but not quite 1.0 (say 0.9). Each such utility function correctly reflects that the adversary is capable of supplying that resource but when the several functions are combined through the product formula the overall feasibility drops.

For instance, if there were five different resource utility functions, each representing a resource that the adversary had in plentiful quantities (yielding utility function outputs of 0.9 in every case) the overall feasibility would drop to 0.9^5 or 0.59. The problem is exacerbated as the number of indicator functions in the model increases. This makes it difficult to compare feasibility values between models with few and many indicators. It also discourages analysts from adding indicators in the course of a project because of the resulting downward shift in the scenarios' feasibility values.

One solution to this problem is to use the geometric mean instead of the product. If there are n indicators in a model, and for each adversary there exists a set of utility functions, $f_1(x_1)$, $f_2(x_2)$, $f_3(x_3)$ ...$f_n(x_n)$, that map the raw amount of each attack scenario's required resources (x_1, x_2, x_3, ... x_n) to the attacker's ability to provide them, then the attack scenario's feasibility for that adversary is given by

$$\sqrt[n]{\prod_{i=1}^{n} f_i(x_i)}$$

This improved geometric mean approach yields *attack feasibility* values of $0.7695^{1/3} = 0.916$ and $0.6897^{1/3} = 0.577$ for the previous examples.

It is true that all of this analysis is based heavily on our choice of utility function curves. Those

curves are constructed partly on intelligence information and partly on assumptions or estimates we have made about an adversary's available resources. If these estimates are poor then the results of the analysis will also be incorrect[21]. Even when precise intelligence data is not available it is essential that at least the limits of the adversaries' capabilities be within order of magnitude correctness.

Choosing curves based on our assumptions about the way we believe that specific groups of people will behave is admittedly not perfect. However, it is certainly more accurate than the binary style of attacker model (**Figure 4**) used previously in pruning. Since the primitive model gave useful results, we expect that the use of a more accurate utility function will yield even better predictions. At the very least it expresses our understanding of the adversary, and exposes our assumptions and thought processes to review and discussion.

Due to variances in human behavior within a threat agent class, no curve will ever be a perfectly accurate description of a specific threat agent's decision-making process. While acknowledging the limitations inherent in this method of calculating *Attack Feasibility*, we believe that it is a useful representation of the ease or difficulty of an attack as perceived by the adversary.

It is sometimes convenient to speak of *Attack Difficulty* (as the opposite to *Attack Feasibility*) The two terms are (philosophically, if not mathematically) the inverse of one another:

$$Attack\ Difficulty = \frac{1}{Attack\ Feasibility}$$

Attacker Motivation is Related to Attack Benefits

Earlier it was suggested that adversaries make decisions on the basis of perceived *cost-benefit*. The calculation of the *Attack Feasibility* value considered the attack scenario costs but does not weigh the benefits the attacker expects[22] to gain from executing an attack scenario. These benefits must also be taken into account in order to understand the desirability of an attack to an adversary.

In the *Burglehouse* example discussed earlier, the attacker's benefits were primarily monetary. In more complex situations multiple types of benefits may accrue from carrying out attack scenarios. Adversaries will be attracted to specific scenarios depending on their nature and the particular combination of rewards. Different scenarios will provide different degrees of motivation to different attackers.

---

[21] An example that demonstrates the consequence of underestimating the resources of the adversary can be found in the cracking of the German Enigma device used during WWII. The German military correctly believed that breaking the enigma encryption purely by human reasoning would be impossible. No human, or even group of humans, could possibly perform all of the trial computations needed to find the correct Enigma rotor and plugboard settings. The Reich's error was in not realizing that the Allies (and Alan Turing) would apply mechanical and electronic computing devices to greatly increase the number of decryption operations possible in a time period.

[22] Attackers make their decisions based on their perceptions of the results of a particular course of action. They may, in fact, be wrong in their estimation. In this case, perception is far more important than reality.

For instance, a cat burglar and a juvenile delinquent have significantly different objectives in burglarizing a house. The cat burglar wants to steal items of high commercial value that are hard to trace and easy to sell. Cash is an obvious prize but jewels and some consumer electronics might also be attractive. The juvenile delinquent might very well have greater interest in a collection of baseball cards or comic books (although cash is always appealing).

Recall that all of the direct interaction between an adversary and their target is captured in the leaf nodes of the attack tree. However, many (and usually most) of the benefits an adversary gains from an attack accrue at higher, logical states in the tree[23]. Usually the greatest attacker benefits are associated with the tree's root node,[24] with additional side benefits occurring at the various intermediate nodes. Since different attack scenarios traverse different paths between leaf nodes and root, the attacker benefits may differ considerably depending on the attack scenario used. In the house burglary example, the dissimilar goals of cat burglars and juvenile delinquents affect the attack scenarios they will choose. A cat burglar would be unlikely to search for jewelry in the garage but the juvenile delinquent may guess or learn that a box of comic books is located there.

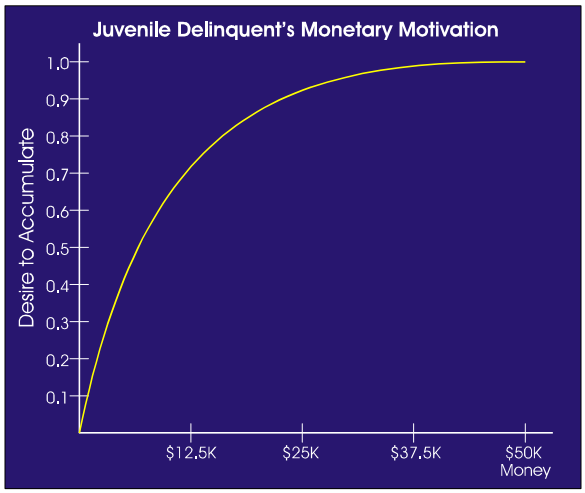


Figure 8 – Juvenile Delinquent's Desire for Increasing Quantities of Money

Most types of rewards exhibit diminishing utility. Even money loses its impact after a certain point. The rarity of billionaires is at least partially due to the fact that most multi-millionaires can't be bothered to continue accumulating wealth. There are very few indulgences available to billionaires that aren't available to hundred-millionaires. There's no point in having toys if you are too busy working to play with them!

Functions which map absolute amounts of the resource to the perceived value are called *attacker benefit utility functions.*

The attractiveness of a given reward is subjective and must be seen through the eyes of the adversary. A juvenile delinquent with a minimum wage job (or none at all) may view $50,000 as unlimited wealth. The diminishing *benefit* of wealth to a juvenile delinquent is shown in **Figure 8**. *Benefit* is a measure of the perceived value of a particular resource. In a juvenile delinquent's world, there is really nothing that cannot be bought with $50,000. Even the fanciest skateboard or video game console could be purchased for a few thousands of dollars. In fact, the juvenile delinquent's benefit function for money shows that they are almost as motivated by $25,000 as $50,000. Either sum would exceed all their desires.

---

[23] Just as the predominant victim impacts tend to occur at higher levels in the tree, so do many of the attackers' benefits.

[24] The root node of an attack tree is often chosen expressly because it is the point where an adversary attains their greatest benefits.

An industrial spy might have much higher aspirations than the juvenile delinquent. The intellectual property or company business plans they seek may be worth many millions of dollars. They may even feel that, below a certain threshold, a particular activity isn't worth their effort. The *attacker benefit* curve for such a person is seen in **Figure 9**. **Figure 9** shows that the Industrial Spy doesn't have much motivation to do anything illicit until the reward hits about $500K. Above $500K the desire increases rapidly until about $5M (at which point the industrial spy's greed is becoming satiated).

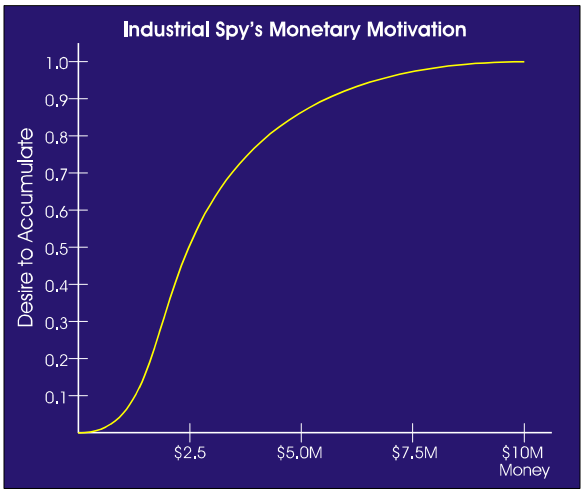


Figure 9 – Industrial Spy's Desire for Increasing Quantities of Money

As mentioned earlier, money is just one of several possible benefits to be garnered through an attack. Revenge, prestige, power and sundry gratifications of desires are all possibilities. These could be represented through other threat agent specific benefit functions (all of which would also yield values between 0 and 1).

Where multiple rewards exist it becomes necessary to combine the output of the corresponding *attacker benefit utility* functions. The multiplication technique used earlier to blend the *ease of attack* functions does not work well for combining the *attacker benefit utility* functions. If multiplication is used, a high score can only result if the attack benefits the attacker in every measurable way. This is not realistic. Even if an attack does not provide every conceivable benefit, an attacker may still find a subset of the potential rewards to be very attractive. For that reason, it is preferable to use a weighted sum to assess the combined value of the *attacker benefit utility* functions.

$$aA + bB + cC + ... + nN = Attack\ Desirability \qquad \text{where } a + b + c = 1$$

For example, suppose that the *Burglehouse* attack model incorporated an indicator that measured the amount of vandalism that resulted from a particular attack (slashing furniture, soiling carpets, breaking glass) and that the agent profile for a Juvenile Delinquent had a corresponding *attacker benefit* function that reflected the thrill that juvenile delinquents get from this type of activity. How might we combine the value of that thrill with possible monetary benefits?

If we believe that juvenile delinquents like money, but value mayhem and destruction even more, we might assign weighting factors of 0.4 for monetary gain and 0.6 for destruction. If the monetary reward for a particular attack yielded $15,000 then (reading from **Figure 8**) this gives a monetary *attacker benefit* value of approximately 0.78. Using the mayhem benefit function (not shown) we might obtain a value of 0.5 for that particular attack. Calculating a weighted combination of these values yields an *attacker benefit* of $(0.4 \times 0.78) + (0.6 \times 0.5) = 0.612$

It would be simplistic to believe that successful attacks bring only positive benefits to an attacker. Aside from the use of resources, an attack may have one or more detrimental effects on

the adversary. The adversary may face time in jail, injury or even death from carrying out the attack. By applying a similar technique to that used for *attacker benefits*, it is possible to calculate a weighted sum of *attacker detriments*. If the sum of *attacker benefits* and *attacker detriments* is positive, it means that the adversary`s overall perception of the attack is favorable and they will be motivated to do it (within their resource constraints). If the sum is negative, it means that the downside outweighs the benefits and they are repulsed from attempting an attack of that nature. To simplify discussions, we will usually speak only of an attack scenario`s *attacker benefits*, but it should always be remembered that this actually encompasses both the benefits and the detriments.

Zero sum games

It might be thought that security is a zero sum game. That is, the attacker's gain is exactly the same as the victim's loss. Although there may be certain cases where this is true, in general, it is not – for two reasons.

Even if the attack involves a direct exchange of some commodity (e.g., money) the attacker and defender may perceive its value differently[25].

More importantly, it is important to understand that attackers operate with their own interests in mind. Any negative effects on the victim are simply a side-effect of the attacker achieving their own goals. For instance, the house burglar may break a window in order to get inside to steal things. Nothing about broken glass motivates the attacker; it is simply a means to an end. The home owner does suffer a damage cost. But, unless explicitly the case, the damage is not the intent of the attacker.

Of course there are instances where one of the attacker's main objectives is to cause suffering on the part of the victim. Such is the case when an attacker is motivated by revenge – seeking to answer some real or perceived injustice. In these cases the victim's losses should be explicitly modeled as a benefit to the attacker.[26]

Statistics and the Hostile World

Statistics are excellent for describing stochastic events. In such cases information from a large sample of incidents is collected and analyzed. This is easy when there are a lot of potential events to study and when they occur due to combinations of independent random factors. For instance, based on past history, disk drive manufacturers publish statistics about the failure rate of their devices. They sell lots of drives and know how many come back as warranty claims in a given time period.

In the hostile world several issues make the application of statistics problematic. As discussed,

---

[25] The author once had an old spare tire and wheel stolen off of his van. A replacement was found at the auto wreckers for $35. It was hard to understand why anyone would take all of the risks of apprehension and a criminal record for something that could be purchased for $35. Evidently, $35 meant a lot more to the thief than to the van owner!

[26] As will be seen shortly, these losses will also be seen from the victim's perspective.

attack scenarios often require the execution of several distinct exploit operations. These exploits are not independent events. In fact, they are highly interdependent and chosen specifically by an adversary to accomplish a goal. Even more puzzling is the fact that an identical exploit operation may be used in a variety of attacks and the likelihood of that operation occurring will be different in the different scenarios (which may require additional scenarios of varying feasibility).

Two technically identical systems may have different likelihoods of attack depending on how attractive they are to different classes of adversaries. Consider three identical data base systems (same hardware, operating system, data base, system configurations) that differ only in the nature of the data being stored. One contains a collection of pizza recipes at the local mom and pop pizzeria. Another contains banking information. The third some super secret plans for a defense system. The pizzeria's adversaries might range from competitors to hungry script kiddies. The bank would face organized criminals with significant technical and financial resources. The defense system would be up against state sponsored labs with extensive resources. Of course there might be thousands of competing pizza joints after the pepperoni recipe but relatively few spy agencies going after the secret plans. These differences mean that statistics collected in one situation may have no relevance in a different situation.

Finally, in the cyber world, some of the most interesting attacks have occurred infrequently or not at all. It is difficult to collect a sample of incidents if the incidents haven't happened. Intelligent adversaries often choose to strike in novel ways because the defender has no prior experience that would have allowed them to prepare for the attack.

Nevertheless, we wish we had something like statistical probability to use in the hostile world. Something that gives us an indication of how likely an attack scenario is to occur. Our goal is to create a metric that will have a similar meaning to statistics but is not derived in the conventional, statistical fashion. We will call this metric *propensity*.

Capabilistic Propensity of Attack

Earlier we asserted that attackers are more likely to perform attacks that provide a high return with a low expenditure of resources. This could be described as the ratio between benefits and costs.

$$Attack\ Propensity = \frac{Attack\ Benefits}{Attack\ Costs}$$

Given that

$$Attack\ Difficulty\ (i.e.\ perceived\ Attack\ Costs) = \frac{1}{Attack\ Feasibility}$$

and that attackers will select attacks based on their desirability, this means that

$$Attack\ Propensity = Attack\ Feasibility \times Attack\ Benefit$$

It would also be convenient if *attack propensity* ranges between 0 and 1 (like a statistical probability), so we will probably want to normalize it in some way.

The objective is to make *propensity* correspond closely to statistical *probability*. But, what do we mean by *probability?* – we use the term all the time without being precise about it. Most of the time when people use the term *probability* they are really referring to *relative frequency*.

In statistics, the fraction of experiments that results in a particular outcome is known as the *relative frequency*. For example, flipping a fair coin 1000 times would be expected to result in about 500 occurrences of "heads", thus yielding a *relative frequency* (or *relative probability*) of 0.5.

Contrast this with the confusingly similar term *frequency*.[27] *Frequency* refers to the actual number of times the event occurs, often specified over a given time period or number of trials. So, if the coin flipping exercise were performed ten times per day then the frequency of "heads" would be "5", or more precisely, 5 heads per day.

In the case of hostile attacks, an "experiment" could be considered to be an encounter between a threat agent and the defender's system that gives the attacker the opportunity to consider an attack. It is hypothesized that the fraction of encounters in which an attacker chooses to perform a particular attack scenario corresponds to the *propensity* value for the scenario. If this is true then *propensity* = *relative frequency*, or at least, *propensity* $\propto$ *relative frequency*.

Note that the strategy suggested for calculating *propensity* automatically normalizes the values. The utility functions that generate a measure of *feasibility* as well as those that provide an estimate of *desirability* both yield values between 0 and 1. The product of these values will likewise vary between 0 and 1.

But, is this *propensity* value we invented actually related to the probability of the attack scenario being realized? It is difficult to prove this conclusively. Controlled experiments are difficult to stage. The best we can offer is that in the boundary conditions (e.g., where the attacker either has abundant resources or completely inadequate resources, and they either highly desire or do not desire the outcome of the attack) the *propensity* value behaves as it should (approaching 1 or 0 as expected). The results away from the boundaries give intermediate values – again, as might be expected. So long as our assumptions about human behavior are reasonably correct the *propensity* value should be meaningful.

There are limitations to assessing each scenario's propensity in isolation. Consider two scenarios, both with a high propensity value but one higher than the other. The scenario with the highest value is so much more attractive to the adversary than the competing scenario, the competing scenario may never be chosen. The value of the lower propensity scenario has been overestimated. If an error is going to occur then it is probably better to overestimate the likelihood of a scenario than to underestimate it. It may result in some wasted mitigation effort

[27] To avoid this confusion, we will try to use the terms *probability* and *propensity* to mean *relative frequency*, and the term *frequency* as explained above.

on the part of the defender, but that is better than being surprised by a scenario that was deemed unlikely.

Notwithstanding this issue, capabilistic *propensity* appears to behave like a *relative frequency*, or at least, the result of the computations are consistent with our model of human behavior. When the benefits are high and the perceived capability requirements are low the *propensity* will be close to unity. If the benefits are low or the resource costs exceed the attacker's capability then the *propensity* will be close to zero. At least for the boundary conditions *propensity* and *relative frequency* seem to match. The correspondence for points away from the boundaries (i.e., points along the utility function curves) will depend on the accuracy of our curves.

Although the goal is to provide a quantitative system of metrics that describe attack probability it must be recognized that predicting human behavior is much more difficult than testing the physical properties (such as strength) of a steel beam. Humans vary significantly (even within classes of threat agents) and there are almost infinite variables that influence human decisions.

Pain Factor – the Victim's Perspective

The discussion above focused entirely on the adversary and their rationale for choices. This allowed us to determine which attack scenarios were most likely to be chosen by an adversary (i.e., the propensity). However, from the victim's perspective, there is the added concern of how much damage will result – the perceived impact of an attack. Modeling victim impacts in an attack tree is similar to the scheme just shown for attacker benefits.

Similar to attacker benefits, impacts on the victim can occur at any node in the tree. The leaf level victim impacts are typically small and the largest impacts usually occur at the root of the tree – although it must be emphasized that this is not always the case. As before, the impacts accumulate (using aggregation functions) as an attack scenario progresses from the leaf level toward the tree's root. At certain key nodes the analyst injects the business impacts that are not derived in pure mathematical fashion.

Of course, as was the case with the various threat agent classes, different victims perceive similar losses differently. A wealthy multi-national corporation might treat a $100,000 one-time loss as insignificant whereas a small business would be devastated by such a blow. Again we employ utility functions to translate raw impact values into the subjective pain or damage perceived by the victim. The weighting mechanism used to derive an overall value for aggregated victim impact is similar to that used previously to calculate the positive impact (benefits) obtained by the attacker.

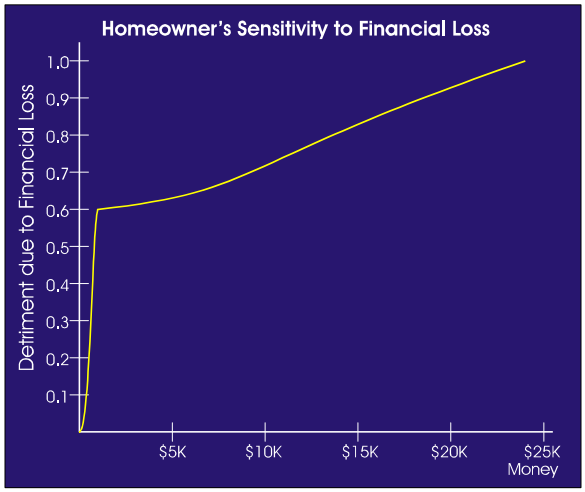


Figure 10 – Homeowner's Perceived Impact

Recalling the house burglary example, the utility function representing the homeowner's perceived damage due to financial loss from burglary is seen in **Figure 10**. The shape seems unusual until you

recall that most homeowners have some form of property insurance. If there is a loss, the homeowner pays the initial damage costs up to the policy's *deductible*. This is represented in the utility function by the steep rise from $0 to $1000 (the deductible). At that point the curve levels out because the insurance company begins to pay the damages. If life (and insurance) were perfect, at this point the curve would become flat. However, experience has shown that insurance rarely covers the entire loss. Sometimes there is a disagreement over the value of the item for which compensation is being paid and not all items may be covered. In many cases items that are stolen or damaged are irreplaceable or hold sentimental value. Thus the curve continues to rise gently[28] and approaches some level (in this case, 1). Of course this curve would be different and more linear over the entire range of losses if the homeowner did not have insurance![29]

The *{Break down door}* scenario has an actual damage figure of $15,250. The intruder does $250 damage to the door and steals $15,000 of goods once inside the house. Using the curve in **Figure 10** this translates to a perceived financial impact of 0.83. The *{Steal opener from car, Break down passage door}* scenario, with $18,500 of damage, has a perceived impact of 0.9.

Recall that when we considered the behavior of the adversary we created distinct sets of functions for each threat agent. Similarly, in *victim impact* analysis, it is also possible that there are several victims. In the case of the house burglary example it is obvious that the homeowner suffers as a result of an attack. However, if the loss exceeds the insurance policy's deductible then the insurance company also experiences a loss. An impact curve describing the pain felt by the insurance company for varying levels damage could be created.

Since victim impact is a component of risk it is important to recognize that risk is always seen from a particular stakeholder's point of view. Whenever the term *risk* is used, always ask "risk to whom?"

Scenario Risk Value

Recalling that

$$\textit{Relative Attack Risk} \equiv \textit{Attack Probability} \times \textit{Attack Impact}$$

and given our hypothesis that

$$\textit{Attack Scenario Propensity} = \textit{Attack Scenario Probability}$$

or, at least,

$$\textit{Attack Scenario Propensity} \approx \textit{Attack Scenario Probability}$$

we are now in a position to calculate the attack risk posed by each threat agent for each scenario.

---

[28] We have assumed that sufficient insurance has been purchased to cover the total destruction of the property. If it is possible that damage could exceed coverage, then the curve would again rise sharply at the point coverage ended.

[29] And if the homeowner's insurance reached its limit (and the homeowner was again responsible for the excess damages) the curve would be different still. After the limit the curve's slope would again increase.

Attack Scenario Relative Risk = Attack Scenario Propensity × Attack Scenario Impact

Calculations for the two house burglary related scenarios discussed earlier are shown below.

Calculation of Juvenile Delinquent Risks for Two Attack Scenarios

First, determine how difficult the attack scenarios are to carry out based on the perceived value of the resources expended by the juvenile delinquent.

Resource Requirements for Attacks by Juvenile Delinquent					
		{Break down door}		***{Steal opener from car, Break down passage door}***	
		Raw Resource Required	Juvenile. Delinquent Feasibility	Raw Resource Required	Juvenile. Delinquent Feasibility
Attacker Cost	Cost of Attack	25	0.9	30	0.79
	Technical Ability	10	0.9	10	0.9
	Noticeability	0.3	0.95	0.28	0.97
(Geometric Mean Rule)		**Overall Feasibility**	0.9164	**Overall Feasibility**	0.8835

Then, examine the perceived benefits the juvenile delinquent anticipates from the attacks.

Benefits Accrued by Juvenile Delinquent from Specific Attacks					
		{Break down door}		***{Steal opener from car, Break down passage door}***	
		Raw Gain	Perceived Value	Raw Gain	Perceived Value
Attacker Benefit	Money gained	$15,000	0.78	$18,000	0.82
	Satisfy Destructive Urge	5	0.5	7	0.8
Compute Weighted Sum 0.4 × Money + 0.6 × Satisfy Urges		**Overall Benefit**	0.612	**Overall Benefit**	0.808

Combine the perceived difficulty and the perceived benefits using a cost-benefit computation. This yields the relative frequency or *propensity* that an encounter between a juvenile delinquent and the house would result in the scenario being carried out.

Capabilistic Propensity of Attack by Juvenile Delinquent			
{Break down door}		***{Steal opener from car, Break down passage door}***	
Feasibility × Benefits	Propensity	Feasibility × Benefits	Propensity
0.9164 × 0.612	0.561	0.8835 × 0.808	0.714

Next, determine the level of suffering the attacks will cause (as perceived by the victim)

Impact Experienced by Homeowner			
{Break down door}		***{Steal opener from car, Break down passage door}***	
Actual Damage	Perceived Damage	Actual Damage	Perceived Damage
$15,250	0.83	$18,500	0.9
Overall Suffering (Single impact, no need to calculate weighted sum)	0.83	$18,500	0.9

Finally, combine the *propensity* of each attack scenario with the perceived victim impact to find the relative risk.

Risk of Two Selected Attack Scenarios (Propensity × Impact)		
Attack Scenario	*Propensity × Impact*	*Relative Risk*
{Break down door}	0.561 × 0.83	0.466
{Steal opener from car, Break down passage door}	0.714 × 0.9	0.643

Relative Risk vs Cumulative Risk

As stated earlier, risk is a combination of both probability and impact. If propensity is used as the probability term, and *pain factor* as the impact term in the risk equation, the result is a measure of the risk associated with a single encounter between the threat agent and the defender's system. We say that propensity is *relative risk* because it is calculated relative to each encounter between the system and the attacker.

The difference between *relative risk* and *cumulative risk* can be illustrated with an example from the insurance industry. A prominent insurance company advertises special, discounted auto insurance rates for seniors, ostensibly because their extensive experience makes them excellent drivers. Closer investigation reveals that seniors are not generally good drivers – their accident rate per mile is surpassed only by new drivers. However, seniors are usually retired and do not use their cars for daily commuting. The total number of miles they drive is much lower than the average motorist. This, it turns out, more than offsets their diminished capabilities. So, although they have a high *relative* (per mile) accident rate they have a low *cumulative* collision rate. Their accidents per year are lower than the average motorist. The insurance company cares about the frequency of their accidents – how often they will have to pay a claim. Thus, they are willing to offer seniors better rates despite their poor relative accident rate.

Similarly, management often cares more about how often successful attacks will occur (incurring some impact) than the more abstract number showing the likelihood of an attack if an encounter between an adversary and their system takes place. We now need a way to estimate how often our adversaries will engage their target.

Opportunity – Estimating the Number of Encounters

Three primary factors contribute to the number of encounters that a system will undergo with a specific class of threat agent in a period of time:

1. The number of adversaries who have plausible access to the defender's system. For physical attacks this generally means the attackers and the target are within the same geographic region. For electronic attacks it implies some degree of mutual network connectivity (Internet, leased lines, dial-up lines).
2. The number of targets (within the threat agent's region of operation) competing for the attention of the attacker.
3. The period of time over which the number of encounters will be estimated.

There are additional factors that also affect the number of encounters. The nature of the attacker affects the characteristics of the exploit. The following questions help identify these factors.

- Will performing the exploit permanently deplete the attacker's resources? This limits each attacker to a single attack, referred to as a *single-shot* attack. Examples of single-shot attackers include suicide bombers and political defectors.
- How long does it take for the attacker to do the attack and then regroup for another attack? The regrouping time can be thought of as the attack preparation or recovery time. Such attacks can be performed repeatedly and are called *single threaded* or *sequential* attacks. The house burglar is an example of a sequential attacker.
- Can an attacker attempt to attack multiple targets concurrently? *Multi-threaded* attacks are very common in automated electronic (computer) attacks. They are also possible when a team of individuals are working together on a particular attack.

- What fraction of an attacker's day is spent performing attacks? Most attackers need to sleep (which rules out about a third of their day). Of course teams of people operating in shifts or an attacker using automated attacks could carry out their efforts 24 hours per day.

For each type of exploit, the number of encounters in a given time period[30] can be estimated by

$$\#\ Single\ shot\ encounters = \frac{\#\ Adversaries}{\#\ Targets}$$

$$\#\ Multi\text{-}threaded\ encounters = \frac{(\#\ Adversaries)\ (Thread\ Factor)\ (Duty\ Factor)}{(Attack\ Time + Recovery\ Time)\ (\#\ Targets)}$$

$$\#\ Single\ threaded\ encounters = \frac{(\#\ Adversaries)\ (Duty\ Factor)}{(Attack\ Time + Recovery\ Time)\ (\#\ Targets)}$$

The *single threaded encounters* and *multi-threaded encounters* calculations yield the number of attacks per whatever time period is used in the *attack time* and *recovery time* parameters. This would typically be minutes, hours or days depending on the nature of the attack. In many situations it is more convenient to convert to the number of encounters per year by multiplying by whatever conversion factor is appropriate. Obviously the number of expected encounters increases as the time period becomes longer.

These formulae are approximations intended to give an order of magnitude estimate of the anticipated number of encounters.. It is expected that future research will provide refinements.

Attack Frequency or Rate of Occurrence (RO)

The number of times a particular attack scenario will occur in a given time period is proportional to the scenario's *propensity* (relative frequency) and the number of encounters that occur in the time period. That is

$$Rate\ of\ Occurrence = Propensity \times Rate\ of\ encounters$$

or

$$Rate\ of\ Occurrence = Propensity \times \#\ encounters/time\ period$$

If the time period chosen is one year, then the frequency is known as the *Annual Rate of Occurrence (ARO).* This term is widely used in conventional risk analysis.

As noted earlier, these calculations are based on the assumption that each scenario is the only one open to consideration by the attacker. The possible interscenario competition means that the calculations may overestimate the attack frequency. Although this is an error, it is a safe error in that it does not result in any potentially high frequency scenario being overlooked. It may encourage the defender to implement defensive controls that are not essential.

[30] For *single shot* attacks it is assumed that the time period is long enough to allow all of the actors time to have an encounter and decide whether or not they want to irrevocably spend their resources. The formulae for *single threaded* and *multi-threaded* yield the number of encounters per the time period units used to specify the attack and recovery times.

Cumulative Risk and System Lifespan

As mentioned earlier, a scenario's cumulative risk is a combination of its propensity and the rate of encounters (over a particular time period). So, for a given scenario,

$$Cumulative\ Risk = (Propensity \times \#\ Encounters/time\ period) \times Time \times Impact$$

or

$$Cumulative\ Risk = Rate\ of\ Occurrence \times Time \times Impact$$

Note that risk increases as the time period increases. The importance of this can be illustrated with an example from the distant past.

Some experts suggest that the probability of earth being hit by an extinction grade (5 to 10 Km size) asteroid is about 10^{-8} per year. Apparently, this event seemed so unlikely that the dinosaurs roaming the earth were apparently unconcerned. In the short term their lack of attention to the matter was understandable – the chances of anything bad happening in any particular dinosaur's lifetime was extremely small. So, they neglected to develop a space program and early warning system to deal with errant asteroids. Unfortunately over the aeons the risk accumulated and one day the dinosaurs were no more.

Defense and aerospace engineers understand that system lifespan is an essential part of risk analysis. The design of military avionics includes a feature known as *anti-tamper* or *tamper resistance*. There are many valuable secrets incorporated in military avionic components. If these components should fall into enemy hands (perhaps due to a plane crash) there is great concern that a capable adversary might be able to extract the secrets and find exploitable weaknesses or simply duplicate the technology without incurring the development costs. To prevent this from happening a variety of technical impediments are included in designs to prevent tampering. Attack tree analysis is frequently used to assess whether the anti-tamper features will withstand attack.

However, it is also recognized that nothing will withstand a determined attack indefinitely. Associated with the design of critical avionics devices is a stated lifespan over which the device is expected to withstand probing. For instance, a given circuit board may be deemed to have a seven year lifespan. Attack tree analysis is used to provide assurance that even if the first boards off the assembly line are given to a hostile foreign intelligence agency that they will be unable to extract secrets from the devices in less than seven years. The avionics boards are put into service with the understanding that their lifespan is limited. In fact, as the design life approaches (say at the five year mark for the example above) work begins on a new replacement board. When design life is reached it is assumed that the boards have been compromised and they are withdrawn from service and replaced with the updated components.

These principles apply to all types of systems. Defenders must be aware that no system will withstand attack indefinitely. It is essential that the security controls be adequate to protect the system throughout its expected lifespan. A useful metric to consider is how long a system will endure until risk reaches some threshold. A threshold of 1.0 is commonly used as the threshold. This leads to the metric of *time until risk reaches unity.*

The Meaning (and Calibration) of Risk

The preceding discussion has delved into methods for estimating likelihood (propensity), victim impact and ultimately relative and cumulative risk. However we have thus far avoided explaining what the risk value actually means. If an attack scenario's cumulative risk value is 0.5 is that good, bad or somewhere in between?

Propensity purports to be a relative probability that varies between 0 and 1. The assessment of the feasibility and desirability factors that make up propensity are admittedly dependent on the accuracy of the utility functions used to derive them. However, as shown earlier, feasibility is correct for boundary conditions showing the attacker's ability to provide the necessary resources for an attack, and it is must be reasonably close for conditions between the boundaries. Desirability is an attempt to grade the attacker's motivation. At least for desirable scenarios, the motivation is assumed to be high (meaning that whether or not the attack scenario can be performed is limited only by feasibility. Thus, propensity is arguably intrinsically calibrated and behaves like a relative probability figure.

The relevance of the second term in the risk equation – victim impact – is less clear. Through the use of utility functions, the absolute losses experienced by the victim have been mapped to a 0 to 1 range of values. The question is, what does this mean?

Does an impact of 1.0 represent a worst imaginable case? The worst case that can be tolerated by the organization? By the department? The answer is simply that it means whatever the analyst wants it to mean. When the analyst defines the impact utility functions they are effectively setting the meaning of an impact value of 1.0.

What strategy might they pursue in calibrating their impact values? An obvious choice would be to set 1.0 to represent the worst case scenario. In some situations this is a good choice. But use caution in pursuing this avenue because it may under represent actual risks.

The author was once commissioned to perform an attack tree-based risk assessment at a technical college. The technical college taught a variety of vocations, ranging from paramedic to electronic technician to automotives to welding. The school was particularly proud of a new building that was under construction for the welding students. One section of the building was an area with metal tables and oxyacetylene gas torches for the students to use to practice. Oxygen and acetylene gas was supplied from a room filled with hundreds of interconnected gas cylinders. Of particular concern were the acetylene cylinders which were all to be connected to a common manifold. A breach of the manifold would cause the highly flammable and explosive acetylene gas from all of the open cylinders to vent.

Interviews with the school's directors established that their number one priority was the safety of students and staff, followed closely by the need to preserve the institutions ability to deliver training. Investigation of the properties of acetylene gas established that if an intruder gained access to the gas storage facility and vented the acetylene cylinders and then provided an ignition source the resulting explosion would level the new building and cause serious damage to the adjacent building. If the attack occurred during school hours as many as a hundred deaths could be expected with several times as many injuries. The disaster would likely result in the closure

of the school for weeks or even months while an investigation took place. Initially this catastrophic attack scenario was used to calibrate an overall impact or pain level of 1.0.

As the modeling progressed, it was brought to the analyst's attention that a more common situation, and one that sadly exists on many campuses, was that of occasional sexual assaults. These generally took place late in the evening when someone (usually a young woman) was returning to her car after studying or being involved in a social event. The question became, what should the impact of such an incident be? Having just said that 100+ deaths, several hundred injured and tens of millions of dollars of damage rated an impact of 1.0, what value would be associated with a sexual assault? If you even assumed a linear scale then a single death would rank as an impact of 0.01. A non-fatal sexual assault would be less than that, perhaps 0.0025. Using such a low value for the impact of a sexual assault would result in a correspondingly low assessment of risk. The study would effectively state that the institution did not perceive any risk from the sexual assault incidents and would find no reason to devote resources to preventing them.

Of course, this was completely out of tune with the values of the school's directors. They took the issue very seriously. In this case, the model was recalibrated to reflect that a single death had an impact of 1.0. A sexual assault was then ranked as 0.2 impact. Given that (sadly) three or four such assaults occurred every year the cumulative impact (over a one year period) resulted in a risk of around 0.6 or 0.8. This allowed easy justification for improved security measures on campus.

But what about the acetylene gas explosion scenario? Although the impact of that scenario was off the scale the propensity was very low. Further, a number of easy to implement mitigation controls became evident that lowered the feasibility of doing such an attack successfully to a value approaching zero. This more than compensated for the off the scale impact value.

In order for threat models to be effective for making security decisions they must be meaningful in the context of the system being studied.

It is also worth pointing out that, in the case of the polytechnic school's sexual assaults, it was assumed that impacts sum linearly. I.e., three events of *impact = 0.2* was equivalent to a single event of 0.6. Whether or not this is true depends entirely on utility function curves that map raw damages to perceived damages. That is, the utility functions must compensate for any non-linearity that may exist between the raw damages and the perceived damages.

Also note that because

$$Cumulative\ \ Risk = Event\ frequency \times Time\ Period \times Event\ Impact$$

it is possible for the cumulative risk value to be greater than one if a sufficient number of events occur within the chosen time period. Even low impact events can pose a significant risk if a sufficient number occur. This again emphasizes that the system's lifespan is an important factor in determining whether or not the risk associated with a scenario is acceptable.

Overall Risk

To this point, our focus has been on assessing the risk associated with individual scenarios. Management often asks about the overall risk associated with a system, not the risk for specific scenarios.

The risk values of the individual scenarios were calculated as if each was the only scenario available for consideration by the adversary. If our assumption that the adversary will always choose the scenario with the highest propensity value (and associated victim risk) is correct then none of the other scenarios matter. But this isn't realistic. Scenarios often have similar or even identical propensity values. The uncertainty in our models makes it difficult to say with certainty which definitively has the highest value.

This is important because nothing precludes a scenario with slightly lower propensity than another from having a much higher victim impact (and consequently greater risk than the scenario with the marginally higher propensity). If we are even slightly off in our propensity estimates, then the victim may in fact be exposed to the higher risk value than initially supposed. Slight differences in the preferences of individuals within a threat agent class almost guarantee this to be the case.

Here is one strategy that might be used to accommodate the uncertainty in our model and provide a better estimate of the overall risk a particular threat agent poses to a system.

1. Sort the scenario table by propensity.
2. Throw away the scenarios with lowest propensity leaving the top scenarios within the model's uncertainty. E.g., if it is supposed that propensity values could be off by as much as 20%, then keep the 20% of scenarios with highest propensity and discard the rest.
3. Find the highest cumulative risk value within the group of the remaining scenarios.

Annual Loss Expectancy

A metric popular with management is *Annual Loss Expectancy (ALE).* Given the *Annual Rate of Occurrence* it is possible to calculate the *ALE* associated with an attack scenario for each victim impact indicator.

$$\textit{Annual Loss Expectancy} = \textit{\# Occurrences (per year)} \times \textit{Scenario Impact}$$

or

$$\textit{Annual Loss Expectancy} = \textit{Annual Rate of Occurrence} \times \textit{Scenario Impact}$$

There are several caveats associated with this calculation. First, it is assumed that the effect of the losses associated with multiple events is additive. This may not always be the case. Second, the *ARO* is computed for each scenario as if it were completely independent from all other scenarios. This is clearly not true since one scenario may be sufficiently attractive to distract an adversary from an otherwise attractive attack.

The overall *ALE* (for all scenarios) from a given threat agent is at least as much as the maximum *ALE* of any of the scenarios in the set. It may be as much as the sum of the *AROs* of all of the scenarios (if they are seen as independent choices by the threat agent).

Of course the choice of the *annual* time frame is arbitrary. The number of encounters and losses can be evaluated for any time period of the analyst's choosing. In general, the number of encounters, losses and risk will increase with exposure time. An exception to this might be a highly targeted attack. In that case the encounters are not random nor ongoing. There may be no encounters from a particular adversary until some real world event occurs which triggers interest in the target.

Finally, management often expects the scenario impact to be expressed in monetary terms. Indeed, money is used as a neutral way of expressing all types of losses. Although it may sound callous, it is quite common for industry to convert deaths and injuries into monetary figures. The cost may reflect money paid out in fines or through lawsuit penalties. Similarly environmental damage can be expressed in terms of penalties and the remediation costs. In situations where these costs are so low as to compromise the public good it is common for governments to artificially inflate them through laws that impose financial penalties for security incidents. This strategy forces companies to behave ethically while still being faithful to their shareholders (who demand maximum return on their investments).

A Risk Paradox (or, the Dwarfs are for the Dwarfs)

Formal analysis sometimes produces surprising insights about how the actions of the defender affect the adversary. There is a temptation to view the attack process from the point of view of the defender. Obviously the defender's interests need to be taken into account. However the two players in the game have separate and generally disjoint interests.

The final book in C.S. Lewis' *Chronicle's of Naria* children's fantasy series *(The Last Battle),* describes a group of dwarfs. Various attempts are made to persuade the dwarfs to participate in a variety of activities. The highly suspicious dwarfs refuse all entreaties because they cannot accept that any action encouraged by others would have positive benefits for the dwarfs themselves. They completely reject the principle of mutual benefit and respond to these offers by chanting loudly, "The dwarfs are for the dwarfs!"

When contemplating the interaction between adversaries and their victims it is important to remember that like C.S. Lewis' dwarfs, "The attackers are for the attackers!". In other words, attacker choices are made solely based on the consideration of their own interests and not on the possible impacts to the victim. As mentioned previously, in the special case where the attacker's goal is to cause victim suffering that should be modeled explicitly by making victim pain one of the attacker benefits.

Consider now a situation where an attacker discovers two attack scenarios. Both have high propensity values (meaning they are both feasible and desirable), however one has a slightly higher propensity than the other so it will normally be chosen by the adversary.

As it turns out, the two scenarios describe quite different paths through the attack tree. In this particular case the scenario with the lower propensity causes a lower impact on the victim than the higher likelihood scenario. Of course, this is not a problem for the victim because the attacker will follow their own self interest and choose the higher propensity scenario.

The analyst notices the higher propensity scenario and devises a control that will greatly decrease the feasibility of this scenario and thus lower its propensity. The adversary responds by moving to their next best choice which is almost as good as their initial choice. Unfortunately for the victim, the impact from the adversary's new choice will be more severe. The likelihood of the attack has decreased only marginally but the pain suffered may have increased by a order of magnitude. Bizarrely, the implementation of a perfectly valid security control has caused the victim's risk to increase!

One strategy for identifying these types of situations is to recognize that our models involve considerable uncertainty. Indeed, due to this uncertainty, we should not place undue importance on the relative order of scenarios with similar propensity values. A better strategy may be to identify the scenarios with highest victim impact within scenario groups of similar likelihood and address the risk they pose.

Scenarios Involving Both Intentional and Random Events

Attack tree analysis was developed to analyze deliberate, hostile activities against a system. An earlier, tree-based approach (known as *fault tree analysis*) has long been used[31] to model random events. It would be highly useful to combine the two techniques to analyze incidents that occur due to either (or both) random and intentional events.

Differences between attack trees and fault trees

There are fundamental differences in the properties of fault trees and attack trees. In a fault tree, each leaf node has an associated probability value (usually obtained from statistics). Using statistical formulae[32] it is possible to calculate the statistical probability of reaching any node in the tree (taking into account the individual probabilities of all of the paths that lead to that state). Unlike an attack tree, it is not necessary to examine each path separately (although this can also be done). In fault trees the leaf level events are treated as independently occurring incidents. Fault tree practitioners call the paths in their models "cut sets" – which we have called attack scenarios in attack tree models.

In an attack tree, the leaf level events are heavily interdependent, particularly where *AND* nodes are involved. If several operations are required to carry out an attack an intelligent adversary will not waste time performing a preliminary activity (even if it is easy) unless they believe they are

---

[31] Fault trees were developed by H.A. Watson at Bell Laboratories for use by the U.S. Air Force on the Minuteman missile project.

[32] Typically *OR* nodes in a fault tree combine probabilities using 1-[(1-a)(1-b)(1-c)...(1-n)], where *a*, *b* and *c* represent the probability values of children . *AND* nodes combine as the product of the probabilities of the children.

also capable of carrying out more difficult subsequent steps in the attack[33]. Sometimes the attacker is not even in a position to attempt some of the leaf nodes unless they first complete other prerequisite steps. Having completed one of a series of steps in a procedure, the next operation becomes the focus of the attacker.

The interdependencies between leaf level events make it impossible to determine, in isolation, the probability of each of the substeps in an attack. Aside from the practical consideration that the statistics may not exist, the probability of the adversary performing operation *A* fundamentally depends on their expectations of the feasibility of *B.* This is one of the main reasons why attack tree analysis focuses on *attack scenarios* and deriving the probability from factors such as feasibility and desirability. An *attack scenario* is a complete set of steps that constitutes a successful attack. The steps are considered as a package.

Mixed Models

Consider the general case involving a tree model with leaf level events that include both random, probabilistic incidents (e.g., natural disasters, equipment failures, human errors) and hostile, resource constrained attacker activities. Any given attack scenario for this tree may consist of

i Probabilistic events – events with known probability (often acts of nature or routine malfunctions)

Calculating the probability of a scenario that contains only probabilistic events is trivial so long as the appropriate statistical data exist. The well known statistical method for computing the probability of multiple, independent events (i.e., an *AND* node) is to simply multiply the individual probabilities together. The formula for computing the probability of any one of a set of *n* possible events (i.e., for an *OR* node) is 1-[(1-a)(1-b)...(1-n)].

Note that probabilistic events have no resource cost values. For example, there is no cost or technical ability required (from anyone) for a hurricane to occur. The probability of these events is independent of threat agents.

ii Capability-constrained events – events that require the adversary to expend resources

The indicator definitions relating to adversary resources make it easy to calculate[34] the total amount of the various resources that will be expended in the particular attack scenario. Passing these costs through the resource constraint utility functions discussed earlier allows an estimation of the feasibility of the scenario. Combining this feasibility estimate with the desirability value (obtained by using the attacker benefit utility functions) yields the *propensity* value for the attack

[33] They may also do a preliminary step if it leads to an intermediate goal they feel is worth achieving. This can be understood through subtree analysis.

[34] *OR* nodes in an attack tree scenario simply inherit the resource requirements passed up by the child that is participating in that scenario. *AND* nodes combine their children's resources through an analyst specified formula.

scenario. If the utility functions are accurate, *propensity* is equivalent to *probability* (or more precisely, *relative frequency*).

We emphasize that the *propensity* is calculated from the set of hostile operations that the attacker must perform. This makes sense because the attacker chooses whether or not to perform the component hostile actions by considering the attack scenario, with all of its operations, as a single unit. So, although we do not know the propensity of each individual operation, we can determine the propensity of the set of operations that make up an attack. In other words, we do not (generally) know the propensity for an adversary to reach particular intermediate nodes in a tree. We do know the propensity that the adversary will reach the tree's root using a particular attack scenario.

iii A mix of both Probabilistic and Capability-constrained events

When we talk about a mixed incident, we are usually referring to a hostile attack that requires some random event as a prerequisite or corequisite. These situations are quite plausible. For instance, areas situated along the Gulf of Mexico typically experience several hurricanes per year. During a hurricane, a facility that is normally manned may be deserted. Disruptions to power and communication lines may mean that burglar alarms do not function (or their operators may believe alarms are a result of the storm). This may embolden an adversary to carry out an attack that they would otherwise not consider.

In a sense, the random event does not actually change the hostile portion of the attack scenario so much as it opens a restricted time window during which it may occur.

Most of the statistics for random events are given as a frequency and a duration. For instance, a component may have a *Mean Time Between Failure (MTBF)* specification and a *Mean Time To Repair (MTTR)*. Hurricanes have a frequency (number of hurricanes per year) and a duration (a few days).

Since the hostile parts of the mixed attack are only plausible during the interval in which the random situation is in effect it means that we should calculate the number of hostile encounters based on the shortened interval. For instance, if two hurricanes per year are expected, with a duration of two days each, then there will be approximately four days of hurricane per year. Analysis of a scenario with a hostile component that depends on a hurricane would require us to calculate the number of expected encounters over a four day period, not 365 days. This would mean that only about 1% of the encounters would occur. The overall probability of the mixed scenario would be about 1% of the same scenario without the random component.

Note that, in the discussion above, the threat agents are not responsible for creating or instigating the random events. They merely take opportunistic advantage of these events when they transpire. There is another important random factor that has not been dealt with so far.

Probabilistic Outcomes of Capabilistic Activities

In some cases, a probabilistic factor is introduced because of a capabilistic action or actions of an adversary. In a sense, it is as if the adversary rolled a dice. The outcome (orientation of the dice) is determined by probability but there would have been no possibility of an outcome unless someone rolled the dice. The adversary creates the event but not the outcome. We call these events *probabilistic outcomes*. Although *probabilistic outcomes* are most often observed at leaf nodes, they can occur at any node in the tree that has a capabilistic component.

At the leaf node level there is a direct interaction between the adversary and the target. In the discussion thus far, leaf level actions have been completely deterministic in nature. If the adversary applied the resources specified in the node's capabilistic requirements, the outcome was assured. This is overly simplistic. In some cases, despite the application of the requisite resources, the leaf level operation may fail due to random factors that are not entirely predictable and beyond the attacker's control.

Similarly, there may be random factors that influence whether or not an attack progresses past an intermediate *AND* or OR node. Despite all of the necessary conditions being satisfied by leaf nodes or subtrees below, the *AND/OR* node may occasionally still fail to be achieved due to random factors.

To model this random component, capabilistic nodes can be given an attribute called the *attack success efficiency (ASE)*. The *attack success efficiency* is input as a value between 0 and 1 that specifies the likelihood that the application of the specified resources will cause the node to succeed.

The user can specify one of two ways in which the *attack success efficiency* term could be interpreted. It could affect either the *feasibility of attack* coefficients or the *attacker benefits* of an attack scenario. The correct interpretation depends on whether the threat agent is astute or naive.

If the adversary is clever they will recognize that one or more of the operations in an attack scenario have a possibility of failure that is beyond their control. They will be aware that the average return will be less than the nominal return associated with a completely successful attack. The effect on their motivation is best understood by devaluing the raw benefits before they are translated to perceived benefits via their respective utility functions. We call this approach, *attacker benefit-based ASE*.

Note that the attack detriments are not affected since they usually apply whether or not the attack is successful. The resulting reduction in perceived attacker benefits will make the attack scenario less desirable and thus reduce the propensity of the attack scenario. This affects both the relative risk (i.e., the risk on a per encounter basis) and the absolute risk.

In other cases, the adversary may be naive or optimistic, not recognizing that the exploits and attack scenario they hope to use have less than a 100% success rate. If this is the case, the ASE should be applied to the scenario frequency term. The number of encounters is multiplied by the *attack success efficiency* (yielding an effective # encounters term) which is used to calculate the

the expected scenario frequency. The relative risk is unchanged, but the cumulative risk of all scenarios involving this node is reduced by the *attack success efficiency* factor. We call this *encounter-based ASE*.

It is possible that a given scenario may have both *attacker benefit-based ASE* and *encounter-based ASE* components. In that case it will be necessary to accumulate the attacker benefit-based ASEs separately from the encounter-based ASE terms. The former will be multiplied together to get an overall attacker benefit-based ASE which will reduce the attacker benefits before they are transformed by the utility functions. The latter will be multiplied together and used to compute an overall effective # of encounters term.

In both *attacker benefit-based ASE* and *encounter-based ASE* cases, the cumulative risk will be decreased for any attack scenario that has components with ASE values < 1.

Total Risk from Hostile and Stochastic Events

The scenarios (or cut sets) in the fault tree models are statistically independent. A reasonable assessment of the overall risk from stochastic events can be found by summing up the risk from each cut set. This is similar to how insurance companies assess risk and charge for different types of coverage in their policies. In house insurance they look at the risk from fire, flood, burglary as separate and unrelated. Generally a fee is charged for each type of coverage.

A strategy for estimating overall hostile risk was shown previously. This strategy should also work for situations where hostile risk and random risk are combined.

Because hostile and purely random risk events are unrelated the overall risk is the sum of the two.

A similar approach can be taken at summing the hostile risk from different adversaries. So long as the different adversaries operate independently, and do not collude, the total risk from all adversaries is the sum of the risk from each one.

Sub-root Analysis

It should be reiterated that all of the analytic techniques discussed thus far explore adversaries' ability to reach the root node goal in the attack tree model. The aggregation functions sum up the costs of reaching that objective. Feasibility and risk are assessed with respect to the root objective.

Nothing precludes an adversary from pursuing a lower level goal in the tree. Reaching that attack state may not satisfy all of their objectives, but may be sufficient to continue to motivate them. If this is the case then it is prudent to study the model with that in mind. This is easily done by taking a subtree of the original model and repeating the analysis with the subtree's top node as the tree's new root.

How do you know if subtree analysis is required? Must analysis be repeated for every node in the tree beneath root on the chance that it will be attractive to adversaries?

A better strategy is to survey the original tree to identify the non-root nodes that have significant attacker benefits or victim impacts. This implicitly identifies the subtrees that might stand on their own as being attractive to attackers or that might contribute to the victim's risk. In most cases there are only a few intermediate nodes that meet these criteria. These subtrees are easily brought out of the main model for further analysis.

Heuristics for Reducing Combinatoric Growth of Scenario Space

As noted earlier, certain tree structures generate exponential growth in the number of attack scenarios represented by the model. This is not an indication of poor security architecture. In fact, as will be seen shortly, adding controls to a system often increase the number of scenarios that must be evaluated. Many (or most) of these scenarios will turn out to be low propensity and low risk – but they must still be evaluated. Even computers meet their match in exponential growth situations. Fortunately there are techniques and heuristics that can quickly eliminate scenarios from consideration.

Scenario Reduction

Consider a situation where there exist two scenarios in a particular attack model. Scenario "A" provides the adversary with the same or greater benefits than scenario "B" and requires the same or less capability resources as "B". The impact on the victim of scenario "A" is also the same or higher than "B".

If our model is correct, there is no situation under which any adversary would choose to execute scenario "B" given that "A" is available. Furthermore, "B" is of the same or lesser impact on the victim. So, from the victim's point of view, all they need to consider is scenario "A".

This technique of scenario reduction has proven extremely effective in reducing the attack scenario spaces in large models. In many cases there are multiple scenarios with similar characteristics. Applying reduction may obscure how an attack might occur (since that attack may have been eliminated in favor of some very similar but different attack). What is not lost is whether the attack scenario poses a risk.

The technique of reduction can be applied at any level or node in the tree. Basically this obscures the detail of how the attacker might reach the top of the reduced tree, but not whether the attacker will use that avenue of attack.

For instance, suppose an attack tree contains a subtree representing the ways in which an attacker might compromise Microsoft Windows. One example of a Windows attack tree describes 800 different attacks. When reduced, the number of scenarios that remain drops to 14. Given that a Windows subtree appears in many cyber related attack tree models, and that it is frequently beneath an *AND* node, this can drop the total number of attack scenarios in the model dramatically.

Ganged Subtrees

In many cases, attackers are faced with multiple instances of some obstacle that they need to overcome. For example, most organizations have a multi-tiered network architecture and use the

same type of doors and locks throughout their facility.

For instance, the perimeter of a company's network may connect to the Internet via a firewall. The firewall connects to a DMZ network that supports web or mail servers. A second firewall connects the DMZ to the business network. The business network is connected (via a firewall) to an internal DMZ which houses a historian server with a copy of data that originates (across yet another firewall) on an industrial control system network. An Internet-based attacker would need to cross four firewalls and compromise hosts on four separate networks.

Consider for a moment only the work involved in compromising the series of hosts. If we suppose that all of the hosts were similarly configured Windows systems (whose attacks were described in the 800 scenario tree mentioned above) then the tree would likely contain an *AND* node with four instances of the Windows subtree beneath it. The number of possible combinations of attacking those four Windows systems would be 800^4 or 4.096×10^{11}. This astonishingly large number is even beyond the capability of most computers to evaluate in a reasonable period of time!

However, from a practical perspective, this is not how any sane attacker would behave. Once they devised a strategy to compromise the first Windows host surely they would use the same technique on subsequent hosts (given that they are all similarly configured). So the number of possible attack scenarios to get past all four hosts would be $800 \times 1 \times 1 \times 1 = 800$, a much more tractable value.

Similar situations can be imagined involving physical attacks. If an attacker has to penetrate multiple doors to move from the outside of a facility to some desired inner room they would surely repeat the exploitation strategy perfected on the first door on subsequent instances (if the doors and locks were similar).

All that is necessary to represent this in an attack tree is to make some type of connection between the various instances of the repeated subtrees. We have coined the term *ganged subtrees* to describe this technique. The term *gang* originates in the field of electronics where controls (such as volume controls on a stereo) are connected in such a fashion that all of the controls move together. Turning up a stereo's volume control increases the volume from both left and right channels.

Countermeasures and Controls

The discussion above has shown how attack trees can model an adversary's behavior with respect to a target, and even how an assessment of risk can be performed. However, surely it is the point of the exercise to prevent attacks or mitigate the effects if they occur. The attack tree models described previously use three different techniques to model controls and countermeasures.

1. The capabilistic resource requirements associated with a leaf node can be increased to reflect an improvement in the leaf level component the attacker was attempting to exploit.

   For example, if an inexpensive, hollow core door (with a cheap lockset) was replaced with a high quality, solid core door (equipped with top grade hardware) then battering through or prying open the door would become more difficult. The *Technical Ability* and

Cost of Attack indicator values for the *Batter Door* and *Pry Open Door* leaf nodes would increase to reflect the improvements in the door components.

2. Changes can be made to the defender's system to make an attack scenario more complex and challenging.

   In cases where an attack required a series of steps (depicted by an *AND* node with several children), then additional children could be added beneath the *AND* node representing new activities contrived by the defender[35]. The new activities would be chosen by the defender to be as difficult as possible.

   For instance, if an attack scenario for obtaining electronic information involved the steps of: *{Enter computer room, Steal data tape, Read tape}* then the attack could be made much more difficult by encrypting all data on tapes. The revised attack scenario would then be: *{Enter computer room, Steal data tape, Read tape, Break encryption}*. Hopefully, the *Break encryption* step would be very challenging to the attacker. The *Break encryption* procedure could be a single leaf node or, more typically, in a subtree describing various approaches to breaking encryption.

   This approach is very useful when a security analyst has been charged with securing a system that is either poorly understood or cannot be changed. Essentially, the analyst agrees to concede that the adversary will prevail against these unknown or unchangeable components. Instead of trying to fix the unfixable, the defender changes the system's architecture such that it no longer matters that the adversary will prevail against the original components. The system is protected by new, hardened mechanisms that cannot be easily subverted, and that prevent the attacker from climbing the tree to the root node (or other high level, high impact nodes).

   Note that adding a control subtree beneath an *AND* node will increase the total number of attack scenarios in the tree. This may seem counterintuitive – why would adding a countermeasure add scenarios? The number of scenarios in a tree is not a good indication of a system's level of security.

   When a control is added beneath an *AND* node, the new control brings with it a set of scenarios by which it can hypothetically be compromised or bypassed. If the control is well devised these scenarios will be difficult for adversaries to perform (which will prevent the adversary from attaining the *AND* node state). However, in order to prove that this is the case each of the *AND*'s scenarios that existed prior to the addition of the control must be analyzed in conjunction with each of the new control's scenarios. Thus the number of scenarios to reach the *AND* node increases, but they are all at least as hard (and most likely much less feasible) than the scenarios that existed before the control was added.

---

[35] If, in the original system, no *AND* node existed because only a single attack step was required, then an *AND* node would be inserted at the appropriate location and both the previously existing step, and the new additional steps, placed beneath it.

In most cases the new control is an attack tree that describes the ways in which the control might be overcome. However, another possibility is to describe the control using a fault tree. This would be appropriate in cases where the failure of the control might be due to random factors rather than hostile activity by an attacker.

For instance, a common network security tool is an *intrusion detection system (IDS)*. An IDS monitors traffic on network segments and compares the packets to a data base of malware signatures.[36] On a busy network the IDS may not be able to keep up with the data flow and packets may be skipped. The IDS hardware may fail or signature updates may result in corruption of the signature data base. Any one of these random events may result in a failure to detect a malicious packet. During the duration of the failure, a window of opportunity is available for an attacker to operate.

Note the similarity of modeling a control as a device subject to failure with the mixed capability-probability model discussed on page 44. Many network and physical security technologies have well known failure statistics and can be conveniently modeled in this way. Amenaza's Secur*IT*ree software allows analysts to explicitly identify countermeasure subtrees as behaving in stochastic fashion and treats them as fault trees.

As discussed earlier, probability formulas are defined for both *AND* and *OR* nodes in fault trees and impacts (attacker benefits and possibly victim impacts) could be associated with any node in the fault tree. Computing the potential impacts in branches involving *AND* nodes is straightforward because all of the *AND's* children must occur to satisfy the logic. So, whatever *AND* aggregation function is defined for the impact indicator (typically sum or maximum) can be used to compute the impact. Situations involving *OR* nodes are more complex.

Any non-null subset of the *OR* node's children could occur, and might affect the impact at the *OR* level. Since each of the children's probability is independent from its siblings, there is no requirement that the probabilities of the *OR's* children tally to 1 - and, in fact, they usually do not. One approach is to use a form of Monte Carlo analysis to roll the dice and see which children become active on a given trial. Then, of the subset of children that are active, further analysis is used to determine which child's impact will be chosen as that of the trial. This approach takes the view that, in the case where several of the *OR's* children become active, they would not (in the real world) all become active at the same instant in time - one would be first and that is the one that would cause the impact. Monte Carlo analysis can select which of the active children is most likely to occur first based on the active children's relative probabilities. Choosing the first active child's impact is not necessarily correct in all situations, but it seems reasonable in most situations. The Monte Carlo analysis is repeated a sufficient number of times that the impact converges toward a particular value. Other approaches may be valid.

[36] Modern IDS also use heuristics and artificial intelligence to identify malicious packets which may not have known signatures.

One way of avoiding these issues is to simply place all impacts in the stochastic countermeasure subtree's root node. Indeed, in many cases the impact of a countermeasure failure will be the same regardless of how it fails, so might be a better strategy than assigning impacts at various levels of the countermeasure subtree.

3. Use Boolean capabilistic indicators and attacker capabilities to filter attacks that will be stopped by certain defenses.

   For instance, certain leaf level activities in a tree might be technically straightforward and low cost, but only feasible for a trusted, authorized insider. These operations would have a *Breach of Trust* indicator value of *True*. The threat agent profile for an insider would reflect the insider's capability to perform these privileged operations whereas an outsider's profile would lack that capability. So, if an organization should implement special procedures to eliminate hostile insiders (background checks, regular polygraph examinations, procedures to ensure that critical activities are always performed by two randomly chosen personnel) then the countermeasure would be represented by setting the attacker's *Breach of Trust* capability to be *False*. This would prevent any of the leaf activities that require *Breach of Trust* from being performed.

These three techniques have proven to be effective in a wide variety of circumstances. However, they are implicit and may not be recognized as countermeasures by someone reviewing the model. It would be useful to be able to represent controls in a more direct fashion.

Countermeasure nodes

A variety of academic papers[37] have been published describing extensions to the attack tree model. Many of these papers make reference to *attack-defense* trees because the extended models claim to factor the defender's response to the attacker's presence once it is detected. Unfortunately, many researchers use the term *attack-defense* tree to envisage different models.

Without claiming to adhere to any of these proposed attack-defense models, the author would suggest that they can generally be portrayed as attack trees in which certain branches exist only conditionally – the branches come into existence if the presence of an attacker is detected.

A possible way of implementing such a model would be to declare certain nodes in an attack tree as *sensor* nodes. If an attack scenario traversed a sensor node as part of the path to the tree's root node, the sensor would be *tripped*. That is, the presence of the adversary would be detected. The model would associate with these sensors different countermeasures that might be deployed by the defender when the sensor was activated.

[37] A particularly good reference is the 2011 paper published by Roy, Kim and Trivedi (ACT: Towards unifying the constructs of attack and defense trees, Arpan Roy, Dong Seong Kim and Kishor S Trivedi, Security and Communication Networks, 2011; 3:1-15). In the paper, Roy et al discuss an extended tree model they call an attack countermeasure tree.

This more closely reflects reality. In many real life situations a defender cannot afford to deploy every control needed to bring all attack scenarios' risk levels to acceptable levels. A compromise is to place sensors in the environment and then deploy the control as required.

For instance, although it might be ideal for a facility to have a guard posted at every door, gate and along the fence perimeter the cost might be prohibitive. Instead, motion and infrared sensors can be placed at these locations in conjunction with comprehensive surveillance cameras. When a sensor detects activity, or a suspicious person is observed on a camera, a guard can be dispatched to investigate.

Attack Graphs vs Attack Trees

An alternate way of modeling threats and attacks is by the *attack graph*. An attack tree is a purely hierarchical structure. Nodes in an attack tree may have many children but (except for the root node) only one parent. Attack graphs relax this restriction and allow multiple parents.

One benefit of this is to permit more compact representations of the attack space. Consider the

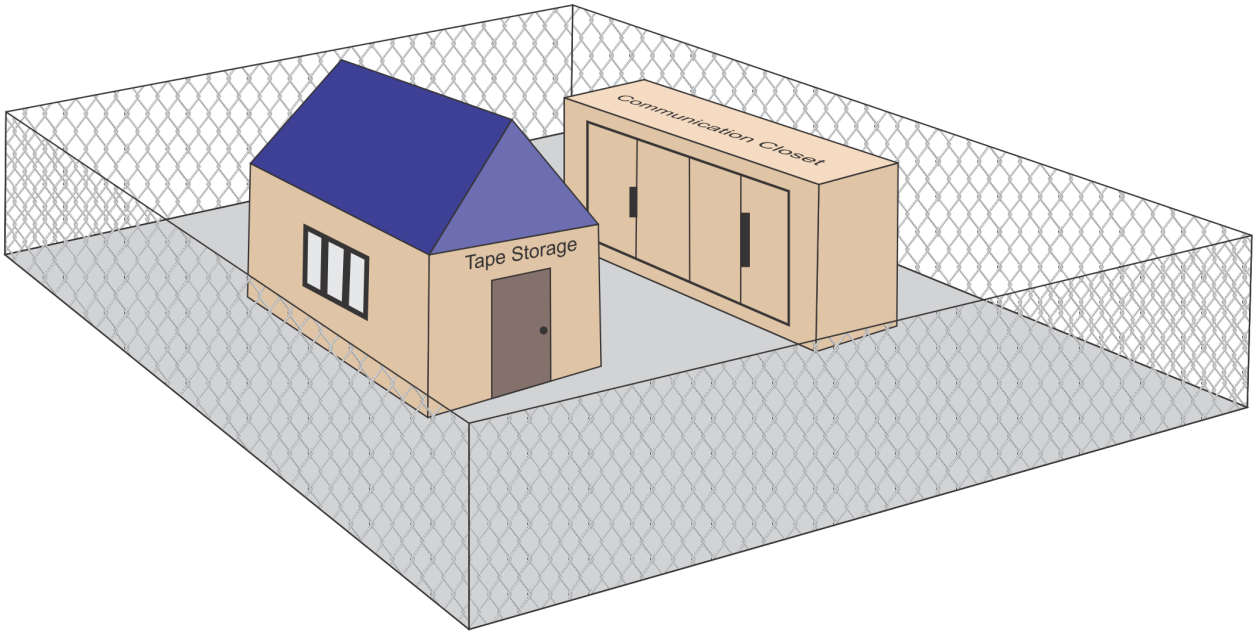


Figure 14 – Data Facility

data facility shown in **Figure 14**. It consists of a yard enclosed by a high fence. Inside the yard is a data tape storage shed and also a communications closet (where network communication cables pass). Suppose an adversary wants to steal information. There are two obvious approaches. A backup tape could be stolen or the attacker could put some type of probe on a cable in the network closet and observe the data passing by. The first steps of these attacks are the same – the attacker must first gain entry to the fenced yard. A simple attack tree describing this is seen in **Figure 15**. Note that there are two identical instances of the *Penetrate fence* subtree.

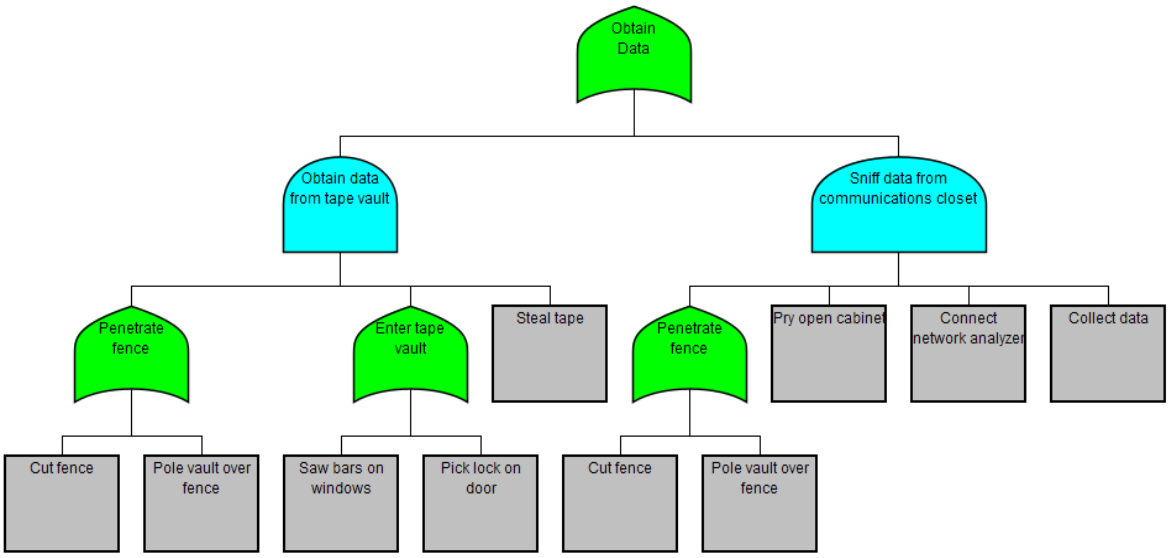


Figure 15 – Data Facility Attack Tree

It would be more succinct to display the two instances of penetrating the fence as a single occurrence (as shown in **Figure 16**). This arguably is a better representation of how an attacker might see the problem.

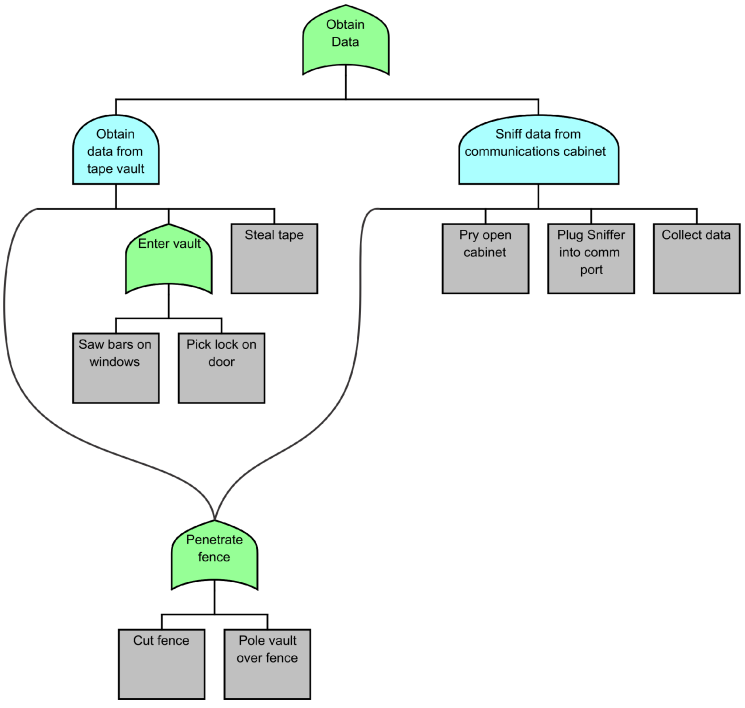


Figure 16 – Data Facility Attack Graph

An attack graph offers much greater flexibility than the strictly hierarchical attack tree. It is possible to show how attacks start at a common, low level point, then move upward through different paths (as in **Figure 16**) only to join together again (as **Figure 15**) does at the root node.

So, why do attack trees at all now that we know about attack graphs? Attack graphs are not without their drawbacks. In this simple example it was easy to draw clear vertices between the *Penetrate fence* subtree and the parents above. However, in more complex situations, the connectors can become very convoluted – looping around nodes or crossing over them. Meaningful attack graphs often look like spaghetti. So, the apparent clarity is often lost.

Neither do attack graphs reduce the number of attack scenarios that must be considered. In that sense they are no better than attack trees (but possibly deceptive in their simplicity).

Is there a way to combine the two models to get some of the benefits of each? Using the data facility situation as an example, and supposing that software were being used to create and display the models, then the *Penetrate fence* subtree might be stored a single time in memory but displayed as separate trees, possibly with an identifier showing that they are linked together.

This strategy is easiest to implement if the shared subtrees occur only at the bottom of the model. That is, they don't rejoin at some mid-level only to fork out again to multiple parents. It is a compromise between the two structures, but a useful one.

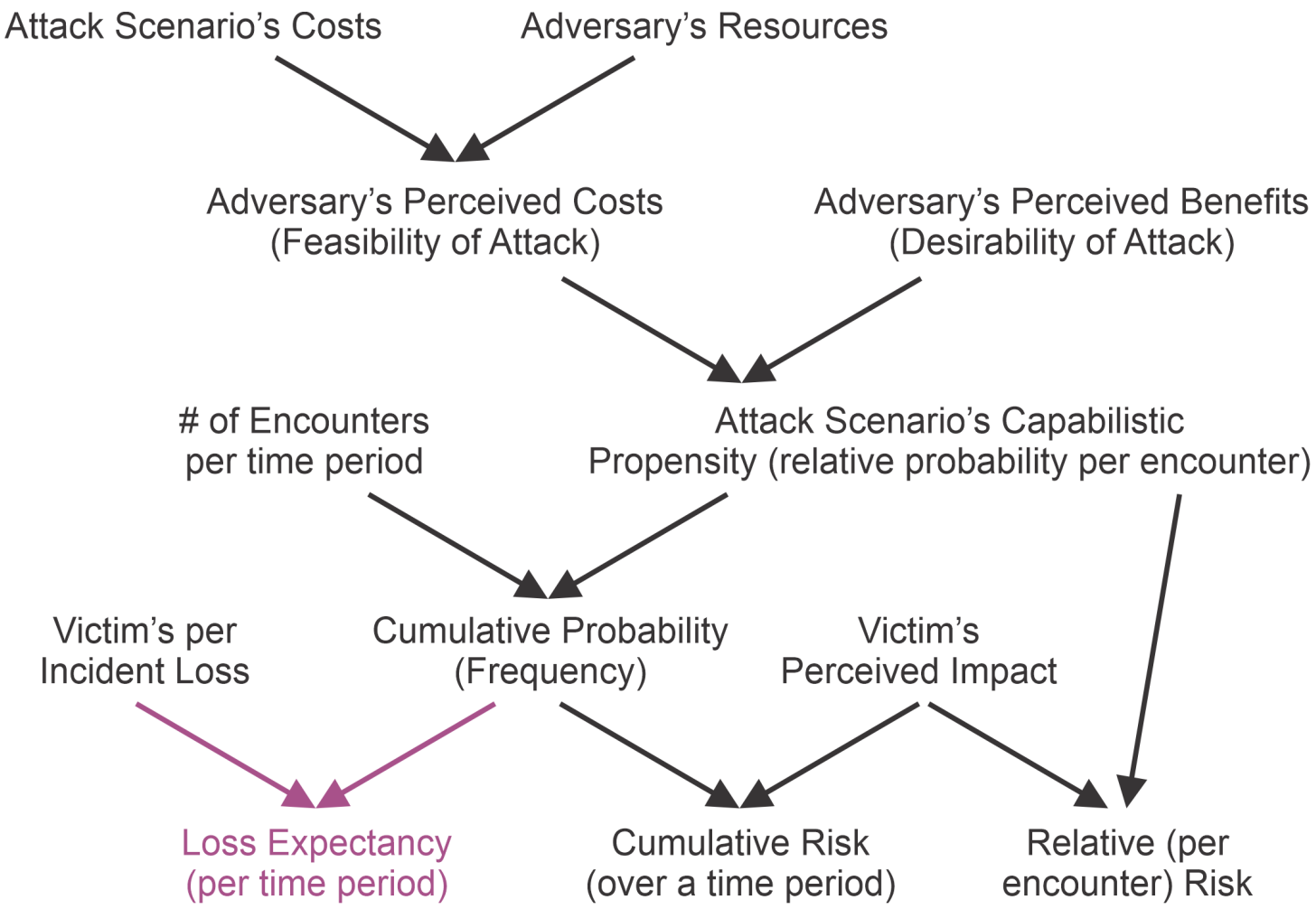
The tension between tree and graph models will likely continue.

Conclusion

There are numerous benefits to attack tree-based risk analysis. They provide an easy, convenient way to capture subject matter experts' expertise and reasoning. Analysts have stated that being forced to describe their system in an attack tree model enhanced their understanding of security issues. The clear logic of the attack tree model enhances discussions between security experts. Recording the assumptions and information that were available at the time of the analysis is valuable for proving due diligence. Most importantly, properly applied attack tree models allow analysts to turn a diverse collection of disconnected facts and assumptions into understanding.

Adversaries have long been willing to use sophisticated tools (particularly in information technology-based attacks). Hopefully the ideas presented in this paper will provide defenders with similar benefits.

Appendix I – Basic Hostile Attack Risk Analysis Flowchart



Glossary

annual loss expectancy — also known as *ALE*. The average losses associated with a given scenario over a one year time period.

attack scenario — a minimal collection of *leaf* level attack tree events that is sufficient to satisfy the *AND / OR* logic of the attack tree and result in the *root* node being achieved. Strictly speaking, an attack scenario consists only of the specified leaf nodes. Often, however, the parent *AND / OR* nodes above the leaf nodes are included in a description of the attack scenario in order to show more clearly the path taken to *root***.**

attack effectiveness — the fraction of an adversary's attempts to perform an exploit that will result in success. This can also apply to the likelihood with which an attack will advance upward past an *AND* or *OR* node given the successful fulfilment of the Boolean conditions of the node's children.

attack scenario — an attack scenario is a minimal set of leaf level events that satisfy the Boolean login in an attack tree and result in the attainment of the tree's root node goal or state.

attack tree — an attack tree is a mathematical, tree-structured diagram or model representing a system that an adversary might want to attack. The model describes the choices and goals available to an attacker. Similar to many other tree-based models, attack trees consist of a top level *root* node that represents the overall objective of the adversary (and usually what the defender wishes to prevent). There are generally a number of different approaches the attacker might use to achieve the high level goal and the diagram is extended to show these alternatives. Alternative approaches for achieving goals are denoted through the *OR* nodes. Processes or procedures are represented through *AND* nodes. The bottommost levels of the tree, *leaf* nodes, describe operations performed by potential adversaries to exploit some vulnerability in the system's defenses. If a set of *leaf* level operations cause the Boolean *AND/OR* logic of the leaf nodes' parents and ancestors to be satisfied, the high level root node goal is attained and a successful attack has occurred.

attacker benefit — the tangible or non-tangible rewards that an attacker receives upon reaching some state as they perform an *attack scenario*. Usually, but not always, the greatest benefits occur at the root node in an attack tree. Attacker benefits are what create an attacker's motivation to do attacks.

attacker detriment	the tangible or non-tangible negative effects that an attacker experiences upon reaching some state as they perform an *attack scenario.* Attacker detriments dissuade an attacker from performing an attack.
attacker motivation	an attacker is said to be motivated to perform an attack scenario if the *attacker benefits* associated with the attack outweigh the scenario's *attacker detriments*.
behavioral indicator	parameters representing the resources and abilities a *threat agent* would need to provide in order to execute an *attack scenario*. The scarcity of resources may make it more difficult for adversaries to perform a given attack, thus affecting their behavior.
capabilistic propensity	a metric of the likelihood that, given an opportunity to do so (an encounter with the target system), a *threat agent* will perform an attack scenario. The metric is based on a combination of the scenario's *attack feasibility* to the adversary and its desirability. *Capabilistic propensity* is closely related to the concepts of *relative frequency* or *relative probability* in statistics. For instance, just as there is a 1 in 6 (or 0.1667) chance that throwing a set of dice will result in a 6, an attack scenario with a *capabilistic propensity* of 0.1667 means that, for every six encounters between an adversary and a target, there is a 0.1667 likelihood they will execute the scenario (subject to caveats discussed in the text).
cumulative risk	for a deliberate, malicious event, it is the risk associated with an attack scenario for the number of encounters between the adversary and the target system anticipated in a given time period. For a stochastic event, it is the risk associated with the number of events of that type expected in the given time period. If the number of adversarial encounters or the number of stochastic events increases with time, the *absolute risk* will also increase. For that reason *absolute risk* is also known as *cumulative risk.* See also, *risk* and *relative risk.*
attack feasibility	a metric calculated by comparing the set of resources needed to carry out a particular attack scenario with a given threat agent's ability and willingness to spend those resources. The opposite of *attack feasibility* is *attack difficulty.*
exploit	(n) a detailed procedure for acting on a vulnerability; (v) to perform a procedure that takes advantage of a vulnerability.
impact	the positive or negative effects on the attacker, or the negative effects on the victim, that result from the execution of an attack

	scenario. See also *attacker benefit, attacker detriment* and *victim impact.*
pruning	a method for evaluating the likelihood and feasibility of an adversary performing a given attack scenario. If the resources required to perform the scenario are beyond the adversary's means, then the scenario is infeasible for that adversary.
relative risk	For a deliberate, malicious event, it is the risk associated with an attack scenario given a single encounter between the adversary and the target system. For a stochastic event, it is the risk associated with a single event of that type. See also, *capabilistic propensity*, *risk* and *cumulative risk.*
risk	the combination of the likelihood of an event and the resulting (usually negative) impact. See also, *cumulative risk* and *relative risk.*
threat	a potential source of danger to a system.
threat agent	a class of people who represent a threat to a system.
victim impact	the tangible or non-tangible losses that a victim experiences as a result of an attacker reaching some state as they perform an *attack scenario*. Usually, but not always, the greatest losses occur at the root node in an attack tree. The combination of an *attack scenario's* likelihood (or propensity) with it's impact on the victim is *risk.*