Investigative Tree Models

Responding to information security incidents in today s complex digital&nb
extensive preparation and planning. The current six step approach to incident response provides a broad framework but leaves the population of details up to the individual investigator. The broad unstructured approach to incident response produces inconsistent results, unpredictable time frames and uncontrolled costs. Investigative tree models provide a structured approach ...

Investigative Tree Models

How much money does it take to respond to an incident?

Author: Rodney Caudle, rodney_caudle@yahoo.com
Advisor: Rick Wanner

Accepted: April 2nd, 2009

Abstract

Responding to information security incidents in today's complex digital environment requires extensive preparation and planning. The current six step approach to incident response provides a broad framework but leaves the population of details up to the individual investigator. The broad unstructured approach to incident response produces inconsistent results, unpredictable timeframes and uncontrolled costs. Investigative tree models provide a structured approach to identifying the questions that need to be answered, identifying the location of data needed to answer the questions and prioritizing the collection of this data during incident response. The structured tree model approach to defining how questions are answered allows the incident response team to respond consistently with predictable results. The structured approach also provides for definable, reproducible structures to be created facilitating controlled cost exposure during an incident response cycle. For more information on investigative tree models and to join the community effort visit http://www.investigativetrees.com.

1. Introduction

Bruce Schneier first wrote about attack trees in his book SECRETS & LIES: DIGITAL SECURITY IN A NETWORKED WORLD (*Schneier, 2000)*. Attack trees can provide a graphical depiction of the defenses of a system and the countermeasures defending the system. Mr. Schneier describes an attack tree as providing *"a methodical way of describing threats against, and countermeasures protecting, a system" (Schneier, 2000)*. However, developing an attack tree can be a time-consuming effort for a finished product with a relatively short useful lifetime. If the attack tree is not maintained with emerging threats and defenses, the effectiveness of the attack tree will decrease quickly.

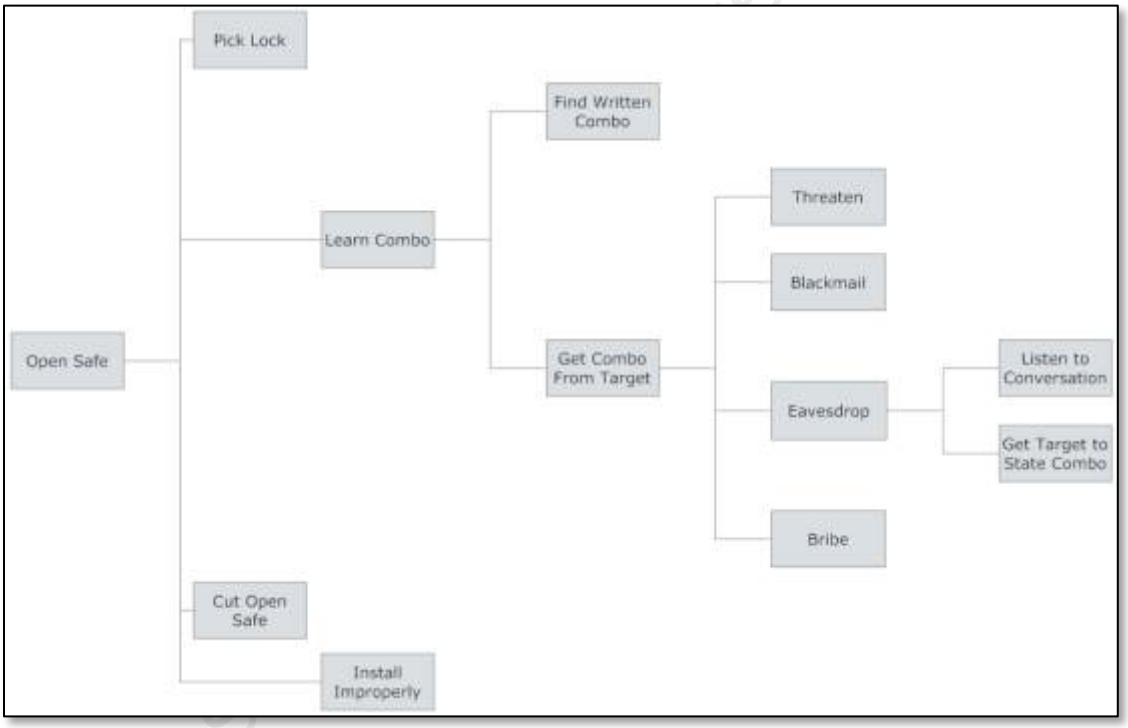


Figure 1: Attack Tree – Safe

Figure 1 provides an example of an attack tree used by Mr. Schneier (Schneier, 2000). The attack tree provides a visualization of different attacks which can be used to gain access to a safe. The attack tree is further broken down with multiple variations showing possible and impossible attacks, cost of performing an attack and other variations on the same theme.

The attack tree is comprised of four components. The first component is called the 'root node'. The root node makes up the fundamental objective represented by the attack tree.

Rodney Caudle <rodney_caudle@yahoo.com>

The second component is called a 'leaf node'. The leaf node is a node within the tree that has no children. A leaf node represents the beginning of a path to determine one possible path through an attack tree. The third component is called an 'OR-node'. The OR-node is used to join the outcome of two or more leaf nodes using a logical OR. The fourth and final component is called an 'AND-node'. The AND-node is used to join the results of two or more leaf nodes using a logical AND. These four components make up the most commonly seen nodes within an attack tree structure.

Another aspect of attack trees that needs to be discussed is the localized perspective. Every attack tree presents a perspective of the creator(s) of the tree, i.e., the objective or root node. The creator(s) are normally focused on a local problem when defining the attack tree. The example in Figure 1 provides a good representative overview for a generic safe. However, more details must be known, before the attack tree can be used to design security for a specific safe. What requirements are needed to actually pick the lock? What is the assurance surrounding the training of installation technicians? Can the safe be breached by cutting through the metal sides, top, bottom or hinges? The answers to these questions depend upon the manufacturer of the safe, model of safe and specifications of the safe. In short, additional parameters are needed for each node of the attack tree.

2. Background

More recent work on attack trees has begun to view the attack tree as a multi-variable problem. Ahto Buldas, Peter Laud, Jaan Priisalu, Mart Saarepera, and Jan Willemson put forth, in their presentation for CRITIS '06, put forth, in their paper *Rational Choice of Security Measures via Multi-Parameter Attack Trees (2006)*, that the determination of whether an IT infrastructure is protected (a) sufficiently, (b) reasonably or (c) not protected adequately can be made by using attack trees with interdependent parameters. They stated that the rational attack would occur only if the attack was profitable and the attacker would choose the attack with the highest outcome. This approach works well for rational attacks with well-known and quantifiable probabilities. However, this approach fails to adequately handle targeted attacks such as spear phishing or social engineering, or

against unknown and zero-day attacks. Never the less, this approach does provide for a mathematical approach for determine the optimal Outcome of an attack tree.

Rinku Dewri, Nayot Poolsappasit, Indrajit Ray and Darrel Whitley proposed an alternative question in their paper titled *Optimal Security Hardening Using Multi-Objective Optimization on Attack Tree Models of Network (2007)*. They proposed that the problem faced by system administrators is "*how to select a subset of security hardening measures so as to be within the budget and yet minimize the residual damage to the system caused by not plugging all required security holes." (Dewri, et. al., 2007)* Their methodology uses a set of strict rules to build the attack tree including: the restriction of a node's value to a Boolean value (True or False) and limiting an non-leaf node to only combine a maximum of two nodes. These restrictions simplify the calculations in an attack tree but cause the attack tree to increase in size and number of levels. All calculations, such as the total cost of a given path, are calculated using attributes of the nodes rather than the result or value of the node like in the previous approach used by Buldas, etc. However, this approach does provide some capabilities for handling known unknowns and targeted attacks that might be encountered in building an attack tree.

Both approaches to defining attack trees provide for representing an attack and a defensive posture (preventative or otherwise) within the attack tree structure. However, in both instances the perspective of the attack tree is to assume that the attack is occurring at the instance in time the attack tree exists. For incident response, the normal perspective is framed around an attack having occurred sometime in the past. This difference in temporal perspective can be problematic in representing attack-defend-respond within the same attack tree. However, this difference can be rectified by assuming a temporal perspective aligned with that of the attack-defend attack tree. This new perspective considers the attack occurring at that instance in time and considers questions like "What evidence is created by this action when it occurs?". This change in perspective allows alignment of information critical to incident response within the attack tree.

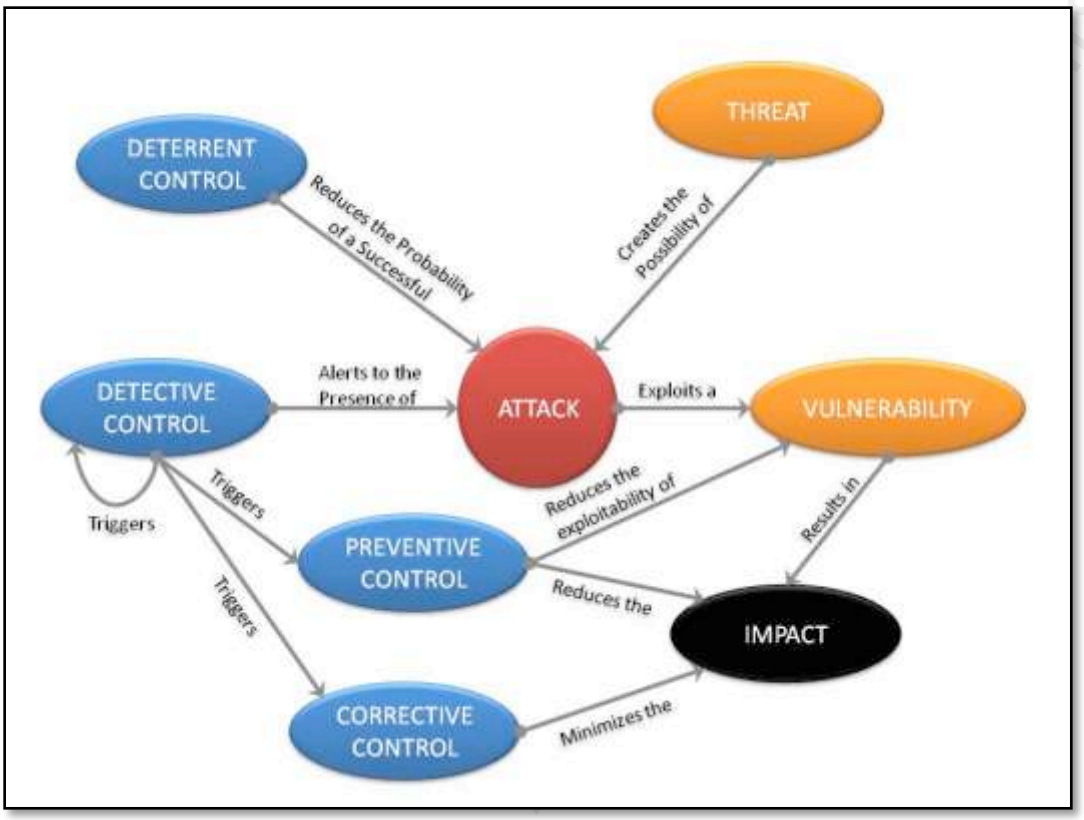
To understand the uses of adding the respond perspective to an attack tree, or more accurately, using an attack tree to build a response tree, consider that one of the primary challenges facing incident response professionals is time and a second is money. Promptly responding to an incident can yield more information due to the degradation inherent to electronic evidence. For example, the timestamps available on a subject file might be overwritten by a later attempt to access the file. Such spoliation issues are a common concern when analyzing electronic evidence. Responding quickly and accurately to an incident can reduce the monetary impact that an incident can cause.

A key criterion to effective execution is adequate preparation prior to execution. Response trees provide an organized approach to collecting the nuggets of information pertaining to a particular incident. The tree structure provides for a logical arrangement of information and methods for collecting the information. This allows the responder to make decisions and exclude avenues quickly if, in the case of a logical AND, one branch renders several others false. However, traditional tree structures are not adequate to capture the intricacies of incident response. New features and functionality need to be introduced which will handle challenges such as feedback loops, availability of information, and critical path decisions.

3. Attack, Defend, Respond

Attack, defend, respond… three sides of the same, albeit strange, coin. The attackers attack a system targeting vulnerabilities or weaknesses. The security controls provide deterrence, prevention, detection and corrective capabilities. The security control may reduce the probability of an attack occurring, provide detection of potential attacks, nullify or reduce the effectiveness of a particular attack vector, or provide assistance with minimizing the impact of an attack. The following picture provides a visualization of these relationships.

Rodney Caudle <rodney_caudle@yahoo.com>



Visualization of the Relationships of Security Controls

The detective controls present within an environment record information about the attacks detected and/or the conditions of the environment. These controls may be embedded within an application or operating system requiring another detective control to monitor the information to produce alerts. The information produced by the detective controls is critical to effective incident response. The response is an investigation into what has occurred driven by the information available, the timeframe to complete the investigation and the funding available. Changing the name from "response tree" to "investigative tree" would be more accurate[1].

3.1. Tree Models

In reality, the construction of a tree models deal with answering questions. The attack tree focuses on answering questions about the most probable avenues of attack to reach a particular outcome. The discussion above *(Dewri, et. al., 2007),* justifies the use of multiple parameters to determine an answer. Some parameters considered in an attack

[1] From this point forward the term "response tree" will be treated as identical in meaning to the term "investigative tree".

tree are available funding, timeframe to secure, profiles of potential attackers, vectors of communication and many others. Dewri, et. al., defines an attack using a mathematical statement based upon a known set of all possible attacks.

> ***Let S be a set of attributes. We define Att to be a mapping*** $Att: S \times S \rightarrow \{true, false\}$ ***and*** $Att(s_c, s_p) =$ ***truth value of*** s_p.
>
> $a = Att(s_c, s_p)$ ***is an attack if*** $s_c \neq s_p \wedge a \equiv s_c \leftrightarrow s_p$. s_c ***and*** s_p ***are then respectively called a precondition and postcondition of the attack, denoted by*** $pre(a)$ ***and post*** (a) ***respectively.***

An example is needed to understand this mathematical statement. Assume there is a vulnerability that exists on a subject system, an action that is occurring to or on the given system and a resultant condition of gaining privileged access on the subject system. Given this set of conditions, s_c is the unexploited vulnerability that exists on the subject system, a is the action that is occurring and s_p is the resultant condition of gaining privileged access on the subject system. According to the statement above there are two conditions to be met for the action a to be called an attack. The first condition calls a an attack if s_c is not the same as s_p. In the example set of conditions, this is true. The second condition calls a an attack if, given that action a is occurring, s_p can only be *true* if s_c is also *true*. However, this statement does not preclude another action from existing that might result in the same post condition. If, given action a, both conditions are true then the action a can be considered an attack.

The second type of tree model is a variation of the first type of model. Kenneth Edge, Richard Raines, Michael Grimaila and Rusty Baldwin stated in their paper "*attack trees highlight the weaknesses in a system and protection trees provide a methodical means of mitigating these weaknesses*". (Edge, et. al., 2007) Given a sample attack tree showing the possible ways to reach a particular outcome, the protection tree adds security controls to nullify a particular attack vector, minimize the impact of a particular vector, reduce the probability of an attack occurring, or detect the presence of an attack. Dewri, et. al., stated the question best in their paper *(Dewri, et. al., 2007)* as: "How do you select a subset of all possible security controls to maximize the effectiveness of your defensive

posture, stay within the given budget, and minimize the residual damage caused by not plugging all security holes?" Given the definition of an attack above, the security control must only nullify the change between s_p and s_c initiated by action a. Dewri, et. al., defines a security control as:

$$\textbf{Given an attack tree } (s_{root}, S, \tau, \epsilon), \textbf{the mapping } SC: N_{external} \rightarrow \{true, false\} \textbf{ is a security control if } \exists s_i \in N_{external} | SC(s_i) = false$$

"In other words, a security control is a preventive measure to falsify one or more attributes in the attack tree, so as to stop an attacker from reaching its goal". (Dewri, et. al., 2007). This is to say a security control, *sc*, is effective in defending against an attack *a* if, given the normal post condition (s_p) when action *a* occurs, if action *a* occurs in the presence of security control *sc*, the resultant post condition s_p is the same as the precondition s_c. Effectively this means that the action *a* can no longer be considered an attack because the two conditions mentioned previously are not met. This is a good definition for an effective security control. However, every control comes at a cost and Dewri, et. al. (2007) assigns a cost C_i to each node in the attack tree. Dewri, et. al. (2007) only provides two options for the cost of a security control: considered (assigns all cost) or not considered (assign no cost).

There are several gaps introduced by this definition that need to be identified. This definition assumes, but does not state, that only preventative controls are considered. Detective security controls are only considered as an included cost of a preventative control. In addition, this approach applies the entire cost or no cost for the security control. This is effective for an environment where the costs of security controls are fixed and well known. However, they do not consider a relationship between effectiveness of a security control and the cost of the security control. The assumption that this relationship resembles a step function instead of a spline function is inaccurate.

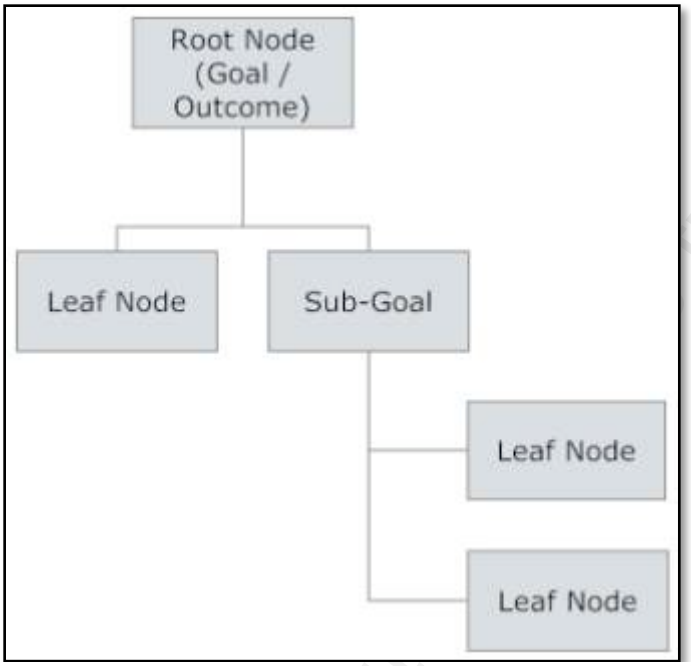
The third type of tree model, proposed in this paper, focuses on the response to an attack. The investigative tree considers a broad spectrum of questions with finer nuances. Examples of questions that need to be handled with this model of tree

include: "When did the event first occur?", "Who is responsible for this event?" and "Did a specific action occur?" However, all of the questions can be summarized by a single broad question: "Given a fixed amount of resources, what investigation will yield the results with the most confidence for a given outcome?"

Regardless of the type of tree model in use, there are certain characteristics that are consistent throughout. The following is a set of rules to follow when constructing a tree model:

1. Root node is the goal or outcome *(Schneier, 1999)*
2. Leaf nodes represent conditions which must be true to reach the desired goal , sub-goal or outcome.
3. A leaf node is considered a sub-goal if the leaf node has children nested underneath.
4. "OR" (sub-goal) nodes represent different ways to achieve the same goal *(Schneier, 1999)*
5. "AND" (sub-goal) nodes represent different steps in achieving the goal *(Schneier, 1999)*
6. Once created, additional values or parameters can be assigned to the leaf nodes allowing for multiple perspectives.
7. A leaf node may become a root node for a new tree model allowing the construction of a library.
8. Once the tree is created, different values (parameters) can be assigned to each node of the tree. *(Schneier, 1999)*

Rodney Caudle <rodney_caudle@yahoo.com>



Tree Model – Generic Components

The discussions above render the conclusion that tree models are beneficial if constructed with Boolean values for leaf nodes and logical combinations for sub-goals and the root node. The use of Boolean values in the attack tree will benefit the consumer by driving the granularity to a low enough level that important details will not be missed. In addition, the use of Boolean values provide for a vast array of tested logical methods to build into the tree structure. However, for simplicity, this paper will deal with only the "OR" and "AND" logical methods for combining leaf nodes.

For an investigation tree the most important consideration is to answer the questions posed by the investigator or client. Building the tree model using questions that can be answered by a value of true (YES) or false (NO) aligns with the structure discussed above. Once the tree structure is defined the tests and procedures that need to be executed can be identified and clearly marked to reach a conclusive result for the designated outcome.

Rodney Caudle <rodney_caudle@yahoo.com>

3.2. Node Parameters

Further expansion of the tree model through the addition of parameters allows the investigator to construct different perspectives. The parameters used can provide insight into the cost of an investigation, relative confidence in the outcome of the investigation and other perspectives that assist an investigator. Additionally, once the tree model is constructed the critical and minimal path can be identified for the investigation.

The parameters that need to be identified will depend upon the focus of the questions that need to be answered. However, a core set of generic parameters can be identified as pertaining to all tree models. The following list of parameters can provide additional information for all nodes within an investigation tree model.

Name	Description	Type
Confidence	The un-impacted confidence level as a result of the actions completed to reach this branch of the tree model.	Number (0.00 to 1.00)
$Confidence_i$	The impacted confidence level as a result of discovery of other artifacts throughout the investigation.	Number (0.00 to 1.00)
Impacted	A Boolean value reflecting whether there is an artifact that impacts the confidence level.	Boolean (0 or 1)
Weight	The weighted value of the node as compared to all peers.	Integer (0 to 100)
Category	The category of action this node represents.	String

General Parameters (all nodes)

The preceding list of parameters focus on the need for establishing a confidence level based on the results of the investigation. The calculation of the confidence

level depends upon the type of node (leaf node, root node or sub-goal). For a leaf node the confidence value is calculated by multiplying the weight by the result of the action. For a sub-goal node using a logical OR connection for the children nodes, the confidence level is calculated by taking the maximum confidence value of all children nodes multiplied by the weight of the sub-goal node. For a sub-goal using a logical AND connection for the children nodes the confidence level is calculated by taking the arithmetic mean of the children's confidence value multiplied by the weight of the sub-goal. If the tree model is constructed appropriately, as an investigation progresses the tree model the overall confidence level should become greater with the highest value resulting at the root node.

The impacted confidence value ($Confidence_i$) is the modified confidence level assuming the investigation turned up artifacts that would result in a reduced confidence level for this node or branch. To understand the concept of impact, consider the presence of a log record stating "UserA logged in to Machine X at 11:30 AM". If the investigation is focused on the account UserA's actions during a time frame that included 11:30 AM this would be a good artifact to have uncovered during the investigation. However, if the investigation turned up evidence of a log modification utility on the machine where the log record was recovered the overall confidence in this artifact should be lowered accordingly. The impacted confidence level provides a way to reflect the changing confidence through any feedback loops that may exist in the tree model. If impacted, the $Confidence_i$ value is calculated by multiplying the Confidence value of the impacted node by the impact value of the leaf node that the impact originated from.

The source of the impact is important and can be controlled using the category parameter. The category is an important parameter that allows for sorting of sub-goals and leaf nodes according to arbitrary organization strings. An example of a category for a node would be "Windows Log Record". Following the previous example of the impacted confidence due to the presence of log modification tools, a check for log modification tools could be configured to impact all confidence levels

of type "Windows Log Record". This leads to the next concept of recalculation of the tree model.

The tree model is not intended or designed to be a static value. The nature of an investigation lends itself to constant fluctuation as new artifacts are uncovered and additional impacts are calculated. However, recalculation can lead to logical loops if constructed improperly. The designer of the investigative tree model must be aware of the possibility of logical loops and take steps to avoid these mistakes prior to beginning an investigation.

With the calculation of the confidence level completed, the second most often asked question is of a financial nature. Specifically, what level of funding for an investigation will result in the highest confidence level? This question is difficult to answer in normal circumstances. The following list of parameters can provide assistance for identifying questions related to financial aspects of the investigation.

Name	Description	Type
Cost	The cost to perform the procedure.	Number (2 decimals)
Time	The elapsed time to perform the procedure. (seconds, minutes, hours)	Number (2 decimals)
Rate	The amount to bill per unit to complete the procedure.	Number (2 decimals)
Units	The number of units that this procedure represents.	Number (2 decimals)

Financial Parameters

The parameters "Cost" and "Time" apply to all nodes. The parameters "Rate" and "Units" only apply to leaf nodes. The parameter "Cost" is calculated differently depending on the type of node. For leaf nodes, the "Cost" value is calculated by multiplying the "Rate" parameter by the "Units" parameter. For sub-goal nodes, the

"Cost" is calculated by taking the maximum value of the parameter "Cost" for all the children nodes. Other variations on the parameter "Cost" can be added to calculate the maximum cost and minimum cost needed to reach the sub-goal.

The parameter "Time" for leaf-nodes can be a static value or calculated as the time required for completing a single unit. For example, completing the forensic acquisition of a hard drive is dependent on the size of the hard drive in question. The parameter "Time" for sub-goal nodes using a logical OR connection for the children nodes, is calculated as the maximum time needed to reach the sub-goal. This requires the determination of all possible paths that result in reaching the sub-goal as well as the identification of the critical path. The critical path is defined in the Project Management Study Guide as "*the longest full path on the project" (Heldman, 2007).* The critical path for reaching a particular sub-goal can be calculated following the critical path method from project management with the addition of a few additional parameters. These additional project management parameters will be discussed in the next section.

The parameter "Time" for sub-goal nodes using a logical AND connection for the children nodes, is calculated as the cumulative time required for completing all children nodes. However, the nature of the AND logical connection might yield a negative result faster if one of the children nodes has a failed result. For this reason the parameter "Time" is needed independent of the parameter "Cost".

The parameters discussed so far provide insight into answering the question of how much funding to provide a particular investigation or incident response. However, due to the nature of incident response and investigations the findings uncovered throughout the investigation may result in shifts in the scope and/or costs. To facilitate the tracking of the investigation project management parameters are needed. The following list of parameters can provide assistance for handling an investigation as a project.

Rodney Caudle <rodney_caudle@yahoo.com>

Name	Description	Type
Dependency	A list of dependent tasks required to be completed before this task can begin.	List
Early Start	The earliest value of elapsed time when this node can begin.	Number (2 decimals)
Early Finish	The earliest value of elapsed time when this node can be completed.	Number (2 decimals)
Late Start	The latest value of elapsed time when this node can be started.	Number (2 decimals)
Late Finish	The latest value of elapsed time when this node can be completed.	Number (2 decimals)
Slack Time	The amount of time available between this node and the next task.	Number (2 decimals)

Project Management Parameters

The project management parameters are focused upon the Critical Path Method as a way to control the flow of the project. Additional methods are available and can be incorporated with the addition of a different set of parameters depending upon the method chosen. However, the logical structure of the tree model lends itself to the Critical Path Method.

The Critical Path Method (CPM) provides support for schedule compression through the concept of crashing and fast tracking. To support these concepts, the parameter "Dependency" is a list of all dependencies for this node. When constructing the work breakdown structure (WBS) for the investigation these dependencies can be taken into account. In addition the dependencies will also impact the start and finish dates in the project management parameters.

There are many more parameters that can be added to the tree model to provide interpretation and methodologies. Moving forward the construction of a tree model

and identification of relevant parameters will be performed utilizing a scenario surrounding e-mail technology at a company. This example will help to understand how the selection of parameters will guide the investigator to successful cost-effective investigations.

4. Case Example: Corporate E-Mail Investigation

To facilitate the construction of a tree model used to define and target an investigation, the following steps are used in this paper. First, define the outcome (root node). Second, state any relevant information provided at the start of the investigation. Third, determine the list of attributes to be populated for each leaf node. Finally, begin populating the tree model with the appropriate questions. Continue adding levels to the tree until every question can be answered and the outcome can be rendered with a Boolean result (true or false).

This document will construct a decision tree based upon a scenario using the commonly found technology of electronic mail, or e-mail. The first step to constructing the tree model for an investigation is to choose the root node. As stated above, the root node is the outcome that is desired. This definition for the root node is very similar to the type defined for an attack tree. The root node of the example tree model will state the following:

E-mail is sent from Susan's mailbox.

The following information is relevant to the investigation. First, Susan is the CEO for her company. Second, she claims to not be the originator of the e-mail. Second, there were several e-mails sent on the date in question within a short timeframe originating from her mailbox. Finally, upon returning to the office the following day, the e-mails were not in her Sent Items, Deleted Items or other e-mail folders. These facts are critical to the investigation if, during the investigation, they can be verified through the production of artifacts.

The next step, once the outcome is defined and the pertinent information identified, is to begin populating the tree model with the questions that need to be answered and the tests associated with the answers. In this example, the immediate questions surrounding the rogue email are:

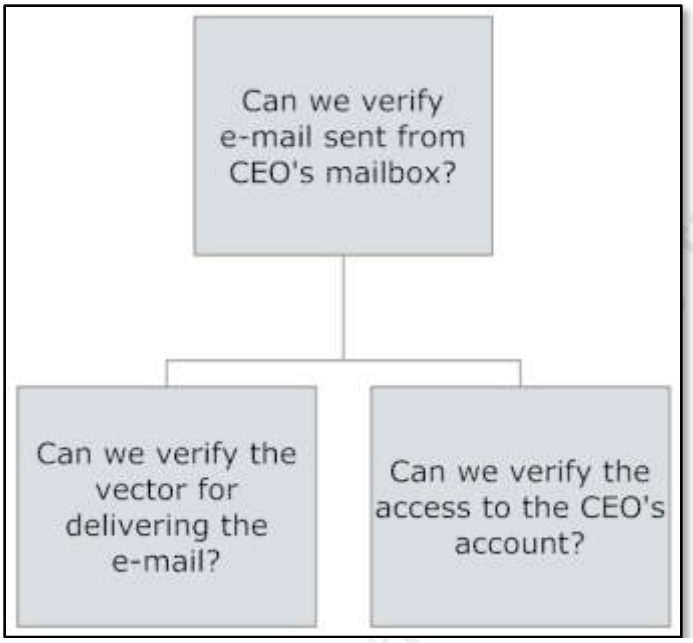
a. How were the e-mails sent?
b. Which account was used to send the e-mails?

There are other questions that need to be answered, but for the purpose of this example focusing on these two allows for a rich demonstration of building an investigation tree model. However, these questions will need to be restated as sub-goals to fit the Boolean nature of the tree model. The sub-goals are the requirements to make the outcome true. Restating the two questions as sub-goals the questions become:

a. Can we verify the vector for delivering the e-mail?
b. Can we verify access to the CEO's account?

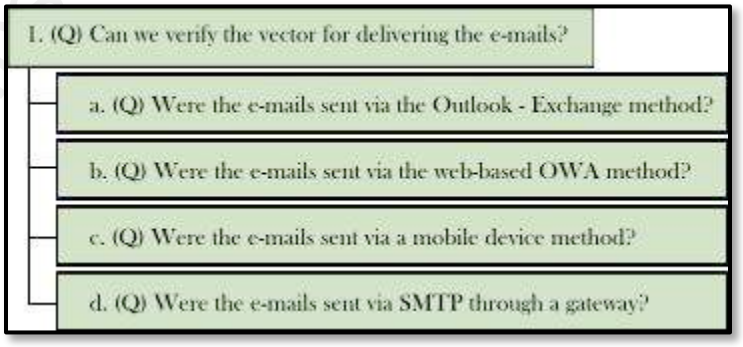
The root node is always a logical AND connection for the sub-goals. This means that both of the sub-goals need to have a value of true for the outcome to be true. This provides a target for the investigation of identifying a vector for delivering the e-mail and identifying how access to the CEO's account was accomplished.

Rodney Caudle <rodney_caudle@yahoo.com>



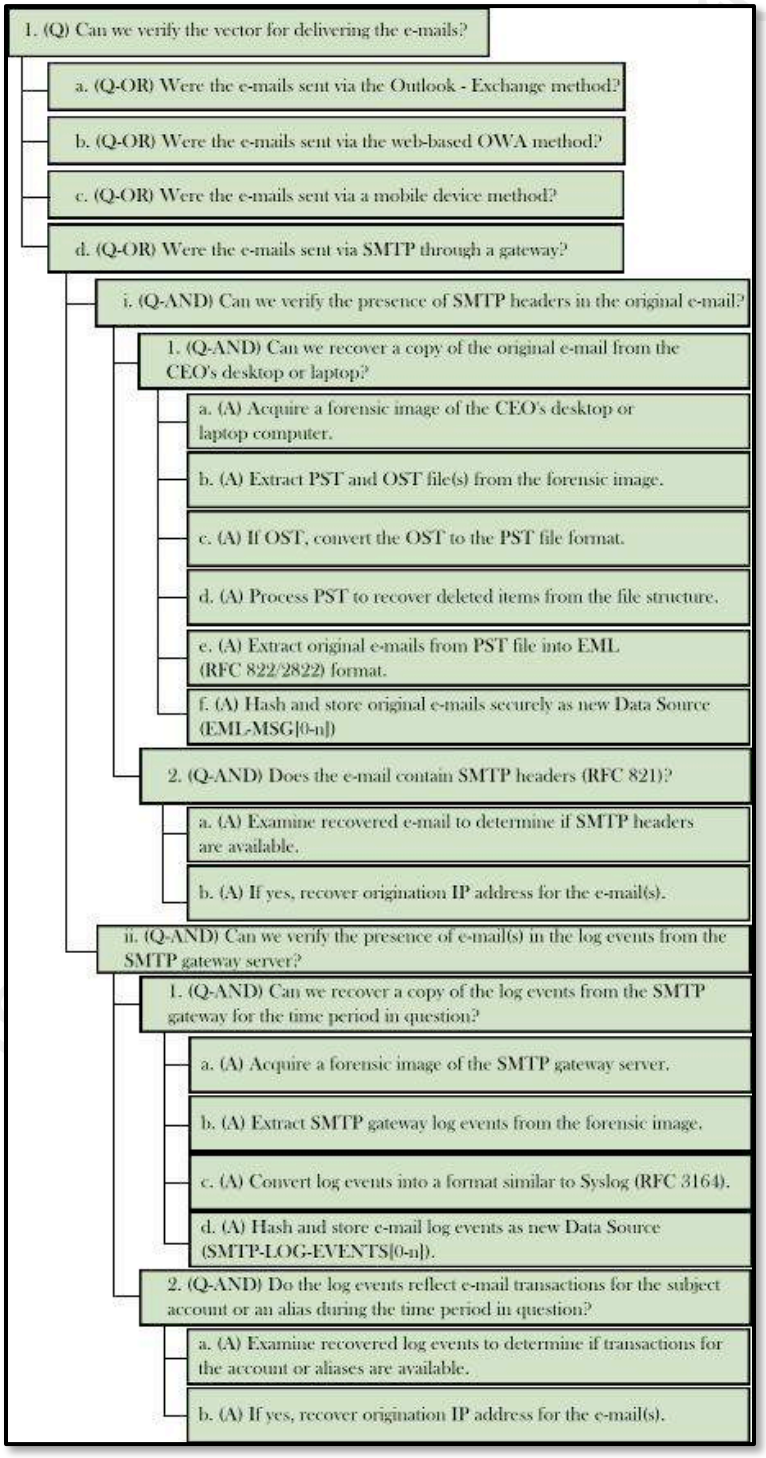
Tree Model – Corporate Email Example

Focusing on the first sub-goal, the vectors available for sending an e-mail include four possibilities. The first is to send the e-mail through an application such as Outlook 2007 which uses an RPC-based protocol to connect to a Microsoft Exchange Server. The second is to send the e-mails using the web-based OWA application. The third is to send the e-mails using a mobile device such as a blackberry. The fourth is to send the e-mails through a SMTP gateway either internal or external to the network. Only one of these methods is required to successfully send the e-mails. This makes the sub-goal a logical OR connection for the children nodes.



Continue to expand the tree structure using questions answered by Boolean responses until only tasks to create answers remain. Considering question (1.d),

determining if the e-mails were sent through a SMTP gateway can happen with several tests. The following is the tree structure with one possible branch fully populated with the remaining questions. In addition, the tests for processing the answers to the questions on the single branch have been added to the tree model.



Rodney Caudle <rodney_caudle@yahoo.com>

This sample tree will give enough details to fully flush out the parameters and demonstrate the power of tree models. The next step is to populate the parameters for the tree model. To populate the parameters start with the leaf nodes, designated by the text "(A)", and work up the tree until reaching the root node.

Selecting the visually lowest branch in the tree provides a starting point. For the sample tree model above the starting point is [1.d.ii.2]. The following is a list of parameters and values that will be used to demonstrate the concepts presented in this paper. Ensure that all dependencies are carefully considered and that all sources of values are recorded for future use.

The first set of parameters is the General parameters. The general parameters are required for all nodes regardless of type. However, the leaf nodes will have original values entered for all parameters. The sub-goal and root nodes will be calculated based on the values of the underlying nodes. From section 2.2 the list of General parameters includes Confidence, $Confidence_i$, Impacted, Weight and Category. The two parameters that are critical at this time are the Weight and the Category. Node [1.d.ii.2] is a question dealing with log events. Assigning a value of "TEXT LOG RECORD" to the Category parameter will capture this reliance upon data from a log event source.

The parameter "Weight" is populated by determining the relationship of this leaf node to all other leaf nodes. To understand, consider the value of knowing the answer to this question, then providing a ranking value (0...100). The value of this parameter will need to be refined as the tree model grows. Consider using a rolling wave planning methodology to refine the values of all Weight parameters through successive iterations. In general, the sum of all Weight parameters for leaf nodes in the tree model should total 100. For the sample tree model above, node [1.d.ii.2] will have a value of 15 applied to the Weight parameter.

The parameter "Confidence" is a calculated parameter determined by multiplying the Weight (15) by the Result. The result is the Boolean value (0 or 1) determined

by whether the children tasks were completed successfully. Upon the completion of all children tasks sufficient data should be available to deduce the value of Result. Alternatively the results should define one answer and only one answer for this question. The sample tree model will have a value of 15 for the Confidence parameter if successful and a value of 0 if unsuccessful.

For the sample tree model listed above, the following table contains the populated set of parameters. The value "calculated" means the value of this parameter will be generated from the values of other parameters. The parameter value "TBD" indicates this parameter cannot be populated until information for the dependencies has been populated.

Name	Section	Value
Confidence	General	*calculated*
$Confidence_i$	General	*calculated*
Impacted	General	*calculated*
Weight	General	15
Category	General	TEXT LOG RECORD
Cost	Financial	*calculated*
Time	Financial	*calculated*
Rate	Financial	$100 / unit
Units	Financial	1 unit / GB
Dependency	Project Management	[1.d.ii.1]
Early Start	Project Management	TBD
Early Finish	Project Management	TBD
Late Start	Project Management	TBD
Late Finish	Project Management	TBD
Slack Time	Project Management	TBD

The parameter Cost indicates the value of "*calculated*" indicating the actual value is calculated using other parameters. In this instance, the value for "Cost" is calculated

using the following equation and is dependent upon the size of the data source being used.

$$Cost = Rate * Units * Size\ of\ Data\ Source$$

Following the same type of equation the value of the parameter "Time" can be calculated dependent upon the size of the data source being used. Once the parameters are fully populated the project management parameters can be populated as well. The project management parameters revolve around the parameter "Time" and associated dependencies. In the instance above node [1.d.ii.2] depends upon the completion of node [1.d.ii.1]. Let's assume that for the example above that the node [1.d.ii.1] will take a value of six hours for the "Time" parameter at a value of 1200 for the "Cost" parameter. Assuming no additional dependencies for node [1.d.ii.1], a work rate of 25GB / hour and a log file sample of 100GB, the resulting set of parameters for node [1.d.ii.2] will be as detailed in the following table. For this node there is no slack time.

Name	Section	Value
Confidence	General	15 or 0
$Confidence_i$	General	--
Impacted	General	0
Weight	General	15
Category	General	TEXT LOG RECORD
Cost	Financial	$1000
Time	Financial	4.0 hours
Rate	Financial	$100 / unit
Units	Financial	1 unit / GB
Dependency	Project Management	[1.d.ii.1]
Early Start	Project Management	START + 6 hours
Early Finish	Project Management	ES+4 hours
Late Start	Project Management	ES

Rodney Caudle <rodney_caudle@yahoo.com>

Late Finish	Project Management	EF
Slack Time	Project Management	0

The final piece of this table is the parameters "Impacted" and "Confidence$_i$". The parameter "Impacted" reflects the presence of data points which would provide an impact to the overall confidence provided by this node. In this example, the node is of a type "TEXT LOG RECORD" so data points which discover tools used to subvert or erase text audit logs would impact this node's parameters. Conversely, the lack of a presence for these tools might provide a positive boost to the overall confidence of this node's results. The personal preference for the investigator should be reflected in the use of these parameters.

Recapping this investigation tree example, the questions answered by this example, assuming the acquisition and information was present in nodes [1.d.ii.1] and [1.d.ii.2] are the following (in reverse order):

> We verified the presence of the e-mail(s) in the log events from the SMTP gateway server by (a) recovering the log events from the server and (b) dissecting the logs to recover specific records of interest. The successful result from this node provides positive confirmation that the e-mails were sent via SMTP through a gateway providing positive identification for the delivery vector.

This example shows a simple test and verification cycle which provided focused analysis for an incident response investigation. As the investigation trees are developed and grown, the returns will be seen more quickly as answers are discovered effectively.

5. Conclusion

This paper has shown how to construct an investigative tree model to formulate a repeatable approach to incident response. The first step is to determine the root question. Following the determination of the root question, the next iterations should continue to build out the levels of the tree model until the questions are able

to provide focus to taking action. Once the tree model is populated with questions, the next step is to add tests or procedures which will provide the necessary answers. Finally, the parameters are populated for the tree model to determine the costs and time constraints. Fully populated, the tree model provides for a detailed methodology for planning and executing incident response and investigations. Enriching the tree model as the incident response team matures provides for a natural information repository.

This paper expressed the need to expend efforts to develop appropriately worded questions. Some types of questions do not lend themselves to a tree model and will need to be restated. However, the questions expressed in an investigative tree model, properly answered, can provide solid evidentiary data for further action once incident response is completed.

Finally, investigative tree models allow an incident response team to answer the question: "How much money is needed to respond to this incident?". This question is what companies are desperately seeking as an answer allows them to make informed risk-based decisions about their business.

For more information on investigative trees visit http://www.investigativetreescom.

6. Future Trends

This paper has shown the many uses of investigative tree models to formulate a repeatable approach to incident response. The Investigative tree model used in this paper was simplistic in nature even if the problem posted was not. Technologies are growing more complex as business networking environments continue to expand and merge. Even technologies as common place as electronic mail is very complicated and can involve multiple data sources. The tracking of multiple data sources across multiple geographically separate locations requires adequate preparation and planning. Investigation tree models provide a structured approach

Rodney Caudle <rodney_caudle@yahoo.com>

to retaining incident response procedures. In addition, investigative tree models, by their nature, lend themselves nicely to the use of project management techniques to manage an investigation.

Future trends toward Cloud Computing will further complicate the incident response process requiring greater investments to be made in planning and facilitating responsive procedures before incidents happen. The current trend to use outsourced environments for multiple customers already exposes clients purchasing those solutions to unexpected risks. If incident response preparation does not extend to the contract negotiations, the purchasing client may receive excessive charges from uncooperative solution providers. Investigative tree models will help prepare a company to ensure that critical data sources will be available when needed.

Rodney Caudle <rodney_caudle@yahoo.com>

References

Schneier, Bruce (2000). *Secrets & Lies: Digital Security in a Networked World.* New York, NY: Wiley Computer Publishing

Schneier, Bruce (1999). *Attack Trees*. SANS Network Security 99, New Orleans, LA.

Epperson, James (1998). History of Splines. *NA Digest,* vol. 98, no. 26. Retrieved June 20, 2009, from Netlib.org Web site: http://www.netlib.org/na-digest-html/98/v98n26.html#1

Dewri, Rinky; Poolsappasit, Nayot; Ray, Indrajit and Whitley, Darrell (2007). *Optimal Security Hardening Using Multi-Objective Optimization on Attack Tree Models of Networks*. Alexandria, Virginia. CCS '07

Blackberry (2009). Blackberry Enterprise Server for Microsoft Exchange: *Administration Guide.* Retrieved July 16, 2009 from na.blackberry.com Web site: http://na.blackberry.com/eng/deliverables/8302/BlackBerry_Enterprise_Server_for_Microsoft_Exchange-5.0-US.pdf

Heldman, Kim (2007). *PMP Project Management Professional Exam Study Guide.* Wiley Publishing, Inc. Indianapolis, Indiana

Edge, Kenneth; Raines, Richard; Grimaila, Michael; and Baldwin, Rusty (2007). *The Use of Attack and Protection Trees to Analyze Security for an Online Banking System*. IEEE Computer Society, 40th Annual Hawaii International Conference on System Sciences (HICSS '07)

Pallos, Michael S. *Attack Trees: It's a Jungle out there.*, Retrieved April 6, 2009 from http://www.bizforum.org/whitepapers/candle-4.htm

Rodney Caudle <rodney_caudle@yahoo.com>

Olzack, Tom (2006). A Practical Approach to Threat Modeling. Retrieved June April 6, 2009 from Adventures in Security Web site: http://adventuresinsecurity.com/blog/wp-content/uploads/2006/03/A_Practical_Approach_to_Threat_Modeling.pdf

Rodney Caudle <rodney_caudle@yahoo.com>

Last Updated: November 23rd, 2017

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS San Francisco Winter 2017	San Francisco, CAUS	Nov 27, 2017 - Dec 02, 2017	Live Event
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZUS	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS Khobar 2017	Khobar, SA	Dec 02, 2017 - Dec 07, 2017	Live Event
SANS Munich December 2017	Munich, DE	Dec 04, 2017 - Dec 09, 2017	Live Event
European Security Awareness Summit & Training 2017	London, GB	Dec 04, 2017 - Dec 07, 2017	Live Event
SANS Austin Winter 2017	Austin, TXUS	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS Frankfurt 2017	Frankfurt, DE	Dec 11, 2017 - Dec 16, 2017	Live Event
SANS Bangalore 2017	Bangalore, IN	Dec 11, 2017 - Dec 16, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DCUS	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Security East 2018	New Orleans, LAUS	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS SEC460: Enterprise Threat Beta	San Diego, CAUS	Jan 08, 2018 - Jan 13, 2018	Live Event
SEC599: Defeat Advanced Adversaries	San Francisco, CAUS	Jan 15, 2018 - Jan 20, 2018	Live Event
SANS Amsterdam January 2018	Amsterdam, NL	Jan 15, 2018 - Jan 20, 2018	Live Event
Northern VA Winter - Reston 2018	Reston, VAUS	Jan 15, 2018 - Jan 20, 2018	Live Event
SANS Dubai 2018	Dubai, AE	Jan 27, 2018 - Feb 01, 2018	Live Event
SANS Las Vegas 2018	Las Vegas, NVUS	Jan 28, 2018 - Feb 02, 2018	Live Event
Cyber Threat Intelligence Summit & Training 2018	Bethesda, MDUS	Jan 29, 2018 - Feb 05, 2018	Live Event
SANS Miami 2018	Miami, FLUS	Jan 29, 2018 - Feb 03, 2018	Live Event
SANS London February 2018	London, GB	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS Scottsdale 2018	Scottsdale, AZUS	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS Secure India 2018	Bangalore, IN	Feb 12, 2018 - Feb 17, 2018	Live Event
SANS Southern California- Anaheim 2018	Anaheim, CAUS	Feb 12, 2018 - Feb 17, 2018	Live Event
SANS Dallas 2018	Dallas, TXUS	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS Secure Japan 2018	Tokyo, JP	Feb 19, 2018 - Mar 03, 2018	Live Event
Cloud Security Summit & Training 2018	San Diego, CAUS	Feb 19, 2018 - Feb 26, 2018	Live Event
SANS Brussels February 2018	Brussels, BE	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS London November 2017	OnlineGB	Nov 27, 2017 - Dec 02, 2017	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced