4-30-2021

Modeling risk analysis of a layered commercial solution for a classified program when a patient attacker is present

Marsella Farnam
marselllaf@aol.com

Modeling risk analysis of a layered commercial solution for a classified program when a patient attacker is present

By

Marsella R. Farnam

Approved by:

Brian Smith (Director of Thesis)
Stanley F. Bullington
Wenmeng Tian
Linkan Bian (Graduate Coordinator)
Jason M. Keith (Dean, Bagley College of Engineering)

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Industrial and Systems Engineering
in the Bagley College of Engineering

Mississippi State, Mississippi

April 2021

Name: Marsella R. Farnam

Date of Degree: April 30, 2021

Institution: Mississippi State University

Major Field: Industrial and Systems Engineering

Director of Thesis: Brian Smith

Title of Study: Modeling risk analysis of a layered commercial solution for a classified program when a patient attacker is present

Pages in Study: 31

Candidate for Degree of Master of Science

Layered security systems pose significant challenges while attempting to monitor security related activities. The varying attributes embedded within each layer as well as the attribute interdependencies within and across layers takes measurement complexity to an exponential state. The many interdependencies at play in an interconnected infrastructure further exacerbates the ability to measure overall security assurance. Then enters the patient attacker who infiltrates one layer of this security system and waits for the opportune time to infiltrate another layer. The ability to simulate and understand risk with respect to time in this dynamic environment is critical to the decision maker who must work under time and cost constraints. This thesis seeks to improve methods for interdependent risk assessment particularly when a patient attacker is present.

DEDICATION

This thesis is dedicated to my family who has supported me over the years to achieve both my Bachelor’s and Master’s Degree, and especially to my mother who made it her life’s goal to see her children go to college. Without their belief and support in me, many things would not have been completed. I also would like to say a special thank you to Dr. Greenwood and Dr. Smith with Mississippi State University as well as Dr. Melissa Dark with Purdue University for their patience as I juggled both life and this degree.

Lastly, I would like to dedicate this work to those who may find themselves in the thick of things that they may be reminded of Isaiah 59:19 “When the enemy shall come in like a flood, the Spirit of the Lord shall lift up a standard against him.” If I have learned anything, it is that for every person who endeavors to mindlessly wreak havoc and chaos upon another human, there is someone just as persistent and tenacious to shed light, bring peace, and execute justice.

ACKNOWLEDGEMENTS

This thesis was birthed as part of a course labeled TECH 581 Problems in National Information Security taken through Purdue College of Technology. A small team of students selected Risk Analysis for Layered Solutions from a detailed list of subjects. This team of students consisted of the author of this thesis, Robert Wells, and John Zage.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Problem statement

Layered security systems pose significant challenges while attempting to monitor security related activities. The varying attributes embedded within each layer as well as the attribute interdependencies within and across layers takes measurement complexity to an exponential state. The many interdependencies at play in an interconnected infrastructure further exacerbates the ability to measure overall security assurance. Then enters the patient attacker who infiltrates one layer of this security system and waits for the opportune time to infiltrate another layer. The ability to simulate and understand risk with respect to time in this dynamic environment is critical to the decision maker who must work under time and cost constraints. This thesis seeks to improve methods for interdependent risk assessment particularly when a patient attacker is present.

1.2 Purpose statement

The foremost concern is the composite security risk of the layered solution. The first step required is identification of suitable attributes for measuring the interdependence between layers. These attributes can provide clues on developing the composition rules for relating the importance of each attribute on the entire solution's risk. From identifying the attributes, calculating their value, and assessing their impact on risk, methods can be proposed on how to combine the attributes into a single interdependent value. To identify the interdependency measurement's effect on interdependent risk, a method is proposed to describe its effect on the whole layered solution. Finally, using information discovered from the previous methods, a unique way is proposed to model the effect of time and open vulnerabilities on the security of the layered solution.

1.3 Motivation

An information system is composed of multiple assets that include hardware, software, users and infrastructure [MD1]. Cyber risk interdependence occurs when multiple assets are linked. Computers physically linked through the Internet, access of machines by other hosts through communication protocols and the use of ubiquitous technologies are interconnections that increase cyber risk. Hackers try to break into these assets through vulnerabilities and, if successful, can repeat the crime if others use the same technology. Countermeasures can be employed to limit one or multiple threats, but are often unsuccessful and their ineffectiveness could be attributed to the many uncertainties in assessing cyber risk. By accounting for the dependencies among risk factors, an organization's cyber risk factors can be more accurately measured and subsequently used in a model to outline optimal strategies for risk mitigation.

CHAPTER II

LITERATURE REVIEW

2.1 Overview:

An information security risk can be quantified as the product of the likelihood of a risk becoming a reality and the impact of a successful threat event against the information assets of an organization or an individual. Threat sources exploit one or more vulnerabilities to create the threat event. The likelihood of a threat event is determined by the number of underlying vulnerabilities, the relative ease with which the vulnerabilities can be exploited, their attractiveness for an attacker, the motivation, the resources and the capability of the attacker, and the presence and effectiveness of existing security controls. Risk analysis identifies the possible risks and estimates the likelihood and the impact of a successful exploitation.

Why is assessing information security risk so complex? Since an information system is composed of multiple assets that include hardware, software, users and infrastructure, hackers attempt to abuse these assets through vulnerabilities. Countermeasures can be employed to limit one or multiple threats. Threats can be initiated by outsiders, customers and employees. Simple linear models proposed by existing approaches are not able to capture such complexities. Many risk analysis methodologies have been developed by researchers and practitioners and can be grouped into three major categories: quantitative, qualitative and a combination of quantitative and qualitative approaches.

2.2 Risk Model Types

According to Mkpong-Ruffin [2009], assessing security risks is predominantly a qualitative process. Most practitioners use the qualitative measures of high, medium and low to describe both the likelihood and impact levels of risk. These types of models make it very difficult to generalize assessments and duplicate results since results are dependent on the assessment process. There are models that use quantitative methods, such as expected value analysis, that consider risk exposure as a function of the probability of a threat and the expected loss due to the vulnerability of the organization to this threat. Examples of these models include Annualized Loss Expectancy (ALE) and Livermore Risk Analysis Methodology (LRAM) [Guarro, 1987].

Other qualitative models use a stochastic dominance approach. These models focus on providing a specific contingency plan to prevent losses by comparing backup and recovery options used in a disaster. The expected value and the stochastic dominance models measure risk as the probability of a negative outcome due to a threat and the probability that counter measures fail to eliminate the threat. However, most security professionals think of risk as an event that either involves a negative or positive effect on achieving some objective and, because of the ambiguity, the positive effect is not modeled [Sun, 2006].

A model created by Sun extended existing methods by providing a rigorous, structured and tractable approach to risk analysis [Sun, 2006]. This approach facilitated the explicit incorporation of the complexity of risks that derive from multiple assets, multiple vulnerabilities to threats and multiple controls pertaining to a single threat. The structure of the model was provided by domain expert experience and knowledge, or it was assumed that the structure was chosen from a general well- known class of model structures. Thus the results of security risk analysis were relatively subjective [Feng]. To overcome the subjectivity, a data driven assessment model based on the knowledge from observed cases and domain experts utilizing a genetic algorithm was explored. A Bayesian network was developed to predict security risks based on historical data [Feng].

Interdependent layers have been analyzed in previous work through the use of an independence variable between layers. Independence is measured through individual critical security attributes, such as language, administrator, compiler and developer association. Postulated in the Commercial Solutions for Classified Program's report was the premise that the greater the independence among layers, the less interdependent the layers become [Martinez]. In systems with a multi-layered approach, intrusions will need to make multiple successful separate attacks. However, without analyzing the interdependence of the layered approach, the same attack could possibly be repeated to penetrate more deeply into the system.

The Open Web Application Security Project (OWASP) provides an assessment of threat risk modeling in a dedicated chapter on the website. In this assessment, five models suited for web development are outlined. This evaluation provides a good overview of commercially available models rather than models outlined in research papers referenced above. For web application design, it is essential to apply threat risk modeling to reduce the time and money spent on useless controls that fail to focus on real risks. In the online review, OWASP recommends Microsoft's threat modeling processes STRIDE and DREAD due to their value in addressing the unique challenges facing web application security and their simplicity when applied by various users. The STRIDE model classification scheme, an acronym formed from categories of web exploits, Spoofing identities, Tampering with data, Repudiation, Information disclosure, Denial of service and Elevation of privileges, characterizes known threats. These threats are not unique to web systems and can be applied to all IT systems in general.

In the article Handling IDS' Reliability in Alert Correlation, the authors provides a two prong approach when developing a risk model by not only using Bayesian network model which utilizes alerts provided by the IDS, but then applies a method to control the false alarms through controlling the confidence of the prediction model (Tabia, 2010). Since the model is using a Bayesian network, its reliability calculations are built upon historical experience. The objective of Tabia's research was to filter through the many false attacks in an IDS such that a severe attack could be determined. In order to detect a severe attack a multi-step attack detection methodology was utilized by tracking the relationship and connections of the various alerts. Filtering and prioritization is then applied, but Tabia pointed out that subject matter experts are critical in this area.

In some cases, a risk model can be developed such that it detects when an unauthorized access has been granted to a legitimate user (Aswani, 2015). In the article "Topic Modeling of SSH Logs Using Latent Dirichlet Allocation (LDA) for the Application in Cyber Security", the authors' method reviews the logs of the certain IP address to identify textual patterns such that brute-force attackers can be identified and differentiated. This method enhances the Hidden Markov Model (HMM) in that it allows for more detailed information of the what and where an attack came from through characterization of the user and IP address. When LDA is used as an addon to HMM, it creates a sort of intuition in that it characterizes the response and actions of the user historically and uses that as a cross comparison to user responses and actions in the present state such that patterns are identified that mirror a brute force attacker. In this model LDA could be developed such that is learns what is normal and any deviation to normal is then easily detected. The fact that LDA is a topic modeling technique makes it ideal for attack detection due to the cyber security world being primarily a textual data environment.

Finally, many models look for what is different in hopes of detecting and/or identifying the attacker. However, if a model was able to not only detect the attacker but to also track and assess the attacker, then additional time and energy could be spent on how and when to neutralize the attack. In the article, Worst-case analysis of joint attack detection and resilient state estimation, the authors present the theory of assessing the stealthiness of an attacker by modeling the performance of the cyber physical systems (CPS) (Forti, 2017). In this model, it assumes a worst case approach in that the attacker understands the system, the estimator, and the algorithms in use by the CPS to identify threats. The attacker works under the guise that the key CPS performance parameters must not exceed past a certain level or loss. The authors use a Bernoulli and Poisson Radom Finite Set to model the attack set while applying a Bayesian filter to resolve the issue of joint attack detection as well as resilient state estimation of the stochastic CPS. To assess stealthiness, the Maximum A posteriori Probability (MAP) attack detection was utilized. The end result yielded performance loss measurements over multiple instances such that CPS performance measurements could be taken from both a joint attack detection and state estimation standpoint. This article further solidified the robustness of using the Bayesian approach in CPS monitoring.

2.3 Risk Model Calculations

The National Research Council highlighted the need to separate out uncertainty from variability when working risk management in a decision maker role (Bier, 2013). They went on to mention that this should be part of the overall risk calculation, and recommended utilization of the two-dimensional Monte Carlo simulation to help distinguish between uncertainty and variability.

Interdependent risk measurement has been applied in multiple scenarios. In one particular journal article, it was applied to the management of inventory levels through the use of Leontief's model (Resurreccion, 2012):

$$x = Ax + c \tag{2.1}$$

In this formula x is the expected output, A is the interdependency matrix and c is the final consumption. In this scenario, A is a matrix of technical coefficients, and x and c are column vectors with z number of elements. When this calculation is synthesized, xi represents the total production of industry sector i. Furthermore, aij is the technical coefficient applied to total production requirements of sector j that is provided by production output of sector i. Lastly, ci is seen as consumption column vector which is the end user demand for sector i. The next step would be to apply this logic to a multi-layered security system.

Risk does change over time when the same threats continue to exist even with different countermeasures in place according to the journal article Managing Risk at the Tucson Sector of the U.S. Border Patrol (Levine, 2013). Risk can be assessed over time even if countermeasures change. In the below formula, *A+i,k* represents an incremented countermeasure in a given area. The change in risk from hazard *j*, if capability *k* is incremented in a specific area *i* to a certain level *A+i,k* would be shown as follows:

$$\Delta^{+}_{i,j,k} = R_{i,j}|_{A^{+}_{i,k}} - R_{i,j} \tag{2.2}$$

Ri,j|A+i,k represents the risk of expected loss in the area had the countermeasure been active. The same works in reverse. If k is reduced in a specific area *i* to the level *A-i,k*, then it would look as follows:

$$\Delta^{-}_{i,j,k} = R_{i,j}|_{A^{-}_{i,k}} - R_{i,j} \tag{2.3}$$

Risk is also affected by time to recover from an incident, and this time to recover in a dynamic model has been measured according to an article titled Modeling Uncertainties in Workforce Disruptions from Influenza Pandemics Using Dynamic Input-Output Analysis (El Haimer, 2014). In this scenario, the environment is not static but dynamic and has the ability to recover. This resilency factor is inversely proportional to recovery period. This particular model was called Dynamic Input-Output Model (DIIM) and utilized the below formula:

$$Q(t+1) = q(t) + K[A^*q(t) + c * (t) - q(t)] \tag{2.4}$$

In this formula, "the inoperability of a sector at the t + 1 equals inoperability at time t, plus the effects of the resilience of the sector". Resiliency is determined by K which represents the rate of recovery of each sector back to their initial production levels after an incident occurs. The next task is to factor in the interdependencies that exist across sectors. A combined interdependent resiliency is released by multiplying K with the inoperability product term:

$$A^*q(t) \tag{2.5}$$

This provides a way to capture interdependency of workforce inoperability of one sector that is interdependent on another sector. This shows the impact to a highly resilient sector if it is heavily interdependent on an inoperable sector following a major incident.

2.4 Defense in Depth Strategies

The protection of an entity's critical resources is a process that includes making decisions on safeguarding important infrastructures. One strategy for protecting such components involves employing a defense in depth strategy [Lippmann, Ingols, Scott, Piwowarski, Kratkiewicz, Artz, 2006]. Defense in depth can be described as an approach to make “risk-informed decisions” [Saleh, Marais, Bakolas, Cowlagi, 2010; p. 1111]. The process of making risk-informed decisions was first intellectualized by the US Nuclear Regulatory Commission [Saleh, Marais, Bakolas, Cowlagi, 2010; p. 1111]. As other industries employed its strategies, this approach to security has undergone several name changes [Seleh et al., 2010; p. 1111]. A specific example of an evolving moniker for defense in depth is the notion of layers of protection, which is an alternate name used within the chemical industry [Seleh et al., 2010; p. 1112].

The application of defense in depth requires that multiple layers of defense are established around an infrastructure and/or device to undermine adversaries while preventing accidents [Seleh et al., 2010; p. 1112]. Thus, in order to set up these traps, the deterrer must think about the assets to be protected by considering the design and operational choices [Seleh et al., 2010; 1112].

The defense in depth technique connects with the cybersecurity field to prevent would be hackers. Hence, defense in depth techniques are used to protect systems. For instance, this procedure could be used to protect resources on enterprise networks [Lippmann et al. 2006]. As a result, the defense mechanism would primarily use multiple layers of firewalls amongst the systems being protected [Lippmann et al. 2006].

2.5 Commercial Solutions for Classified (CSFC) Program

The National Security Agency (NSA) Commercial Solutions for Classified (CSFC) program was created in response to the need for NSA’s clients to use commercially readily available hardware and software to carry out their respective missions [National Security Agency, 2012]. The CSFC Program enables the approval of products by manufacturing them with defense in depth concepts. One example of the many products pertinent to defense in depth is the CSFC’s commercial off the shelf (COTS) smartphones [Buibish, Johnson, Emery, Prudlow, 2011; p. 1438]. In this specific example, the defense in depth methods applied to a smartphone ensures that the classified data being transferred from one user to another are secured. In the case of smartphones, the NSA used a defense in depth method to bolster security (Buibish 2011, p. 1438). This approach is not limited to smartphones as it can be applied to many other devices.

2.6 Commercial Off The Shelf (COTS) Products

Using COTS products improves the speed at which the government can deliver services to clients. The use of COTS is a shift from the other devices used by the government. Government off the shelf (GOTS) devices take longer to make and are costly to produce (Carney, Morris, Place 2003). Accordingly, the cost effectiveness and shortened delivery period of COTS products, in addition to the decreased amount of time it takes to deliver a product, are reasons why COTS is becoming more popular [Tran, Liu 1997, p. 361]. Thus, the efficiency provided by COTS products also enables smaller companies to compete with larger ones [Tran, Liu 1997, p. 362].

While there are many benefits to using COTS, there are also drawbacks. For example, security is a “critical technology gap” that deters many companies from using COTS [Buibish et al., 2011; p. 1434]. This security gap could prevent users who are technologically challenged from using devices that give them an advantage within a tactical environment [Buibish et al. 2011, p. 1434].

The manufacturer presents another problem with COTS products. In addition to the high costs of developing COTS products, the manufacturer is forced to keep spare parts for a specific period of time [Koch & Dreo Rodosek, 2012]. Accordingly, this becomes an issue if a product is being used beyond its shelf life. A current example of this being an issue is with military equipment, given that such equipment can be sued for about 10-20 years [Koch & Dreo Rodosek, 2012].

The actual manufacturing process to design COTS products is another decision relevant risk a user must evaluate. For example, the design and fabrication of Integrated Circuits (ICs) are commonly executed by a number of companies for one particular product to minimize expenses associate with making the product [Koch & Dreo Rodosek, 2012]. Moreover, users performing tasks with COTS devices do not have the authority to influence the manufacturing process [Koch & Dreo Rodosek, 2012].

2.7 Simulation Environment:

The next step is to develop a simulation model that will determine if the risk model will work as designed in a given environment. The first task will be to create a realistic environment. Since most of NSA data is considered classified, access to that data for purpose of this thesis would not be feasible. However, in order to achieve a realistic environment, the environment will need to be dynamic.

Upon reviewing several articles from the Risk Analysis Journal as well as the Reliability Engineering and Systems Safety, there are several methods that have been utilized to create the type of environment for running simulations. The first method is a Monte Carlo approach where random samples are taken from a probability distribution. Computations then can be made on the inputs and results aggregated [Cox, 2012, pp.1607-1629]. One journal article stated the importance of separating out uncertainty from variability and utilizing the two -dimensional Monte Carlo methodology as a simulation model [Bier, 2013, pp. 1899-1907]. Another approach is to utilize a Bayesian Model Averaging approach which allows for inferences to be made when uncertainty exists with the statistical model [Cox, 2012, pp.1607-1629]. The Bayesian approach uses the Bayes' theorem formula. This approach has two nodes that study the cause -and- effect relationship [Shin, 2015, pp. 208-217]. The child factor is focused on the cause element and the parent contains a result element of the child. Ultimately, it allows you to compare one variable with another at one moment in time.

CHAPTER III

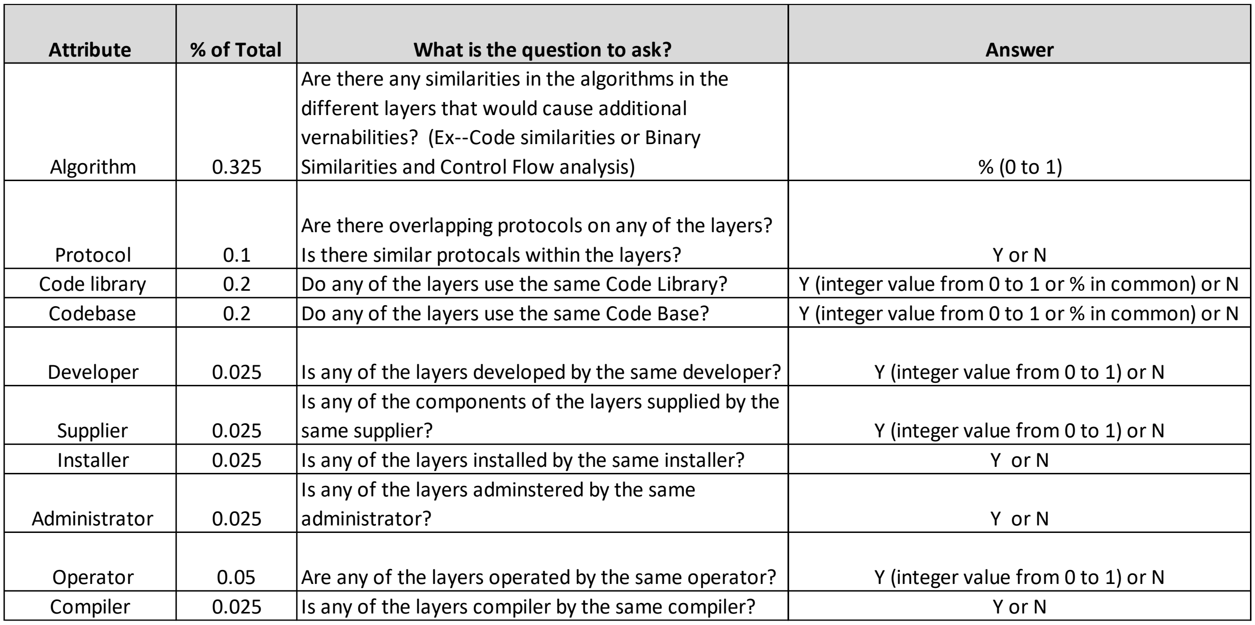
RISK MODEL AND SIMULATION

3.1 Interdependent Attributes:

The first phase involved taking a listing of interdependent attributes provided in a briefing given by NSA as part of the 2012 RSA Conference The attributes chosen for this model were selected based on their behaviors matching those expected of an interdependency measurement. The following 10 attributes were selected: algorithm, protocol, code library, codebase, developer, supplier, installer, administrator, operator and compiler. To decide which attributes were most critical, a weighting average system was calculated and then employed a Delphi study to ask NSA experts in the field their opinions of the most critical attributes of a system. Using these weights multiple attributes, each a measure of interdependency, were turned into an overall interdependence measurement.

The outcome of this task enabled the team to focus on a smaller set of interdependent attributes as shown below:

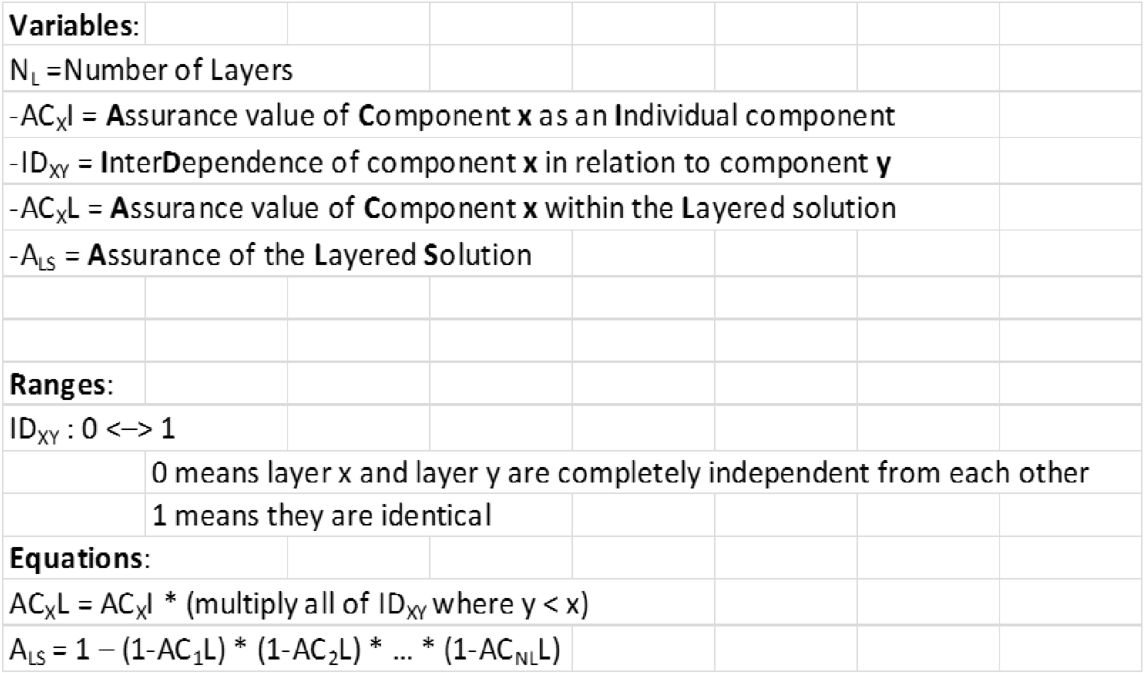
Table 3.1 Listing of Weighted Attributes



Attribute	% of Total	What is the question to ask?	Answer
Algorithm	0.325	Are there any similarities in the algorithms in the different layers that would cause additional vernabilities? (Ex--Code similarities or Binary Similarities and Control Flow analysis)	% (0 to 1)
Protocol	0.1	Are there overlapping protocols on any of the layers? Is there similar protocals within the layers?	Y or N
Code library	0.2	Do any of the layers use the same Code Library?	Y (integer value from 0 to 1 or % in common) or N
Codebase	0.2	Do any of the layers use the same Code Base?	Y (integer value from 0 to 1 or % in common) or N
Developer	0.025	Is any of the layers developed by the same developer?	Y (integer value from 0 to 1) or N
Supplier	0.025	Is any of the components of the layers supplied by the same supplier?	Y (integer value from 0 to 1) or N
Installer	0.025	Is any of the layers installed by the same installer?	Y or N
Administrator	0.025	Is any of the layers adminstered by the same administrator?	Y or N
Operator	0.05	Are any of the layers operated by the same operator?	Y (integer value from 0 to 1) or N
Compiler	0.025	Is any of the layers compiler by the same compiler?	Y or N

3.2 Calculations:

The next phase was to develop the risk model. Using the attribute weights developed in the prior chapter, a method to can now be used to combine multiple attributes, each a measure of interdependency between two layers, into an overall interdependence measurement between two layers. This however can't be used directly for calculating the assurance of a multi-layered system. To do this, a layered assurance method must be created to calculate assurance in the layered system given all interdependency measurements between every set of layers. The following calculations were utilized to accomplish this task:



Variables:

N_L =Number of Layers

-AC_XI = **A**ssurance value of **C**omponent **x** as an **I**ndividual component

-ID_{XY} = **I**nter**D**ependence of component **x** in relation to component **y**

-AC_XL = **A**ssurance value of **C**omponent **x** within the **L**ayered solution

-A_{LS} = **A**ssurance of the **L**ayered **S**olution

Ranges:

ID_{XY} : 0 <–> 1

0 means layer x and layer y are completely independent from each other

1 means they are identical

Equations:

$AC_XL = AC_XI$ * (multiply all of ID_{XY} where $y < x$)

$A_{LS} = 1 - (1\text{-}AC_1L) * (1\text{-}AC_2L) * \ldots * (1\text{-}AC_{NL}L)$

Figure 3.1 Listing of Calculations used in Simulation

3.2.2 Method for calculating Assurance value of Component x within the Layered solution (AC_XL)

Variables:

-AC_XI = **A**ssurance value of **C**omponent **x** as an **I**ndividual component

-ID_{XY} = **I**nter**D**ependence of component **x** in relation to component **y**

The value assigned is in the range from 0 to 1

0 implies layer x and layer y are completely independent from each other

1 implies they are identical

To compute the value of AC_XL the assurance value of component x as an individual component is multiplied by the product of interdependence values of layered neighbors or

$$AC_XL = AC_XI * (\text{multiply all of } ID_{XY} \text{ where } y < x). \qquad (3.1)$$

While not able to measure the risk caused by a patient attacker with the attributes, an estimate was developed for the risk of a system based on the known vulnerabilities each layer in the system has as well as the location of the layer in the system. The following is the method used to adjust the assurance based on time.

3.2.3 Time-adjusted Total Potential Vulnerability

Each identified layer (l) has a total potential vulnerability (TPV) based on the number of days since installation (D) or

$$\text{TPV } (l, D) \qquad (3.2)$$

At installation D = 0.

Each layer can also be reset or removed separately. When that is the case, then the entire total potential vulnerability is eliminated or

$$\text{TPV}(l, D) = 0. \qquad (3.3)$$

For each layer, there is a risk of break-in. This risk is measured per day and set at a constant value of .05 or

$$r(\text{BreakIn}) = .05. \tag{3.4}$$

Layers can also possess a known but unpatched vulnerability at a point in time. The point-in-time measure corresponds with a TPV time factor D. If a layer *l* on day *D* possessed an unpatched vulnerability then

$$\text{Is\_Vuln}\ (l,D) = 1. \tag{3.5}$$

If there are no known unpatched vulnerabilities the above equation is set to 0 or

$$\text{Is\_Vuln}\ (l,D) = 0. \tag{3.6}$$

To model the progression of the layered assurance, start with the install date of the level or

$$\text{TPV}\ (l, 0) = 0. \tag{3.7}$$

Beginning at level 0, the total potential vulnerability of the current day or

$$\text{TPV}\ (0,D) \tag{3.8}$$

is equal to the total potential vulnerability of the previous day(TPV (0, *D-1*)) plus any known level vulnerabilities (is_Vuln (0, *D*)) multiplied by the risk of a break in for level 0 or

$$\text{TPV }(0, D) = \text{TPV }(0, D\text{-}1) + \text{is\_Vuln }(0, D) * \text{r(BreakIn)}. \tag{3.9}$$

Consequently, to compute the potential vulnerability of the current day for any level *l*, the total potential vulnerability for *l* is (given the formula above) multiplied by the total potential vulnerability of the subsequent previous layer or

$$\text{TPV }(l, D) = \text{TPV }(l, D\text{-}1) + \text{is\_Vuln }(l, D)* \text{r(BreakIn)} * \text{TPV}(l\text{-}1, D) \tag{3.10}$$

Having calculated the total potential vulnerability or TPV (*l, D*), the **A**ssurance value of **C**omponent **x** (Layer x) within the **L**ayered solution (AC_XL, See section 2 below for the calculation of AC_XL) can now be adjusted for time adjusted vulnerabilities by multiplying AC_XL by 1 minus the total potential vulnerability calculated for the current day (*D*) or

$$AC_XLt = AC_XL * (1\text{-TPV}(x, D)) \tag{3.11}$$

giving the time-adjusted layered assurance value**.**

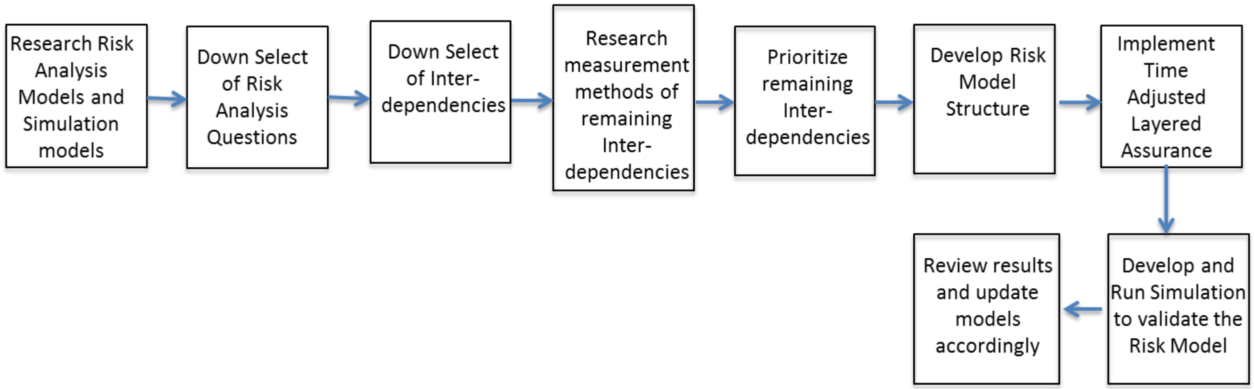
Using the time-adjusted layered assurance value, the entire Assurance of the entire Assurance of the Layered Solution (A_{LS}) was calculated. The A_{LS} is calculated by subtracting 1 minus the product of 1 minus each layer's AC_XLt, where x starts at 1 up to the maximum number of layers (N_L).

$$A_{LS} = 1 - (1-AC_{1}Lt) * (1-AC_{2}Lt) * \ldots * (1-AC_{NL}Lt) \qquad (3.12)$$

3.3 Simulation:

Now it was time to determine if a simulation could be built to not only model a patient attacker, but to also model the dynamic nature of a SCADA environment that contains a vulnerability and is able to correct the vulnerability. The following flow chart was put together to walk through the steps that were needed to build the simulation model.

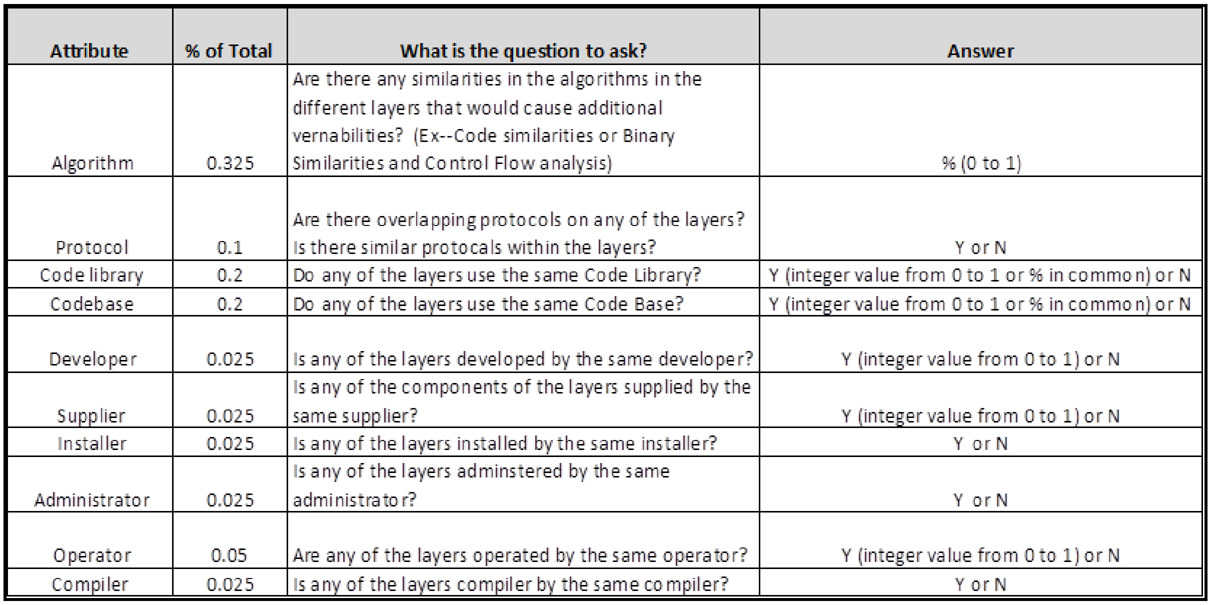
Table 3.2 Simulation Model Flow Chart


Research Risk Analysis Models and Simulation models
Down Select of Risk Analysis Questions
Down Select of Inter-dependencies
Research measurement methods of remaining Inter-dependencies
Prioritize remaining Inter-dependencies
Develop Risk Model Structure
Implement Time Adjusted Layered Assurance
Develop and Run Simulation to validate the Risk Model
Review results and update models accordingly

Prior to building the actual simulation model, it was important to define all inputs and outputs of every layer and level of the risk model. One key parameter is delineating the interdependency between different layers and levels of the chosen risk model. This can be accomplished by utilizing the Input-Output Model as introduced by Wassily Leontief [Santos, 2007, pp.1283-1297]. The I-O model allows for interconnectedness within layers and levels to be described from a quantitative perspective. Once this is clearly understood, then the next step is to utilize software such as Agena RiskPro Version 6, Matlab, or even Excel to clearly see and understand the interactions between the different layers [Shin, 2015, pp. 208-217]. For this model, Microsoft Excel was utilized.

Step 1 of building the simulation model was to incorporate the attributes that each layer of security could possibly have along with assigning each an interdependency calculation. Interdependency was identified by asking certain questions of each attribute. If the answer is no, then interdependency did not exist between that attribute and another attribute. If the answer was yes, then interdependency between attributes did indeed exist. Once interdependency was known to exist, meaning an answer was yes, the next task was to determine the percentage of interdependency by assigning a value of 0 to 1. The list of questions for each attribute and the answer type is shown in the table below.

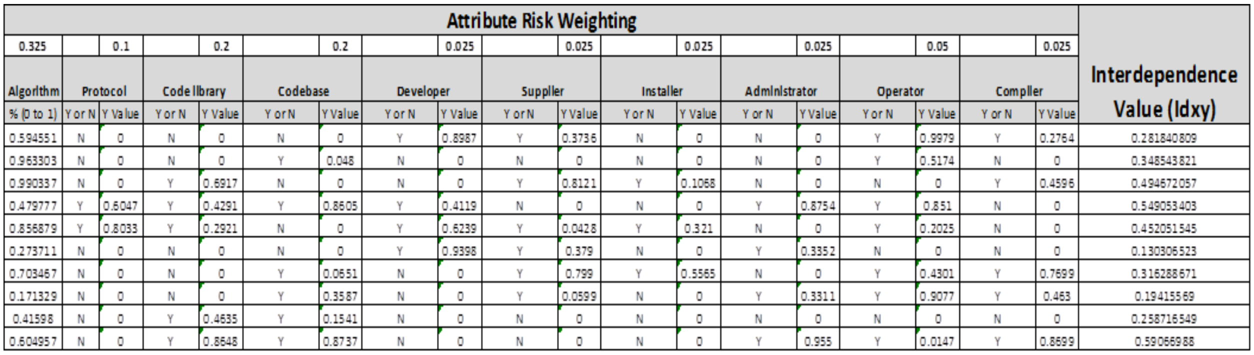
Table 3.3 List of Attribute Questions



Attribute	% of Total	What is the question to ask?	Answer
Algorithm	0.325	Are there any similarities in the algorithms in the different layers that would cause additional vernabilities? (Ex--Code similarities or Binary Similarities and Control Flow analysis)	% (0 to 1)
Protocol	0.1	Are there overlapping protocols on any of the layers? Is there similar protocals within the layers?	Y or N
Code library	0.2	Do any of the layers use the same Code Library?	Y (integer value from 0 to 1 or % in common) or N
Codebase	0.2	Do any of the layers use the same Code Base?	Y (integer value from 0 to 1 or % in common) or N
Developer	0.025	Is any of the layers developed by the same developer?	Y (integer value from 0 to 1) or N
Supplier	0.025	Is any of the components of the layers supplied by the same supplier?	Y (integer value from 0 to 1) or N
Installer	0.025	Is any of the layers installed by the same installer?	Y or N
Administrator	0.025	Is any of the layers adminstered by the same administrator?	Y or N
Operator	0.05	Are any of the layers operated by the same operator?	Y (integer value from 0 to 1) or N
Compiler	0.025	Is any of the layers compiler by the same compiler?	Y or N

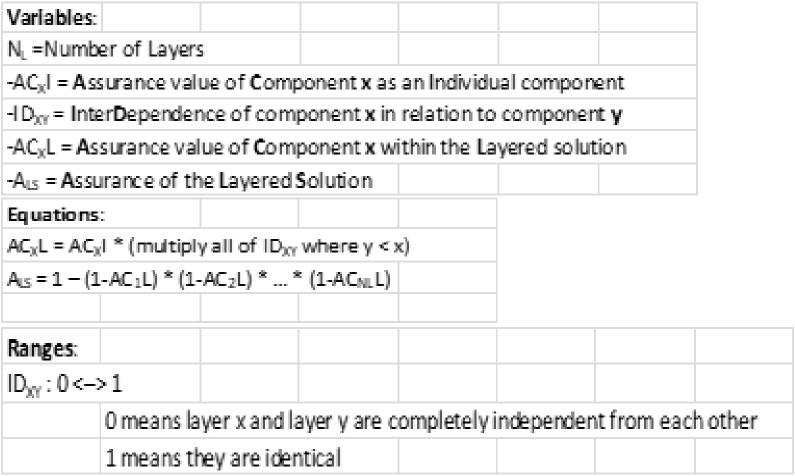
The next step was to develop a method to have the answers automatically generated by the simulation. Microsoft Excel Random Number generator was used to develop the yes or no response along with an integer value of 0 to 1 which would also represent a percent of interdependency. Since each layer of security is assumed to have 10 attributes, then every layer of security would have to be assessed. As shown in the table below, each row represents a layer of security, and each column represents one of the 10 attributes. Using the Excel random number generator, a value is assigned to each attribute on each layer, and then the total or aggregate interdependency was determined by multiplying the value across each of the attributes.

Table 3.4	Attribute Interdependence Value



Attribute Risk Weighting																			Interdependence Value (Idxy)
0.325		0.1		0.2		0.2		0.025		0.025		0.025		0.025		0.05		0.025	
Algorithm	Protocol		Code library		Codebase		Developer		Supplier		Installer		Administrator		Operator		Compiler		
% (0 to 1)	Y or N	Y Value	Y or N	Y Value	Y or N	Y Value	Y or N	Y Value	Y or N	Y Value	Y or N	Y Value	Y or N	Y Value	Y or N	Y Value	Y or N	Y Value	
0.594551	N	0	N	0	N	0	Y	0.8987	Y	0.3736	N	0	N	0	Y	0.9979	Y	0.2764	0.281840809
0.963303	N	0	N	0	Y	0.048	N	0	N	0	N	0	N	0	Y	0.5174	N	0	0.348543821
0.990337	N	0	Y	0.6917	N	0	N	0	Y	0.8121	Y	0.1068	N	0	N	0	Y	0.4596	0.494672057
0.479777	Y	0.6047	Y	0.4291	Y	0.8605	Y	0.4119	N	0	N	0	Y	0.8754	Y	0.851	N	0	0.549053403
0.856879	Y	0.8033	Y	0.2921	N	0	Y	0.6239	Y	0.0428	Y	0.321	N	0	Y	0.2025	N	0	0.452051545
0.273711	N	0	N	0	N	0	Y	0.9398	Y	0.379	N	0	Y	0.3352	N	0	N	0	0.130306523
0.703467	N	0	N	0	Y	0.0651	N	0	Y	0.799	Y	0.5565	N	0	Y	0.4301	Y	0.7699	0.316288671
0.171329	N	0	N	0	Y	0.3587	N	0	Y	0.0599	N	0	Y	0.3311	Y	0.9077	Y	0.463	0.19415569
0.41598	N	0	Y	0.4635	Y	0.1541	N	0	N	0	N	0	N	0	N	0	N	0	0.258716549
0.604957	N	0	Y	0.8648	Y	0.8737	N	0	N	0	N	0	Y	0.955	Y	0.0147	Y	0.8699	0.59066988

Once interdependency for each layer was known, the next step in the simulation was to calculate assurance. Assurance was calculated for each component or layer of the system along and then the prior interdependency calculation was tied into each layer. This interdependent value could increase, decrease, or keep the assurance value the same for the layer. A total assurance value could then be calculated for that particular security layer of the system. This same set of calculations could then be applied to all layers of security and a total assurance value cold be applied to the total layered solution or to the total system. The equations used and their definition is shown in the tables below for clarity.



Variables:
N_L =Number of Layers
-AC_xI = **A**ssurance value of **C**omponent **x** as an **I**ndividual component
-ID_{XY} = **I**nter**D**ependence of component **x** in relation to component **y**
-AC_xL = **A**ssurance value of **C**omponent **x** within the **L**ayered solution
-A_{LS} = **A**ssurance of the **L**ayered **S**olution

Equations:
$AC_xL = AC_xI$ * (multiply all of ID_{XY} where $y < x$)
$A_{LS} = 1 – (1\text{-}AC_1L) * (1\text{-}AC_2L) * … * (1\text{-}AC_{NL}L)$

Ranges:	
ID_{XY} : 0 <–> 1	
	0 means layer x and layer y are completely independent from each other
	1 means they are identical

Variables:

-N_L =Number of Layers

-AC_xI = **A**ssurance value of **C**omponent **x** as an **I**ndividual component

-ID_{XY} = **I**nter**D**ependence of component **x** in relation to component **y**

-AC_xL = **A**ssurance value of **C**omponent **x** within the **L**ayered solution

-A_{LS} = **A**ssurance of the **L**ayered **S**olution

-IxMax = maximum effect of 1 interdependence measurement on x

Assumptions:

-Assume each layer can affect an equal proportion of the assurance of other layers

-Maximum effect of interdependence would reduce the effectiveness of the layered solution to that of 1 layer

Ranges:

ID_{XY} : 0 <–> 1

Equations:

$IxMax = (AC_xI - (1 - (AC_xI\text{^}(1/ N_L)))) /(N_L -1)$

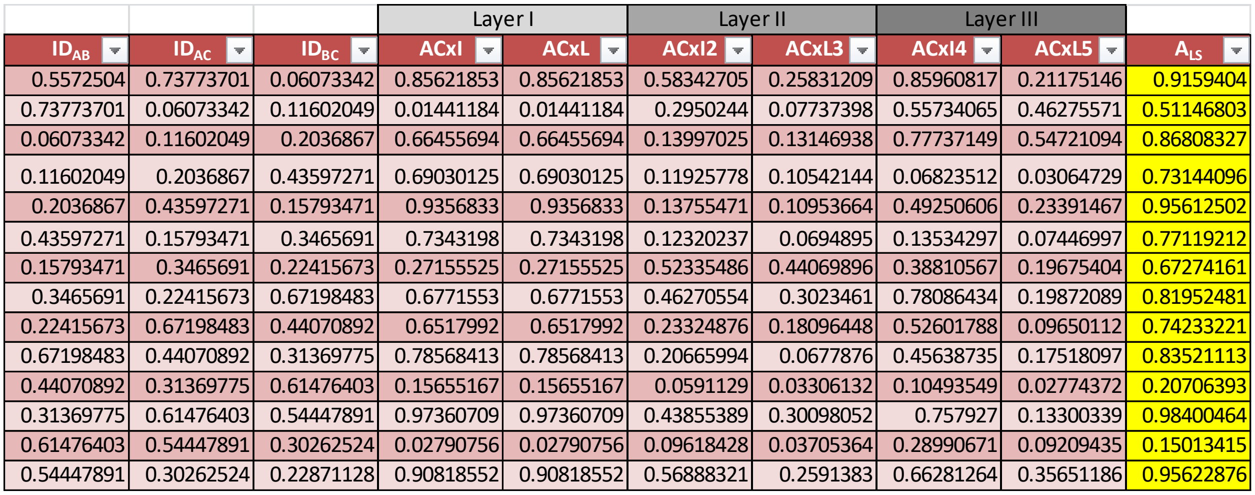
$AC_xL = AC_xI$ *(multiply all of $ID_{XY\ \text{where}\ y < x)}$

$A_{LS} = 1 – (1\text{-}AC_1L) * (1\text{-}AC_2L) * … * (1\text{-}AC_{NL}L)$

Figure 3.2 Simulation Variables

To perform the above task in Excel, the random number generator was once again applied to each of the equations above, and the end result is reflected in the below table.

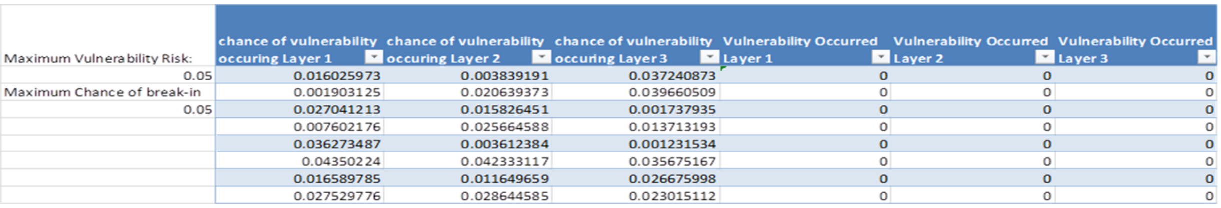
Table 3.5 Assurance Value for Each Layer



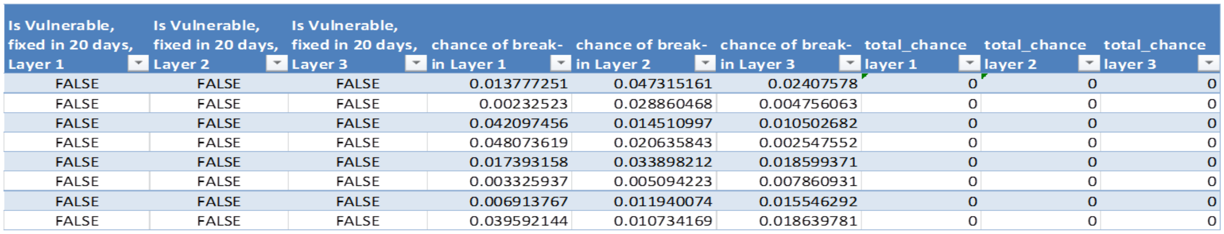
			Layer I		Layer II		Layer III		
ID_{AB}	ID_{AC}	ID_{BC}	ACxI	ACxL	ACxI2	ACxL3	ACxI4	ACxL5	A_{LS}
0.5572504	0.73773701	0.06073342	0.85621853	0.85621853	0.58342705	0.25831209	0.85960817	0.21175146	0.9159404
0.73773701	0.06073342	0.11602049	0.01441184	0.01441184	0.2950244	0.07737398	0.55734065	0.46275571	0.51146803
0.06073342	0.11602049	0.2036867	0.66455694	0.66455694	0.13997025	0.13146938	0.77737149	0.54721094	0.86808327
0.11602049	0.2036867	0.43597271	0.69030125	0.69030125	0.11925778	0.10542144	0.06823512	0.03064729	0.73144096
0.2036867	0.43597271	0.15793471	0.9356833	0.9356833	0.13755471	0.10953664	0.49250606	0.23391467	0.95612502
0.43597271	0.15793471	0.3465691	0.7343198	0.7343198	0.12320237	0.0694895	0.13534297	0.07446997	0.77119212
0.15793471	0.3465691	0.22415673	0.27155525	0.27155525	0.52335486	0.44069896	0.38810567	0.19675404	0.67274161
0.3465691	0.22415673	0.67198483	0.6771553	0.6771553	0.46270554	0.3023461	0.78086434	0.19872089	0.81952481
0.22415673	0.67198483	0.44070892	0.6517992	0.6517992	0.23324876	0.18096448	0.52601788	0.09650112	0.74233221
0.67198483	0.44070892	0.31369775	0.78568413	0.78568413	0.20665994	0.0677876	0.45638735	0.17518097	0.83521113
0.44070892	0.31369775	0.61476403	0.15655167	0.15655167	0.0591129	0.03306132	0.10493549	0.02774372	0.20706393
0.31369775	0.61476403	0.54447891	0.97360709	0.97360709	0.43855389	0.30098052	0.757927	0.13300339	0.98400464
0.61476403	0.54447891	0.30262524	0.02790756	0.02790756	0.09618428	0.03705364	0.28990671	0.09209435	0.15013415
0.54447891	0.30262524	0.22871128	0.90818552	0.90818552	0.56888321	0.2591383	0.66281264	0.35651186	0.95622876

The final step of the simulation was to account for the dynamic nature of a security system in that some vulnerabilities are fixed within a certain time frame. The first step was to assign a value which basically defined the chance of a vulnerability occurring as well as the maximum chance of a break-in in that layer. The maximum vulnerability risk factor along with the maximum chance of break-in factor was both set to 0.05. Both factors were generated using the Excel random number generator which was set between 0 and 0.05. Since this model assumed some vulnerabilities were corrected within 20 days, this value was automated by using a True/False random generator feature in Excel as well. The total value of vulnerability or break-in occurring was calculated by adding the total across each layer and determining if the value was above 0.05.

Table 3.6 Total Value of Break-in Occurring



Maximum Vulnerability Risk:	chance of vulnerability occuring Layer 1	chance of vulnerability occuring Layer 2	chance of vulnerability occuring Layer 3	Vulnerability Occurred Layer 1	Vulnerability Occurred Layer 2	Vulnerability Occurred Layer 3
0.05	0.016025973	0.003839191	0.037240873	0	0	0
Maximum Chance of break-in	0.001903125	0.020639373	0.039660509	0	0	0
0.05	0.027041213	0.015826451	0.001737935	0	0	0
	0.007602176	0.025664588	0.013713193	0	0	0
	0.036273487	0.003612384	0.001231534	0	0	0
	0.04350224	0.042333117	0.035675167	0	0	0
	0.016589785	0.011649659	0.026675998	0	0	0
	0.027529776	0.028644585	0.023015112	0	0	0



Is Vulnerable, fixed in 20 days, Layer 1	Is Vulnerable, fixed in 20 days, Layer 2	Is Vulnerable, fixed in 20 days, Layer 3	chance of break-in Layer 1	chance of break-in Layer 2	chance of break-in Layer 3	total_chance layer 1	total_chance layer 2	total_chance layer 3
FALSE	FALSE	FALSE	0.013777251	0.047315161	0.02407578	0	0	0
FALSE	FALSE	FALSE	0.00232523	0.028860468	0.004756063	0	0	0
FALSE	FALSE	FALSE	0.042097456	0.014510997	0.010502682	0	0	0
FALSE	FALSE	FALSE	0.048073619	0.020635843	0.002547552	0	0	0
FALSE	FALSE	FALSE	0.017393158	0.033898212	0.018599371	0	0	0
FALSE	FALSE	FALSE	0.003325937	0.005094223	0.007860931	0	0	0
FALSE	FALSE	FALSE	0.006913767	0.011940074	0.015546292	0	0	0
FALSE	FALSE	FALSE	0.039592144	0.010734169	0.018639781	0	0	0

Running the simulation proved that the greater the risk weighting of the attribute, the more critical the attribute is to the layered solution and the more that attribute affected layered interdependence. To determine if a list of attributes analyzed gets too large for the assessment of layered depends upon whether a risk weighting is applied to each of the attributes. As long as risk weighting is applied to each attribute, and the total weighting for all attributes is equivalent to 1, then the number of attributes could continue to grow to much larger numbers. However, being able to determine which attribute had the most impact could become harder to identify.

Another highlight of the simulation found that it is possible to measure the risk caused by a patient attacker with a set of given attributes. The simulation basically said that as time progressed, the chance of a break-in occurring on a given layer steadily increased until you started to see break-ins on other layers. One specific example showed that during the first 27 days of the simulation, no chance existed of break-in, but from day 28 to 100 each layer started to show an increasing chance of break-in. Then on day 101, one layer had a 100% chance of break-in followed up on day 163 with a 100% chance of break-in on the second layer. Lastly, day 233 showed a 100% chance of break-in on all 3 layers which rendered your layered solution useless.

CHAPTER IV

CONCLUSION

This work lends itself to discovering new ways to more accurately predict the assurance of a layered solution. Relating two layers to each other using the interdependence measurement methods listed in this thesis allows for the flexibility of fitting the model to real world cases. Certain attributes may be more relevant in certain fields than others. In addition, this thesis showcased a method of how to find the multiple layered assurance value from single entity assurance values and the interdependence values between each pair of layers. Using these methods could provide more accurate assessment for layered assurance, leading to picking better combinations of layers.

In addition, the work on relating the change of the layered assurance over time can give an idea of how often a layered solution needs to be replaced or changed. This time period can be estimated more accurately by implementers than the current research, due to their ability to use data available to them to make better predictions for the chance of a vulnerability occurring in any one day and the chance of a break-in for each day.

For future improvements, a better method for determining the weightings of attributes in the interdependence measurement could be utilized. Why are certain weights more important than others? How can the weights be estimated using analytical means rather than rational assumptions? Also, is it possible in the time adjusted assurance to include the risk from unknown vulnerabilities? Also, when a layered system is added to the overall risk model, the risk model can use methods laid out in this thesis to convert the multiple nodes of a layered system into one overall node, reducing the complexity of the overall risk model. To see an overall risk model use this work would support the validity of the layered assurance model.

REFERENCES

Annual Loss Expectancy (ALE) Calculator, Security site. com https://asecuritysite.com/Coding/ale

Aswani, K., Cronin, A., Liu, X., and Zhao, H., "Topic modeling of SSH logs using latent dirichlet allocation for the application in cyber security," 2015 Systems and Information Engineering Design Symposium, Charlottesville, VA, 2015, pp. 75-79.

Bier, Vicki M, and Shi-Woei Lin. "On The Treatment Of Uncertainty And Variability In Making Decisions About Risk." Risk Analysis: An Official Publication Of The Society For Risk Analysis 33.10 (2013): 1899-1907. MEDLINE. Web. 11 Feb. 2016.

Buibish, A. M., Johnson, N. E., Emery, D., & Prudlow, M. (2011, November). Cryptographic solutions for COTS smart phones. In *MILITARY COMMUNICATIONS CONFERENCE, 2011-MILCOM 2011* (pp. 1434-1439). IEEE.

Carney, D. J., Morris, E. J., & Place, P. R. (2003). Identifying commercial off-the-shelf (COTS) product risks: the COTS usage risk evaluation (No. CMU/SEI-2003-TR-023). CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.

Cox, Louis Anthony (Tony). "Confronting Deep Uncertainties In Risk Analysis." Risk Analysis: An International Journal 32.10 (2012): 1607-1629. Business Source Alumni Edition. Web. 6 Feb. 2015.

El Haimar, Amine, and Joost R Santos. "Modeling Uncertainties In Workforce Disruptions From Influenza Pandemics Using Dynamic Input-Output Analysis." Risk Analysis: An Official Publication Of The Society For Risk Analysis 34.3 (2014): 401-415. MEDLINE. Web. 11 Feb. 2016.

Feng, N. and X. Yu, “A Data-driven Assessment Model for Information System Secrutiy Risk Management”, Journal of Computers, Vol. 7, No. 12, 2012, doi:10.4304/jcp.7.12.3103-3109

Forti, N., Battistelli, G., Chisci, L., and Sinopoli, B. "Worst-case analysis of joint attack detection and resilient state estimation," 2017 IEEE 56th Annual Conference on Decision and Control (CDC), Melbourne, VIC, 2017, pp. 182-188.

Guarro, S.G.,”Principles and procedures of the LRAM approach to information systems risk analysis and management”, Journal of Computers and Security”, Vol. 6, No 6, 1987, pp 493-504, Elsevier Advanced Technology Publications, Oxford, UK, UK

Martinez, C , Haverkos, R. Commercial Solutions for Classified (CSfC), Risk Analysis.

Koch, R., & Dreo Rodosek, G. (2012). The Role of COTS Products for High Security Systems. NATO CCD COE .

Levine, E S, and Julie F Waters. "Managing Risk At The Tucson Sector Of The U.S. Border Patrol." Risk Analysis: An Official Publication Of The Society For Risk Analysis 33.7 (2013): 1281-1292. MEDLINE. Web. 11 Feb. 2016.

Lippmann, R., Ingols, K., Scott, C., Piwowarski, K., Kratkiewicz, K., Artz, M., & Cunningham, R.(2006, October). Validating and restoring defense in depth using attack graphs. In *Military Communications Conference, 2006. MILCOM 2006. IEEE* (pp. 1-10). IEEE.

Mkpong-ruffin, Idongesit, "*Quantitative Risk Assessment Model for Software Security in the Design Phase of Software Development*" Thesis, Auburn University May 15, 2009, http://hdl.handle.net/10415/1584

National Security Agency (2012). Commercial Solutions For Classified (CSFC) Frequently Asked Questions (Non-technical).

OWASP, https://www.owasp.org/index.php/Threat_Risk_Modeling updated 29 September 2010.

Resurreccion, Joanna, and Joost R. Santos. "Multiobjective Prioritization Methodology And Decision Support System For Evaluating Inventory Enhancement Strategies For Disrupted Interdependent Sectors." Risk Analysis: An International Journal 32.10 (2012): 1673-1692. Business Source Complete. Web. 11 Feb. 2016.

Saleh, J. H., Marais, K. B., Bakolas, E., & Cowlagi, R. V. (2010). Highlights from the literature on accident causation and system safety: Review of major ideas, recent contributions, and challenges. *Reliability Engineering & System Safety*, *95*(11), 1105-1116.

Shin, J,. Son, H., Khalil, R., Heo, G., (2015). Development of a cyber security risk model using Bayesian networks. *Reliability Engineering & System Safety*, 134(15), 208-217.

Sun, L, R. P. Srivastava and T. J. Mock, "An Information Systems Security Risk Assessment Model under Dempster-Shafer Theory of Belief Functions," Journal of Management Information Systems, Vol. 22, No. 4, 2006, pp. 109-142.doi:10.2753/MIS0742-1222220405

Tabia, K,. and Leray, P., "Handling IDS' reliability in alert correlation: A Bayesian network-based model for handling IDS's reliability and controlling prediction/false alarm rate tradeoffs," 2010 International Conference on Security and Cryptography (SECRYPT), Athens, 2010, pp. 1-11.

Tran, V., & Liu, D. B. (1997, January). A risk-mitigating model for the development of reliable and maintainable large-scale commercial-off-the-shelf integrated software systems. In *Reliability and Maintainability Symposium. 1997 Proceedings, Annual* (pp. 361-367). IEEE.