TECH 581: Problems in National Security : Final Report

Project Summary

Project Title: Risk Analysis

Project Date: January 12, 2015 – May 1, 2015

Authors: John Zage – Purdue University--Information Security—jzage@purdue.edu

Robert Wells – Purdue University—Political Science—rmwells@purdue.edu

Marsella Farnam – Mississippi State University—ISE– mrj1@msstate.edu

Problem Title: Risk Analysis of a Layered Solution

Area Description: Assessing the security of a layered solution analyzing layer interdependencies.

Key Words: Software Security, Vulnerabilities, Risk Analysis, Bayesian Model Averaging, Defense in Depth

Project Description: Security improvement is a foremost concern for many organizations and measurement is one of the most powerful techniques to monitor security-related activities. To measure security risk, many models decompose systems into isolated subsystems and recombine them for system-level measures. This approach is inadequate when analyzing complex, interdependent systems. For systems that exhibit interdependent risk behaviors, the interconnected infrastructures and their interdependencies require different treatments. It is important to assess the software security risk of individual software applications and determine the composite security risk of an entire solution or system to highlight the complex interconnected nature of security risks across the components. This project seeks to improve methods for interdependent risk assessment.

Executive Summary:

Project Title: Risk Analysis

Project Date: January 12, 2015 – May 1, 2015

Authors: John Zage, Purdue University, Robert Wells, Purdue University, Marsella Farnam, Mississippi State University

Key Words: Software Security, Vulnerabilities, Risk Analysis, Interdependent Risk, Bayesian Model Averaging, Defense in Depth

Abstract: Security is very important to individuasl and organizations and measuring the current security risk is vital to maintaining and improving the overall security of deployed systems. Multiple layers of defense and traps established around an infrastructure and/or device are often used to undermine adversaries. When multiple layers are used, it is referred to as a bundled solution. While it is important to assess the software security risk of individual software applications, assessing the composite security risk of an entire solution or system is even more important. The composite assessment will allow ecurity analysts to quantify the resulting change in the security risk with the addition and/or deletion of applications within the system solution.

The composite risk of a system can be measured by determining the composite assurance of the layered solution. There is an inverse relationship between risk and assurance. As risk is reduced, the degree of assurance increases proportionally and vice versa. To measure the composite assurance, we identified relevant attributes corresponding to the assurance strength of an individual application, assessed the impact of interaction between entities and finally calculated an overall composite assurance value. After selecting ten attributes, we created an experimental method to combine these measurements into one relational value. A comparison is made between every layer and another experimental method converts the individual assurance values from each component with their interdependency relational values to other

components into an overall assurance value. Lastly, we created a third experimental method to adjust the overall assurance value based on time-deployed and open vulnerabilities.

Table of Contents

1. Introduction

Problem statement

Security improvement is a foremost concern of many organizations and measurement is one of the most powerful techniques to control an activity. Therefore, it is important to assess the security risk of individual applications and determine the composite security risk of an entire system. This assessment will allow analysts to quantify the resulting change in the security risk with the addition and/or deletion of applications within the solution. To determine the composite security risk, measurable attributes and composition rules to combine the individual interdependent risk measurements must be identified.

Purpose statement The foremost concern is the composite security risk of the layered solution. We first identified suitable attributes for measuring the interdependence between layers. These attributes provided 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 were proposed on combining the attributes into a single interdependent value. To identify the interdependency measurement's effect on interdependent risk, we will propose a method to describe its effect on the whole layered solution. Finally, using information discovered from the previous methods we proposed a unique way to model the effect of time and open vulnerabilities on the security of the layered solution.

Motivation An information system is composed of multiple assets that include hardware, software, users and infrastructure [MD1]. Cyberrisk 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 cyberrisk. 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 cyberrisk.

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.

2. Literature Review:

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 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 attemp 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.

a. Risk Models

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 create 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.

b. 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 indurstries 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].

c. 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.

d. 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].

e. 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 class 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.

f. Simulation Model:

The second major task is to build the actual simulation model by first defining 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 will be to utilize modeling 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].

3. *Methods and Procedures:*

The goal of our research is to understand the risk over time when a patient attacker has penetrated one layer of a layered security solution and waits for a vulnerability to open on another layer. The team went through three main steps to provide real data that supported the use of the risk model. The first step

was Exploratory research which was primarily focused on risk analysis from an interdependent and dependent perspective when working with a commercial layered solution. The primary resources used in this stage are Risk Analysis, Reliability Engineering and System Safety, and IEEE publications. At the end of the Exploratory stage, the team had developed the list of questions below in an attempt to bound the overall goal of the project.

1.Basic Question for layered solution

a. How do attributes affect layered interdependencies?

b. Which attributes are more critical to the layered solution?

c. When does the list of attributes analyzed get too large for the assessment of layered assurance?

d. Is it possible to measure the risk caused by a patient attacker with a set of attributes?

2. Importance of attributes

a. Does each attribute play a different role for overall interdependence, and should they be weighed according to their importance?

b. How can a shared administrator threaten the layered solution?

c. How will classifying the importance of each attribute aid in measuring the risk caused by a patient attacker?

3. Lifecycle Phases (Including importance of attributes)

a. Do dependencies create different risks depending on what phase of the lifecycle they are in?

b. If each lifecycle phase has different risks, does analyzing attributes within each phase increase the accuracy of the measurement of assurance?

c. By defining dependencies between attributes and risks based on their lifecycle phase, are we able to define attributes given the risk we wish to evaluate (the patient attacker risk)?

The next step was Constructive Research. This step involved three different phases. The first phase was to select a set of the questions above as a starting point for project scope conversations with NSA Tech Directors involved with Risk Analysis. These conversations were part of an every-other-week teleconference with the Risk Analysis team members, Mississippi State University and Purdue professors, and NSA Tech Directors. This team agreed to questions 1 and 2 above, but the focus was further narrowed to specifically answer question number 1. This decision allowed for more focus on the interdependent risk and risk over time caused by a patient attacker with a set of attributes.

The next phase involved taking a listing of interdependent attributes provided in a briefing given by NSA as part of the 2012 RSA Conference. In order to perform a select from the list, the team had to complete the following steps:

a) Define each of the attributes
b) Research the use of each attribute in cyber security attacks
c) Select based upon definition and research
d) Review decision of selections with Dr. Dark, Dr. Morris, and NSA Tech Director
e) Finalize down select list using feedback from professors and NSA Tech Directors.

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

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 vunerabilities? (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? IAre 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
Codebase	0.2	Do any of the layers use the same Code Base?	Y (integer value from 0 to 1 or
Developer	0.025	Are any of the layers developed by the same developer?	Y (integer value from 0
Supplier	0.025	Are any of the components of the layers supplied by the same supplier?	Y (integer value from 0
Installer	0.025	Are any of the layers installed by the same installer?	Y or N
Administrator	0.025	Are 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
Compiler	0.025	Are any of the layers compiler by the same compiler?	Y or N

The final phase was to develop the risk model. This started with developing the attribute list so that we had a risk weighting applied to each attribute as well as develop questions that would be asked regarding each attribute within a layered solution as shown in the above table. To further enhance the risk weighting score, a Delphi study was completed by utilizing NSA Tech Director inputs into the overall risk weighting score. The results of that study can be found in a separate section within this report. Once this was completed, the next step involved developing calculations that took each of the attributes and their weightings into account. The following list of calculations were utilized within the risk model:

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

The third step was empirical research where the risk model was validated through simulation. Running the simulaton 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. Answering 1c, 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 weigthing 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. Our 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.

a. Procedure:

There were multiple steps involved within this semester's project work in Risk Analysis. The below flow chart walks through the various steps that were completed to ensure measurable success on the project:

At the time of the mid-term progress report, only half of the above tasks had been completed, but to date all tasks have been completed. There are a few steps that are needed to proceed further with this project, and those are outlined in the Future Work section of this report.

b. Schedule:

The below schedule was utilized to ensure success of the project. Accomplishment of the tasks below were kept on track through bi-weekly telecons which included Risk Analysis team members, Mississippi State University and Purdue professors, and NSA Tech Directors. The schedule utilized for this semester is approximately as shown below:

Feb 13, 2015 Expanded literature review finished

Feb 25, 2015 Progress report draft and draft presentation finished

March 6, 2015 Finalize attribute listing

March 27, 2015 Finalize risk model calculations

April 3, 2015 Begin simulation set-up

April 10, 2015 Develop time adjusted factor for risk model and simulation

April 17, 2015 Complete simulation and draw conclusions

April 30 Complete final report

May 8, 2015 Final presentation and project Power Point presentation

c. Deliverables:

There were several deliverables provided as part of the Risk Analysis team's research, and they include :

1. Delphi study on attribute risk weighting values
2. Development of a risk model
3. Creation of a simulation to test out the risk model
4. A report discussing the developed interdependent risk model
5. A final powerpoint which provides a high level view of the project
6. A poster outlining the risk model and the simulation results

d. Limitations and Delimitations:

The main limitation was time constraints. However, the following items could be part of next semester's project team's work. This includes a full in-depth analysis of all calculations used and possible engagement of a mathematician to valide that all calculations are being used in the appropriate context. Also, the additional questions that were not answered as part of this semester's research could be researched further and added to content provided in this report.

A delimitation was not being able to test the risk model in a real environment. The simulation showed that the risk model could work and provided additional logic into how it should be set up. However, more could be gained by deploying the study in a real environment where additional scenarios could be assessed.

4. Findings

a. Overview

The attributes chosen for our model were selected based on their behaviors matching those expected of an interdependency measurement. We selected the following 10 attributes: algorithm, protocol, code library, codebase, developer, supplier, installer, administrator, operator and compiler. To decide which attributes were most critical, we arranged a weighting average system and then employed a Delphi study to ask experts in the field their opinions of the most critical attributes of a system. Using these weights we are able to turn multiple attributes, each a measure of interdependency, into an overall interdependence measurement.

For layered systems comprised of more than 2 layers, we needed a method to apply interdependence measurements which are only a comparison between 2 layers. To do this, we created a practical approach to reducing the assurance value of each layer depending on that layers relations to other layers as well as its location in the layered system. As a final step, we developed a method to account for the passing of time and the increasing likelihood of a break-in with known vulnerabilities.

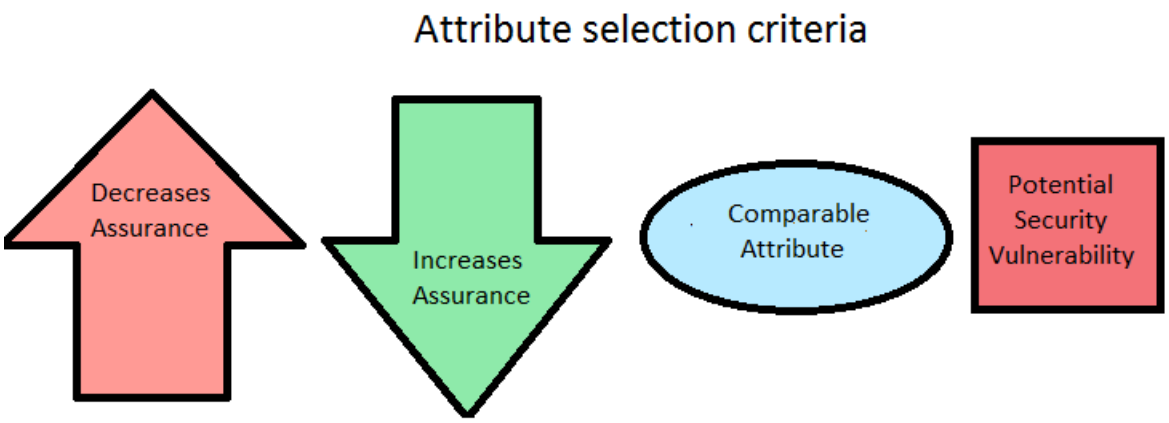
b. Detailed Findings

Our initial research questions were the following:

1. How do attributes affect layered interdependencies?
2. Which attributes are more critical to the layered solution?
3. Which attributes are more critical to the layered solution?
4. Is it possible to measure the risk caused by a patient attacker with a set of attributes?

Our answer to question 3 was to select a list of attributes that correlate to what we would expect to see for a measurement of interdependence. When we have two layers with a certain attribute that is similar, we would expect to see a degradation in assurance. Similarily, using two layers with the same attribute being completely independent, we should be able to see a potential increase of security. We need to be

able to compare two attributes together from the two layers to find a measurable difference, and also need to be able to list potential security vulnerabilities than can occur due to those attributes. Our resulting list included the following:



10 attributes:

1. Algorithm
2. Protocol
3. Code Library
4. Codebase
5. Developer
6. Supplier
7. Installer
8. Administrator
9. Operator
10. Compiler

Once we had our list of important variables, we used a weighted average for the calculation of our overall interdependence. We assumed each attribute of interdependence would have a range from 0 to 1, from completely independent to completely interdependent. To reach an answer to question 1, we

employed a Delphi study to understand which attributes were more critical to the layered solution. We employed a modified Delphi study that only consisted of one round due to time constraints. Our Delphi study solicited expert participants from the INSuRE project via email. This initial inquiry asked these individuals to use 100 points to rank the relative importance of ten variables we sought to include in risk analysis simulation model. This method is similar to other research that has requested experts to rank the importance of a subject [Chipchase, Schabrun, Cohen, Hodges, Ridding, Rothwell, & Ziemann, 2012)].

		Algorithm	Protocol	Code Library	Codebase	Developer	Supplier	Installer	Administrator	Operator	Compiler	Total
Neal Z.		5	5	20	25	15	10	5	10	5	0	100
Adam H.		14	14	14	14	12	4	8	8	10	2	100
Matt B.		10	5	5	20	20	5	10	10	10	5	100
Kyle C.		5	5	10	15	20	1	10	15	15	4	100
Michael H.		5	15	20	15	5	5	5	10	15	5	100
Rick L.		10	10	15	15	15	10	10	5	5	5	100
Wayne P.		7	10	8	8	11	6	12	14	17	7	100
Chris J.		20	0	5	10	10	5	5	20	20	5	100
	Average	9.5	8	12.125	15.25	13.5	5.75	8.125	11.5	12.125	4.125	

Using these weights, we have a method 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, we created a layered assurance method to calculate assurance in the layered system given all interdependency measurements between every set of layers.

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

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

While we were not able to measure the risk caused by a patient attacker with the attributes, we were able to estimate 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.

c. 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

$$TPV\ (l, D).$$

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

$$TPV(l, D) = 0.$$

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

$$r(BreakIn) = .05.$$

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

$$Is\_Vuln\ (l,D) = 1.$$

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

$$Is\_Vuln\ (l,D) = 0.$$

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

$$TPV\ (l, 0) = 0.$$

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

$$TPV\ (0,D)$$

is equal to the total potential vulnerability of the previous day($TPV\ (0, D\text{-}1)$) plus any known level vulnerabilities ($is\_Vuln\ (0, D)$) multiplied by the risk of a break in for level 0 or

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

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

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

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

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

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

Using the time-adjusted layered assurance value, we can calculate the entire Assurance of the Layered Solution (A_{LS}). 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).

$$A_{LS} = 1 - (1\text{-}AC_1Lt) * (1\text{-}AC_2Lt) * \ldots * (1\text{-}AC_{N_L}Lt)$$

d. Simulation Summary:

The overall goal of the simulation was to help answer the following questions:

1.Basic Question for layered solution

a. How do attributes affect layered interdependencies?

b. Which attributes are more critical to the layered solution?

c. When does the list of attributes analyzed get too large to analyze for the assessment of layered assurance?

d. Is it possible to measure the risk caused by a patient attacker with a set of attributes?

Questions 1a and 1b can be answered by looking at the risk weighting of each of the attributes. 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. The answer to 1c, 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 weighing 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.

Lastly, it is possible to measure the risk caused by a patient attacker with a set of given attributes, and this was modeled by the simulation. 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. Our specific example showed 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 by day 163 with a 100% chance of break-in on the second layer. Lastly, day 233 showed a 100% break-in on all 3 layers which rendered your Layered Solution useless.

5. Issues:

There were several issues encountered by this semester's team which were addressed in some logical way by the team, but should be viewed as an opportunity for improvement in the next project team.

The first issue was developing solid attribute risk weighting scores. When research did not yield any logical method of weighting the risks, Dr. Dark at Purdue recommended a Delphi study be performed which would solicit inputs from various NSA Tech Directors. However, due to time constraints, there was limited feedback. Another method may work better, or if given additional time the Delphi study could have yielded additional results.

A second issue was the mathematical calculations for the risk model. The team did utilize as much research as possible, but it may be a good idea to run all calculations by a mathematician to ensure all methods are used in the proper context.

A third issue was determining the distribution type to utilize in the simulation. The simulation actually used Monte Carlo methods which deploy continuous random distribution. However, due to time constraints, additional analysis should be performed to ensure that continuous random distribution is the right distribution to apply in this scenario.

Lastly, although the simulation did validate the Risk Model and showed how the Risk Model could work, actually deploying the risk model in a real environment could provide additional insights. It could also show what the model was lacking, and where additional changes could be made. It would also provide a more realistic picture of how the Risk Model works so that changes could be implemented.

6. Conclusion

Our work lends itself to discovering new ways to more accurately predict the assurance of a layered solution. Relating two layers to each other using our interdependence measurement methods 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, we 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, our 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.

7. Future Work

To improve our work, we require a better method for determining the weightings of attributes in our interdependence measurement. 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?

Using our work, an overall risk model for the network system should be more manageable. When a layered system is added to the overall risk model, the risk model can use our methods 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.

8. Bibliography:

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

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.

Chipchase, L., Schabrun, S., Cohen, L., Hodges, P., Ridding, M., Rothwell, J., ... & Ziemann, U. (2012). A checklist for assessing the methodological quality of studies using transcranial magnetic stimulation to study the motor system: an international consensus study. Clinical Neurophysiology, 123(9), 1698-1704.

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.

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

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 .

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

Mokkink, L. B., Terwee, C. B., Patrick, D. L., Alonso, J., Stratford, P. W., Knol, D. L., ... & de Vet, H. C. (2010). The COSMIN checklist for assessing the methodological quality of studies on measurement properties of health status measurement instruments: an international Delphi study. Quality of Life Research, 19(4), 539-549.

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

Ncube, C., & Dean, J. (2002). The Limitations of Current Decision-Making Techniques in the Procurement of COTS Software Components.

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

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.

Stolfo, S, Bellovin, M, Evans, D. (2011). Measuring Security: On the Horizon. IEEE (pp. 60-65).

Stonebumer, G., Goguen. A., Feringa, A. "Risk Management Guide for Information Technology Systems: Recommendations of the National Institute of Standards and Technology." Special Publication 800-30. July 2002.

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

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.

9. Biographical sketches

Marsella Farnam

Industrial and Systems Engineering graduate student. She completed several courses in engineering systems analysis as well as development of risk analysis and assessment models. Marsella's career requires that she understands requirements flow down from procurement specifications to acceptance requirements from both a hardware and software standpoint. Nearly 90% of the work that she performs is in a secure environment which has additional acceptance and certification requirements. Marsella has developed risk models that have resulted in control plans which were built to proactively mitigate potential noncompliance.

Robert Wells

Robert is currently an MA student in the Department of Political Science at Purdue University. Robert's academic interests are at the intersection of psychology and international relations. He is specifically interested in the public's perception about the proliferation of unmanned aerial vehicles (i.e. drones). His research investigates public attitudes about drones being used in a variety of sectors (e.g. agriculture, policing, and international warfare). He uses experiments and qualitative interviews to examine attitudes about US drone policy. Robert's co-written chapter titled: "Drone Use: The Future of Surveillance." In Large Scale Event Security Planning and Emergency Management. Edited by Eric Dietz and David Black Taylor & Francis, is forthcoming.

John Zage

John is an Information Security PhD Student at Purdue University. His research direction is the application of natural language processing to human risk assessments in relation to information systems. His previous work at the U.S. Army Research Laboratory on analyzing the risk in tactical edge networks required the creation of new risk models for tactical edge networks, and led to his current research direction.

10. Tasking

Spring 2015 Risk Analysis Team Task Delegation	Team Member		
Task	M. Farnam	R. Wells	J. Zage
Research Defense in Depth		X	
Research Risk Models			X
Research Simulation Models	X		
Create Risk Analysis Questions	X		X
Define Interdependent Attributes	X	X	X
Downselect Interdependent Attributes	X	X	X
Research Measurement Methods of Attributes	X	X	X
Prioritize Attributes	X	X	X
Refine Risk Model		X	X
Develop Simulation Model	X		X
Review results and make necessary changes	X	X	X
Develop and Submit Proposal Draft	X	X	X
Develop and pitch mid-term Progress Report Presentation	X	X	X
Write Mid-term Progress Report	X	X	X
Develop and Submit Final Proposal	X	X	X
Develop Project Poster	X	X	X
Develop and pitch final presentation	X	X	X
Develp and submit Final report	X	X	X