Engineering Management
Field Project

A Method for Estimating the Financial Impact of Cyber Information Security Breaches Utilizing the Common Vulnerability Scoring System and Annual Loss Expectancy

By

Michael B. Lindsey

Spring Semester, 2010

An EMGT Field Project report submitted to the Engineering Management Program
and the Faculty of the Graduate School of The University of Kansas
in partial fulfillment of the requirements for the degree of
Master's of Science

_____
Tom Bowlin
Committee Chairperson

_____
Ray Dick
Committee Member

_____
Herb Tuttle
Committee Member

_____
Travis Berkley
Committee Member

Date accepted:_____

Acknowledgements

I would like to thank Tom Bowlin, Ray Dick, Herb Tuttle, and Travis Berkley for their guidance and critique of this research project. I would also like to thank Dan LaMastres for his editing assistance and Amy Lindsey for her mathematical genius. Without their assistance this work would not have been possible.

To my family, I extend my greatest appreciation for the sacrifices that you have made in support of this academic journey. The time apart cannot be made up in kind. Huyen, Taylor, and Riley - your love, patience and understanding has been extraordinary and is one of the greatest gifts I have ever received.

Executive Summary

Information security is relatively new field that is experiencing rapid growth in terms of malicious attack frequency and the amount of capital that firms must spend on attack defense. This rise in security expenditures has prompted corporate leadership teams to scrutinize corporate security budgets. Information security risk, and the related financial impact, is not as easily calculated as other traditional sources of enterprise risk.

This research provides one method by which a firm may calculate the likelihood of a successful cyber security attack and the resulting financial impacts. The method incorporates annual loss expectancy and cost-benefit, which are tools familiar to most mid-level managers responsible for budget creation.

Table of Contents

List of Tables and Figures

1. Introduction

The practice of securing information is likely as old as humankind. Cyber information security, however, is a relatively young field that has risen in importance as businesses increasingly rely on information technology (IT) systems for operations.

Many midsize and large corporations employ people, technology, and processes to protect their information assets. For these firms, some of the most difficult questions to answer are:

- "What security metrics should be captured?"
- "What attacks are my systems vulnerable to?"
- "What is the likelihood that the company will experience a data breach?"
- "How much are the IT assets in the corporation worth?"
- "How much should the corporation spend to protect its IT assets?"

All of these questions can begin to be answered by utilizing risk assessment and risk management techniques. This research provides a method for calculating a corporation's financial exposure to cyber information security breaches using components of the common vulnerability scoring system (CVSS) in conjunction with asset values, annual loss expectancy (ALE), and cost to remediate.

Research regarding information security expenditures has largely focused on concepts such as return on investment (ROI), return on security investment (ROSI), and cost-benefit analysis, as well as, ALE and various forms of threat modeling. In the author's

own industry experience, corporations are actually significantly less sophisticated in their information security budgeting practices. The author contends that most information security expenditures are made with little quantitative data and are commonly based on the intuition of internal experts or the marketing influence of those who sell security products. What has been missing is a simple approach that mid-level managers, who are not information security experts, can use to understand the risk introduced in their corporation via security vulnerabilities. The approach should be simple in concept, use readily available tools, and reduce the uncertainty of risk to a level that is "just good enough".

2. Literature Review

The literature review for this research explores some of the common business drivers, or market forces, that create the need for information security. Common cyber security vulnerabilities and threats are also defined. After establishing the need for information security, conventional financial methods for the justification of security spend are analyzed. This research argues arriving at the frequency at which security vulnerabilities will be exploited is the most significant barrier to producing accurate estimates of loss expectancy, and this problem may be solved through a novel use of the CVSS.

2.1 Business Value of Information Systems and Data

Information systems comprise much of the internal infrastructure that supports modern businesses. Quite simply, the effective use of IT systems results in more efficient business processes which ultimately lower costs and increase profits. It follows that the physical machines and software that runs upon them have a value to the business that is greater than that retail price paid.

A 2006 study reports that 80% of firms maintain business continuity programs (Gale 1). Business continuity programs that include IT infrastructure are easily justified, as finance departments have access to historical data that assists them in calculating the likelihood that an earthquake, wildfire, tornado, or hurricane might occur. In addition, the mean time between failures for IT systems is often published and predicts with some accuracy the likelihood of machine failure. These data sets, in turn, help businesses plan their business continuity spending.

Data that is stored, processed, or transmitted by information systems also has value. In fact, businesses can mine data stored in IT systems and turn it into knowledge used to make business decisions. That is, data queries may be structured in such a way they can be used to answer questions, like "Where is the bottle neck in our distribution channel?" and "Why do our customers buy more widgets in the North Central Region on Monday mornings than on Tuesdays?" This data, especially when it contains privacy data, financial information, or trade secrets, is also valuable to malicious individuals and organized crime. Typically, a firm will want to protect the confidentiality, integrity, and availability of all critical systems and information.

2.2 Business Drivers: Standards and Regulations

The need to provide funding for resources to secure information increases as the value of the information increases to the business, or as industry and government create standards and regulations that mandate controls. These drivers create the requirement for security. In the case of industry and government mandates, where a business has little or no other option, funding for security expenditures is easily justified. The payment card industry data security standard (PCI-DSS) is one such industry initiative. A consortium formed by the credit card brands, the PCI-DSS is a prescriptive standard that dictates how merchants must handle card holder data. Some states have begun to codify this standard into law. As of January 1, 2010, all merchants in the state of Nevada must be compliant with the PCI-DSS standards (Wiener). Other examples of government regulation include the Health Insurance Portability and Accountability Act of 1996 (Congress) and California's privacy law SB1386 (Peace). However, when the only driver for information security expenditures is the value of the information to the corporation,

justification of the investment is usually required via a strong business case. Unfortunately, business cases for information security expenditures are not easily produced.

2.3 Business Drivers: Black Market Value

Aside from industry standards and regulations, the intrinsic value of data on the black market also creates a business driver to secure data and systems. The Verizon Business RISK team, formerly Cybertrust, produces a comprehensive annual data breach report using data collected from 150 forensic engagements involving the compromise of over 285 million records (Baker 8). It makes a strong case for the need to secure information security assets because it shows stolen data has real value on the black market, thus making IT systems a target worthy of attack. The report published in 2009 estimates credit card magnetic stripe data was worth about $0.50 a record in 2008 (Baker 8). This figure is down from approximately $10-$16 USD per record in 2007, but illustrates beautifully how the supply and demand of the black market is alive and functioning to drive illicit activities, just as it does in legitimate markets (Baker 8).

2.4 Security Vulnerabilities

Security vulnerabilities can result from various conditions. Breaches to physical security, such as when an unattended laptop is stolen, are one example. For the purposes of this research, physical security vulnerabilities are beyond scope. The three primary conditions that lead to a cyber security vulnerabilities referenced in this research are denial of service conditions, incorrectly configured system parameters, and a broad array of code related defects.

2.4.1 Denial of Service

A denial of service (DoS) condition is one that affects the availability of a system. A DoS is typically accomplished by causing an application fault by providing some input that the program was not expecting, and thus, does not understand how to process. More recently, large armies of personal computers are compromised to form a "botnet" (Lejeune 1). Once a malicious person has gathered enough "bots", he or she may instruct them to flood a target system with network traffic causing the target to be too busy to handle requests from legitimate users (Lejeune 1). This technique is known as a distributed denial of service (DDoS). The threat of DDoS is typically used by criminals to extort money, and is a vulnerability type that is not covered in this work.

2.4.2 System Mis-configuration

As defined by Sergy Chernov, "Change control is a process of submitting, authorizing, implementing, and reviewing changes to the company's computing environment software/hardware/networking) to minimize downtime and to keep all IT teams informed regarding proposed changes" (Chernov 9). For companies that lack mature IT change control and IT audit programs, improper system configuration is a risk for unwittingly introducing security vulnerabilities. Examples of system mis-configuration might include blank or default passwords, or incorrect file system permissions. These mistakes make system breaches all too easy for the attacker.

2.4.3 Code Defects

Code defects are another very broad, but common problem. Companies not in the business of making software and hardware typically rely on commercial off the shelf

(COTS) products written by established firms. Unfortunately, some of the largest software companies release products with security vulnerabilities to the market. Microsoft , for example, has implemented a security software development lifecycle process (Microsoft 1). In light of the effort to produce more secure software, Microsoft still "released a total of 74 bulletins in 2009 to address 189 vulnerabilities" (Prince 1). For the purposes of the research, code defects will be defined as any coding error that results in a security vulnerability. In reality, there exists an entire taxonomy for categorizing code defects that lead to a security vulnerability, but that minutiae does not need to be explored in this research.

2.5 Cyber Security Threats

Cyber security threats can broadly be categorized as originating from three sources: internal, external, and partners (Baker 9). Internal threat agents are those human resources that are employed at a firm. Internal threat agents typically have some level of elevated access, and in fact, accounted for 20% of the breaches in 2008 (Baker 9). Partners can be described as those firms with which a company engages in a business relationship (Baker 9). Partners oftentimes logically represent an extension of a corporation's network and bring with them a level of trust (Baker 9). Firms must contractually enforce that partners follow security practices that are as good as or better than those of a firm. Partners were implicated in 32% of the data breaches in 2008 (Baker 9). External threats are those originating from outside of a firm. There is no implied trust with the parties that comprise the threat agents in this group (Baker 9). Many are linked to organized crime, government entities, and hacking groups and were responsible for most of the documented breaches (74%) in 2008 (Baker 9). Note these

percentages sum to more than 100%, as breaches often involve actors from more than a single group (Baker 9).

2.6 Financial Calculations

Prior to making any capital or operational expenditures, firms will seek to realize a positive financial outcome. In the case of information security spending, this concept may be thought of as a reduction in loss. Common financial calculations used by firms in decision making include ROI, ROSI, and cost-benefit.

2.6.1 Return on Investment

One classic method for determining whether or not a business should provide resources to a project is ROI. The traditional simple calculation for ROI over one period of time is (Brotby 27):

ROI = (Net Income) / (Net Investment)

If a practitioner rearranges this equation to include net savings, it can begin to become useful to security managers (Brotby 27):

ROI = (Net Savings) / (Net Investment)

The ROI method of calculating the value of a security investment is simple; however, it assumes that one can accurately calculate what net savings are. That is, it does not factor in the concept of risk which makes this calculation of little value (Brotby 30). In

addition, it ignores the time value of money, assumes that the investment itself and its benefits will last for the depreciable life of the asset, and does not give weight to the timing of cash flows (Brotby 29). Other more robust methods of ROI calculation, such as net present value (NPV) and internal rate of return solve some of these issues.

2.6.2 Return on Security Investment

Another method of calculating the value of security investments is ROSI. ROSI is similar to ROI, but includes the concepts of risk exposure and risk mitigation. ROSI is calculated as (Sonnenreich, Albanese et al. 46):

ROSI = [(Risk Exposure) (Percent Risk Mitigated) – (Solution Cost)]/ (Solution Cost)

One problem with ROSI is risk exposure is a value is typically estimated by experts and is highly subjective when calculated by internal resources.

2.6.3 Cost-benefit

Cost-benefit analysis has also been proposed as a method for calculating security spend. Lawrence Gordon and Martin Loeb have written extensively about using cost-benefit analysis for security budgets. They suggest "firms should invest up to the point where the last dollar of information security investment yields a dollar of savings" (Gordon and Loeb 121). One method they propose employs NPV that provides a "risk-adjusted discount present value of expected benefits with expected costs" (Gordon and

Loeb 122). As is the case with ROSI, cost-benefit requires the opinion a security practitioner to estimate the probability and magnitude of losses.

2.7 Frequency

The most difficult task in calculating security spending is determining the likelihood a security vulnerability will actually be exploited. If one looks to the insurance industry for guidance, we see that there exists a field called actuary science whose purpose is to calculate financial risk when there is some level of uncertainty. Fortunately for actuaries, in many areas, there exists an extensive history of data upon which to rely. For example, an actuary can produce an accurate insurance premium for a 45 year old male smoker with high cholesterol, diabetes, and a family history of heart disease. The cache of historical data available to actuaries allows for the frequency of uncertain events, and thus financial risk, to be easily calculated. Unfortunately, for the information security industry, the historical data needed for these types of calculations does not exist. To compound the problem, technology moves at a very rapid pace. Thus, the threats, vulnerabilities, and exploits of today might be meaningless tomorrow.

2.7.1 Common Vulnerability Scoring System

As noted, one of the most difficult security values to calculate is the likelihood a vulnerability will actually be exploited. In many organizations, likelihood or frequency is estimated by an internal expert. The problem with this approach is individuals tend to produce varying opinions and could over-estimate or under-estimate likelihood. Additionally, it is very difficult for a single practitioner to be well versed in all technologies of a system. Web developers, database administrators, network engineers,

and operating system administrators spend entire careers honing their craft. The depth of knowledge required to analyze a particular vulnerability outside one's discipline, and its likelihood of being exploited, is very difficult for a single person to do. Furthermore, the cost of hiring a person or team of persons, with the knowledge to accurately achieve this is beyond what many firms can afford. A more reliable and easier method for calculating frequency can be derived from the CVSS.

The CVSS is a vendor-neutral vulnerability scoring system, and is maintained by the Forum of Incident Response and Security Teams (FIRST). CVSS scores are composed from three metric groups - Base, Temporal, and Environmental (Mell, Scarfone et al. 3). Each of these global metric groups contains a set of internal metrics as seen in Figure 1 (Mell, Scarfone et al. 5). When used as intended by the authors of the CVSS, the metrics produce a score that helps practitioners determine the risk for a vulnerability (Mell, Scarfone et al. 4). Scores generated on a scale of 0.0-10.0, are useful for the prioritization of mitigation strategies (Mell, Scarfone et al. 1). These scores do not, however, predict the likelihood or frequency that an exploit will occur.

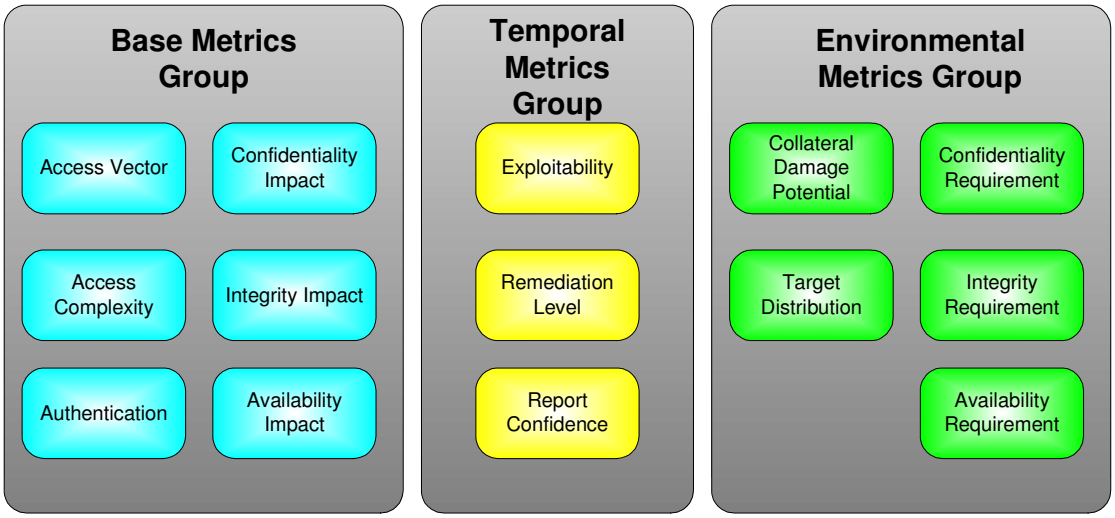

Figure 1: CVSS Metrics Groups (Mell, Scarfone et al. 5)

2.7.1.1 Base Metrics Group

The Base metric group seeks to represent the qualities of a particular vulnerability that are consistent across user environments (Mell, Scarfone et al. 6). It is further decomposed as having the following metrics (Mell, Scarfone et al. 7-9):

Access Vector (B_AV) – Describes the location from which a vulnerability can be exploited.

Access Complexity (B_AC) – Describes the complexity of the attack once an attacker has gained access to the system

Authentication (B_Au) – Describes how many times an attacker must authenticate to exploit a system.

Confidentiality Impact (B_C) – Metric to describe if exploitation leads to unauthorized access to data.

Integrity Impact (B_I) – Metric to describe if an attacker has the ability to modify data stored on system in such a manner that its integrity has been diminished and the data can no longer be trusted as genuine.

Availability Impact (B_A) – Metric to describe if successful exploitation leads to degraded system performance, or even complete shutdown.

2.7.1.2 Temporal Metrics Group

The Temporal metric group represents qualities of a vulnerability that are subject to change over time (Mell, Scarfone et al. 10). The sub-components of the Temporal metrics group include (Mell, Scarfone et al. 10-11):

Exploitability (T_E) – Describes whether or not the tools exist to exploit a vulnerability. Often, exploitation is theoretical.

Remediation Level (T_RL) – Metric to describe the currently available mitigation options.

Report Confidence (T_RC) – Metric to describe the degree of confidence in the reported vulnerability. Often, vulnerabilities are announced sans technical details.

2.7.1.3 Environmental Metrics Group

The Environmental metrics group captures data specific to a firm's infrastructure (Mell, Scarfone et al. 11). Since this data will not be used directly in this research, it will not be described in detail. More information can be found in the FIRST publication "*A Complete Guide to the Common Vulnerability Scoring System Version 2.0*" (Mell, Scarfone et al.).

2.7.2 Obtaining Frequency from CVSS

Researchers Houmb and Franqueria describe a method for using CVSS sub-components to calculate both frequency and impact in their research "*Estimating ToE Risk Level using CVSS*" (Houmb and Franqueira). This method is further described in "*Quantifying Security Risk Level from CVSS Estimates of Frequency and Impact*"

(Houmb, Franqueira et al.). In their method, both frequency and impact are utilized to generate risk level through the use of a Bayesian belief network (BBN) (Houmb, Franqueira et al. 7). A BBN is a tool that can be used to model relationships between uncertain events that have influence upon one another. The researchers complete model is illustrated in Figure 2.

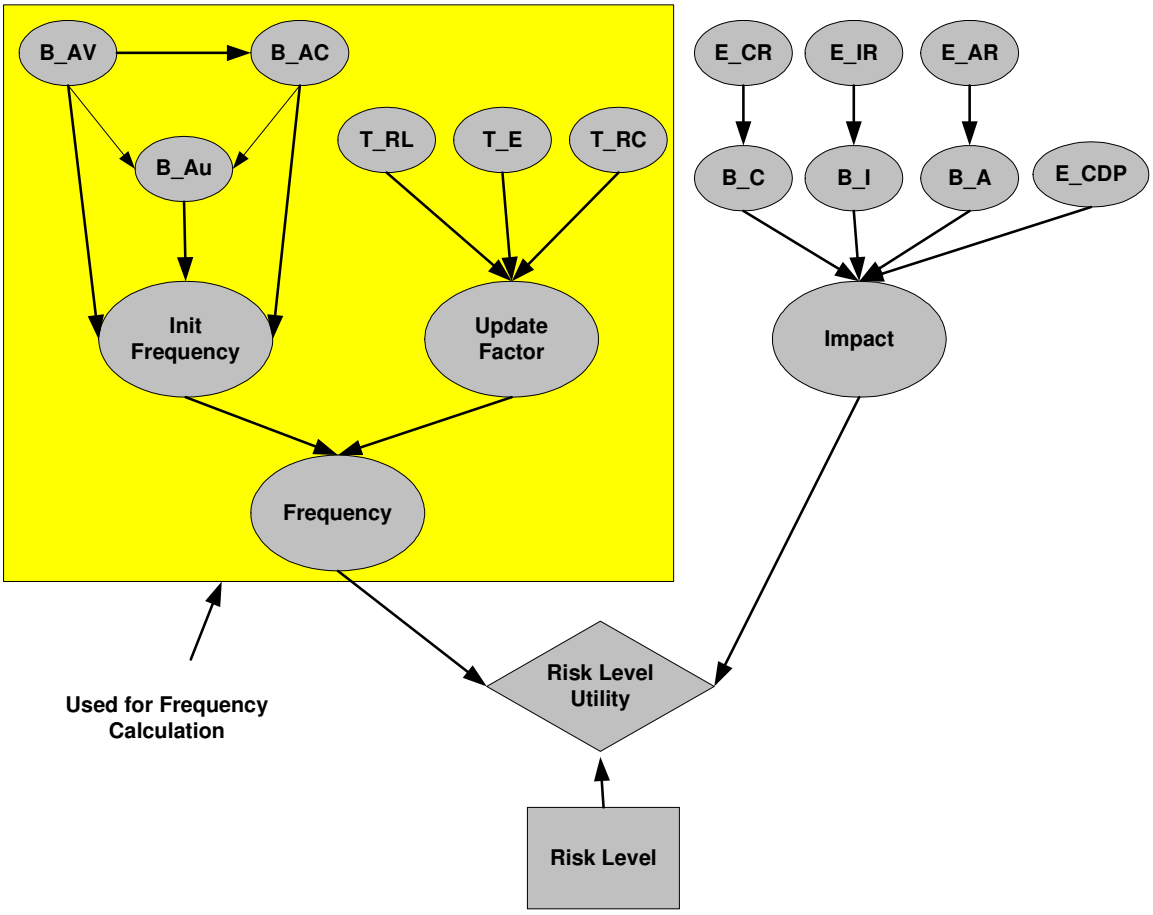

Figure 2: Houmb BBN (Houmb, Franqueira et al. 6)

Houmb, et al. use the sub-components of the CVSS as input to their BBN to arrive at calculations for frequency, impact, and risk level (Houmb, Franqueira et al. 7). This approach is elegant for three reasons:

1. The data is presumed to be accurate. Most vulnerability CVSS sub-scores are authored by the makers of the technology product in question or professional vulnerability bulletin analysts (Mell, Scarfone et al. 5). This point is important because scores are created by experts most familiar with the technology in question, and thus, their ratings are presumed to be properly calibrated.

2. Almost all known vulnerabilities are catalogued by the National Vulnerability Database published by the National Institute of Standards and Technology and sponsored by the United States Department of Homeland Security's National Cyber Security Division (NIST 1). The result is that most known vulnerabilities and their CVSS scores are readily available.

3. Most commercial and open-source vulnerability scanners include CVSS scores as a reporting feature.

The idea behind the structure of the components of frequency calculation in the BBN is that the easier a vulnerability is to exploit, the more likely it is that it will be exploited (Houmb, Franqueira et al. 7). This idea assumes hackers will target more vulnerable systems before expending the extra effort to exploit more difficult or unreliable vulnerabilities. For input, the following attributes from the Base metrics group

are used: Access Vector (B_AV), Access complexity (B_AC), and Authentication (B_Au) (Houmb, Franqueira et al. 6). These elements are used to derive the misuse initial frequency (MF_{init}) (Houmb, Franqueira et al. 6). Three attributes from the Temporal group are also included (Houmb, Franqueira et al. 6). They are Remediation Level (T_RL), Exploitability (T_R), and Report Confidence (T_RC) (Houmb, Franqueira et al. 6). These elements are used to derive the misuse update factor (MF_{uFac}). Ultimately, both MF_{init} and MF_{uFac} are combined to produce a final value representing misuse frequency (MF) as seen in Figure 3 (Houmb, Franqueira et al. 6). The portion of the BBN responsible for calculating the impact of a vulnerability will not be used in the method described in this research.

There are three equations used to calculate MF in this model. The first makes use of the Base metrics variables to calculate the MF_{init}, a second uses Temporal attributes to generate MF_{uFac}, and final calculation produces MF (Houmb and Franqueira 721).

$$MF_{init} = \int_{N\_1} P(B\_AV, B\_AC, B\_Au)$$

$$MF_{uFac} = \int_{N\_1} P(T\_E, T\_RL, T\_RC)$$

$$MF = \int_{N\_1} (MF_{init} \times MF_{uFac})$$

Values for these equations are sourced from the numerical weights assigned from the CVSS sub-scores as depicted in Table 1 (Houmb and Franqueira 721).

Table 1: CVSS Metrics Values (Houmb and Franqueira 721)

CVSS Metrics Group	CVSS Attribute	Rating	Rating Value
Base Metrics	Access Vector (B_AV)	Local (L) Adjacent Network (A) Network N)	0.395 0.646 1.0
	Attack Complexity (B_AC)	High (H) Medium (M) Low (L)	0.35 0.61 0.71
	Authentication Instances (B_Au)	Multiple (M) Single (S) None (N)	0.45 0.56 0.704
Temporal Metrics	Exploitability Tools & Techniques (T_E)	Unproven (U) Proof-of-Concept Functional (F) High (H)	0.85 0.9 0.95 1.0
	Remediation Level (T_RL)	Official Fix (OF) Temporary Fix (TF) Workaround (W) Unavailable (U)	0.87 0.90 0.95 1.0
	Report Confidence (T_RC)	Unconfirmed (UC) Uncorroborated (UR) Confirmed (C)	0.90 0.95 1.0

2.7.3 Other Methods of Frequency Calculation

The calculation of frequency in relation to the exploitation of security vulnerabilities has garnered the attention of many consultants and academics. Two of the areas explored by other researchers for frequency calculation include game theory and Monte Carlo simulations.

2.7.3.1 Game Theory

Other methods of determining risk where uncertainty exists have been proposed by researchers. Game theory is one such method. Game theory attempts to model the actions of two competitive actors. In the case of Information Security, these might be described as the "firm" and the "malicious threat agent". In the publication, "*Risk Assessment of Malicious Attacks Against Power Systems*", researchers have created a model using game theory which produces a value for risk (Bompard, Ciwei et al. 1). Game theory models of this type require a strong understanding of graduate level mathematics. It is the belief of the author that this model, due to complexity, would not be practical to implement by the average mid-level manager responsible for creating an information security budget.

2.7.3.2 Monte Carlo Simulation

Monte Carlo simulation is another method by which it is possible to produce a value for risk in the presence of uncertainty. One simplistic approach used to calculate frequency via Monte Carlo simulation is demonstrated in research authored by James Conrad (Conrad 1). Conrad notes that, "A Monte-Carlo simulation enables an analyst to quantify the uncertainty in an expert's estimate by defining it as a probability distribution rather than just a single expected value." (Conrad 1). While Conrad's approach is valid, it does rely on experts to provide estimates as input to the model. Thus, a firm would be required to employ analysts with a level of security knowledge commensurate with desired accuracy of estimations.

2.8 Summary

Data stored in IT systems has both value to the businesses that rely upon it, as well as to black market actors. The intrinsic value of this data in conjunction with mandatory government controls and contractual obligations create the requirement to secure it. Given that security vulnerabilities are routinely discovered in IT systems, it follows that businesses must evaluate the financial costs associated with mitigating them. Paramount to the success of creating an accurate business case for these expenditures is the estimation of frequency that a successful exploitation will occur. By utilizing publicly available CVSS data, mid-level managers can estimate frequency and successfully show the business benefit of security vulnerability mitigation.

3. Research Procedure

This report illustrates how mid-level managers can use simple tools to calculate cost avoidance through security spend. The components for this calculation include system inventory, value of systems to the business, vulnerability scan data, and simple ALE and cost-benefit calculations. The high-level steps involved in this research procedure are:

- Definition of scope, to include system inventory
- Execution of a security vulnerability assessment
- Calculation of exploitation frequency based on CVSS
- Definition of potential sources of financial impact
- Calculation of ALE, SLE, and cost-benefit

3.1 Scope Definition

The first step followed in this research procedure was to properly scope the systems under review. All systems under review were inventoried and logically grouped into one or more "system codes". The "System code" concept was utilized to show relationships between devices or components across a distributed architecture.

3.2 Vulnerability Assessment

Following the collection of inventory, a vulnerability assessment was performed. A security vulnerability assessment is the process of looking for flaws in a system that would allow a malicious user to obtain access to data or cause the system to move to a non-functioning state.

3.3 Determine Frequency

The next step presented in this research involves calculating the frequency of vulnerability exploitation. The method of determining this frequency value makes use of a CVSS values extracted from vulnerability scan data.

3.4 Determine Sources of Financial Impact

After a vulnerability assessment was performed, the financial impact of downtime for the systems under review was determined. Financial impact data was collected from internal experts with knowledge of the various sources that could result in financial losses.

3.5 Calculate Financial Impact

Lastly, there are three basic calculations utilized in this research to arrive at the figures to support financial decisions. The financial calculations performed were SLE, ALE, and cost-benefit.

4. Results

The research procedure outlined in this work was executed on production IT infrastructure and all mathematical calculations were modeled within a spreadsheet program.

4.1 Scope Definition / Inventory

Determining system scope is best illustrated with an example. Consider a typical eCommerce web site, which is likely to include all of the following:

- Multiple servers, running one or more operating systems (may be virtual)
- COTS web software (for presentation and business logic)
- COTS database
- Network infrastructure, to include: routers, switches, load balancers, firewalls
- Custom applications that are written in-house

A diagram of a typical eCommerce environment is illustrated in Figure 3. Each of the sub-components of the system may be subject to one or more security vulnerabilities, and successful exploitation of any could cause downtime for the entire site.

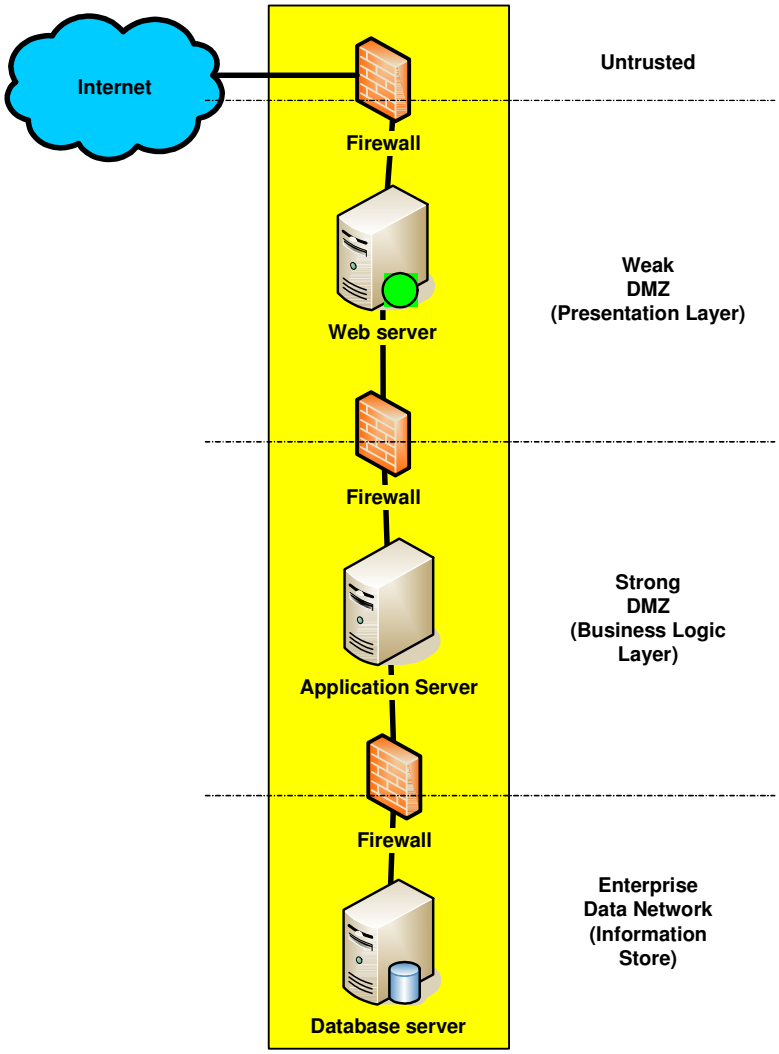

Figure 3: Typical eCommerce Infrastructure

In this research, a network vulnerability scan was performed on components typical of an eCommerce infrastructure using a commercially available security scanner. The architecture under review consisted of a web-based application with presentation, business logic, and database layers. In addition, the architecture included a networking infrastructure to support packet flow and provide security.

Each system in the architecture was categorized as belonging to a group, or "system code". This grouping is beneficial for showing the physical system relationships

in applications that have a distributed architecture. In this example, there is a single application (ECOM) supported by two infrastructure "system codes" (NET & SEC). In a true production environment, there would very likely be multiple applications supported by the NET and SEC "system codes". It follows that successful attacks on systems on infrastructure (NET and SEC) could affect all applications that rely upon them. However, the existence vulnerabilities on a second application (say ECOM2) may or may not result in collateral damage to ECOM. It is the responsibility of the business to understand the intricacies of these "system code" relationships. Table 2 shows the relationship between "system code" and physical device and Figure 4 depicts the entire environment.

Table 2: System Codes and Hostnames

System Code	Hostname	IP Address
NET	Router _A	10.0.0.1
NET	Switch_A	192.168.5.11
NET	Switch_B	192.168.5.12
NET	Switch_C	192.168.5.13
ECOM	Web_Server_A	10.0.1.3
ECOM	App_Server_B	192.168.3.10
ECOM	Database_A	192.168.5.10
SEC	Firewall_A	10.0.1.2
SEC	Firewall_B	192.168.2.1
SEC	Firewall_C	192.168.4.1

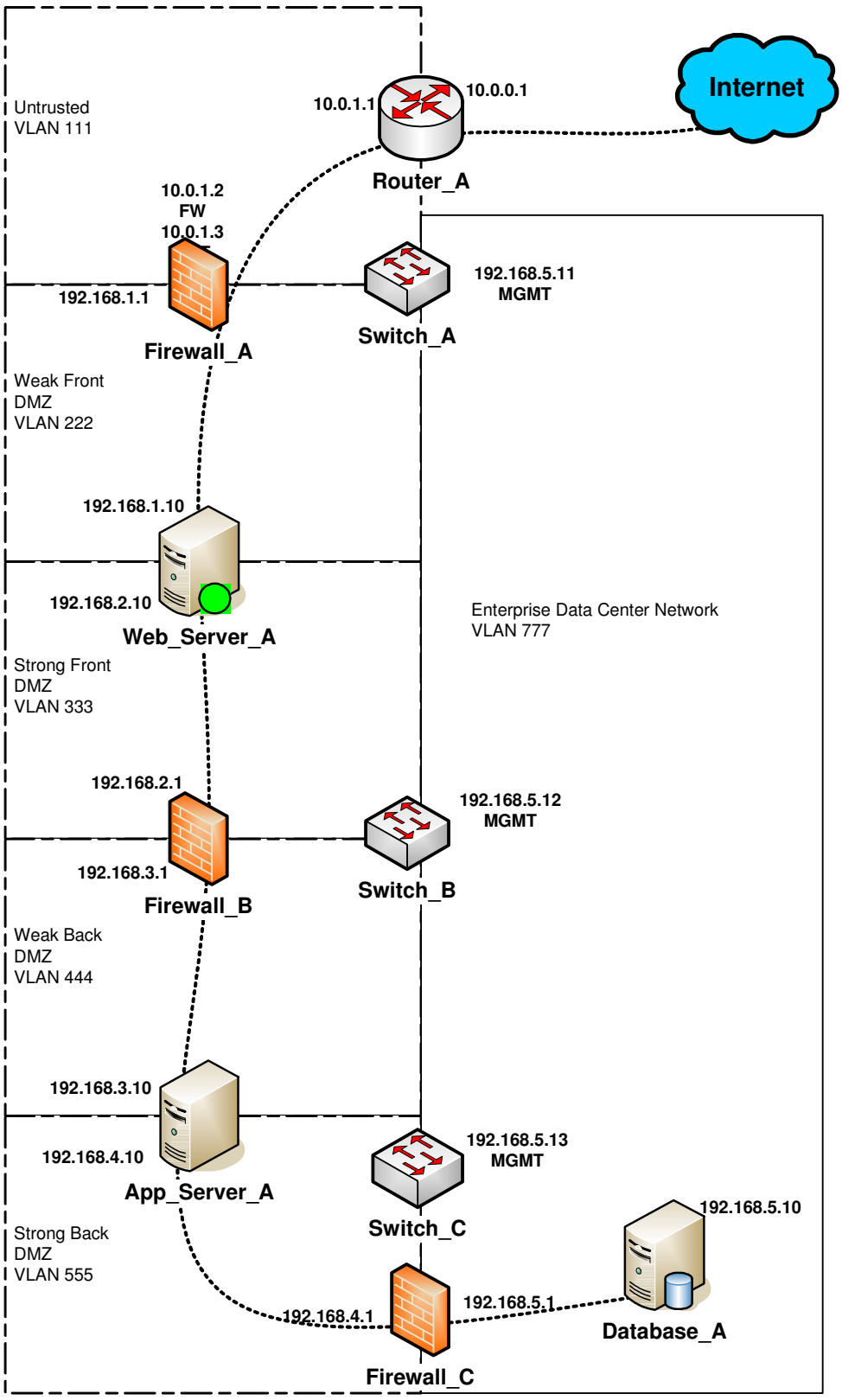

Figure 4: Test Environment

4.2 Vulnerability Assessment

There are many techniques used in a vulnerability assessment exercise, ranging from interviews with system owners to the use of automated tools, however, vulnerability testing methodology typically falls into one of two categories. "Black box" testing is a methodology where a tester has no knowledge of the system prior to the engagement. Conversely, in "white box" testing a tester will have full knowledge and access to all system components. The "white box" testing methodology usually includes access to source code, as well. In practice, most firms use a modified "black box" approach. For example, a practitioner will have some knowledge of the environment they are testing (data flow, architectural diagrams, test user accounts, etc). For the vulnerability assessment performed in this research, the complete network design was known.

There are many free and open-source tools available on the market that are reasonably accurate at identifying known security vulnerabilities. Network vulnerability scanners are applications that work by first "fingerprinting" devices, a method of remotely identifying device type, followed by the execution of platform specific rules that probe a device for the existence of documented vulnerabilities. The vulnerability scan data in this research was collected with a commercial product.

For this exercise, the nCircle IP360 network vulnerability scanner was chosen. This commercial product was selected for its ability to provide all necessary vulnerability data and CVSS values in a text-based and tokenized comma separated values (CSV) format. The CSV format is beneficial for this research due to the ease in which standard UNIX text manipulation tools can be used ('sed', 'grep', and 'awk'). The commercial version of the Nessus security scanner was also considered, but lacked a simple method

for extracting CVSS scores. A raw version of the nCircle IP360 vulnerability output, with some extraneous fields removed, can be seen in Figure 5.

```
DNS Name,IP Address,Vulnerability Name,Vulnerability Advisories
Router_A,10.0.0.1,Telnet Available,nCircle CVSS Base Vector:
(AV:N/AC:L/Au:N/C:N/I:N/A:N)
```

Figure 5: Raw Vulnerability Data

The results of the network based vulnerability scan unveiled 405 vulnerabilities in the test environment. The raw report files were parsed with UNIX utilities to output a format suitable for working within a spreadsheet program. Microsoft Office Excel 2002 was used for this example, but other spreadsheet programs should perform equally well. The data used to populate a tab titled "Inventory_Vuln" is illustrated in Figure 6.

				CVSS Base Metric Scores			CVSS Temporal Metrics Score			
System Code	Hostname	IP Address	Vulnerability Name	Access Vector (B_AV)	Attack Complexity (B_AC)	Authentication Instances (B_Au)	Exploitability Tools & Techniques (T_E)	Remediation Level (T_RL)	Report Confidence (T_RC)	Confidentiality (B_C)
NET	Router_A	10.0.0.1	Cisco IOS and Cisco Un	N	M	N	F	OF	C	P
NET	Router_A	10.0.0.1	Cisco IOS and Cisco Un	N	M	N	F	OF	C	P
NET	Router_A	10.0.0.1	Cisco IOS BGP Transitiv	N	L	N	U	OF	C	N
NET	Router_A	10.0.0.1	Cisco IOS Border Gatew	N	M	N	F	OF	C	N
NET	Router_A	10.0.0.1	Cisco IOS Border Gatew	N	L	N	U	OF	C	N
NET	Router_A	10.0.0.1	Cisco IOS Crafted IP Op	N	L	N	U	OF	C	C

Figure 6: Vulnerability Inventory

4.3 Frequency Calculations Based on CVSS Score

The method for calculating the ARO is inspired by the work in "*Estimating ToE Risk Level Using CVSS*" (Houmb and Franqueira 7). The ARO, a percentage, is calculated by determining the average of all the $M_{FreqInit}$ values multiplied by the average of all the MF_{uFac} in a system. The $M_{FreqInit}$ and MF_{uFac} for a single vulnerability are

simply calculated by using the using the CVSS values in Table 1 to calculate a percentage. Substituting ARO for MF produces the following equation:

$$ARO = \int_{N\_1} \left(MF_{Init} \times MF_{uFac} \right)$$

A second spreadsheet worksheet, title "Constants", was created to house constant data used in these calculations. The table of constants appears in Figure 7.

CVSS Base Metric Scores		
Access Vector (B_AV)	**Abbreviation**	**Value**
Local (L)	L	0.395
Adjacent Network (A)	A	0.646
Network (N)	N	1.000
Attack Complexity (B_AC)		
High (H)	H	0.350
Medium (M)	M	0.610
Low (L)	L	0.710
Authentication Instances (B_Au)		
Multiple (M)	M	0.450
Single (S)	S	0.560
None (N)	N	0.704
Confidentiality Impact (B_C)		
Complete (C)	C	

CVSS Temporal Metrics Score		
Exploitabiltiy Tools & Techniques (T_E)		**Value**
Unpoven (U)	U	0.850
Proof-of-Concept (PoC)	POC	0.900
Functional (F)	F	0.950
High (H)	H	1.000
Remediation Level (T_RL)		
Official Fix (OF)	OF	0.870
Temporary Fix (TF)	TF	0.900
Workaround (W)	W	0.950
Unavailable (U)	U	1.000
Report Confidence (T_RC)		
Unconfirmed (UC)	UC	0.900
Uncorroborated (UR)	UR	0.950
Confirmed (C)	C	1.000

Figure 7: CVSS Constants

With this data, it is possible to perform searches within the spreadsheet program to look up the values necessary to calculate MF_{init}, MF_{uFac}, and MF for each individual security vulnerability discovered (based on Houmb's method). The resulting spreadsheet columns are depicted in Figure 8. These columns and rows exist in the same worksheet as the complete vulnerability data. At the core, these calculations are simply averages.

CVSS Base Metric Scores				CVSS Temporal Metrics Score					
Access Vector (B_AV)	Attack Complexity (B_AC)	Authentication Instances (B_Au)	Confidentiality (B_C)	Exploitability Tools & Techniques (T_E)	Remediation Level (T_RL)	Report Confidence (T_RC)	Initial Frequency	Frequency Update Factor	Frequency
N	M	N	P	F	OF	C	77.13%	94.00%	85.57%
N	M	N	P	F	OF	C	77.13%	94.00%	85.57%
N	L	N	N	U	OF	C	80.47%	90.67%	85.57%
N	M	N	N	F	OF	C	77.13%	94.00%	85.57%
N	L	N	N	U	OF	C	80.47%	90.67%	85.57%

Figure 8: Frequency Values

4.4 Determining Sources of Financial Impact

Many firms cannot accurately state how much revenue is lost during periods of downtime or how issues affecting one sub-component of a system translate to issues for another. Firms with mature IT organizations, however, will have an outage management function that can produce data depicting the cost of system downtime. Total financial impact to a firm might include the sum of areas, such as:

- Fines/fees resulting from breaking contract service level agreements (SLA)
- Revenue lost from the inability to sell a product or service
- Damage to a firm's brand or reputation
- Fines from regulatory bodies
- Lost employee productivity

The sources of financial impact will vary from firm to firm, and each category must be defined uniquely for the business. In addition to defining financial impact sources, the method in this research utilizes low, medium, and high to arrive at an average daily financial loss. The inclusion of these tiers is completely arbitrary. The tiers simply provide a level of granularity to the business analyst. The security vulnerabilities responsible for low, medium, and high events are not directly tied to the financial impact of the event. In other words, these figures could represent the financial impact resulting from a power failure or a fire – just as easily as the exploitation of a vulnerability that leads to the need for complete restoration a system from a known good backup. This concept is important for two reasons:

1. It is likely any outage management department will have data representing financial impacts to systems resulting in general downtime.
2. As mentioned previously, this approach seeks to produce data that is "just good enough" given other data that a firm already possesses. No effort is made to produce actual relationships between the result of exploiting a particularly vulnerability and the length at which it causes downtime. All successful exploits are assumed to cause financial impact - be that a denial of service, loss of data, or the cost of rebuilding a system.

4.5 Calculating SLE, ALE, and Cost-benefit

SLE in this method is derived by determining the start stop times of an expected outage. This duration should be based on previous a practitioner's knowledge of similar outages. For example, most firms will have knowledge of the length of time that it takes to repair a system to a known good state following a catastrophic system failure. The

total expected outage duration is then multiplied by the average cost per minute of an outage for that specific system. The cost per minute is simply an average of all the potential sources of financial loss for a particular system (loss of ability to sell product/service, regulatory fines, etc). Finally, if there is risk of data loss, the number of records in a system is multiplied by the cost of losing a single record.

$$SLE = DurationInMin \times CostPerMin + (CostPer\,\mathrm{Re}cord \times Number\,\mathrm{Re}cords)$$

$$DurationInMin = OutStopTime - OutStartTime$$

$$CostPerMin = \left(\sum^{n} LossPerDay \times 1/n \right) \div 1440^{*}$$

* *1440 minutes per day*

To support SLE calculations, the spreadsheet used for this model contains a third worksheet tab, titled "Fin_Impact" to hold the financial impact resulting from various sources described in section 4.4. These sources would change depending on a firm's line of business, but for this model have been defined as:

- SLA by Contract
- Sales Revenue Lost/Incident
- Brand/Reputation Damage
- Regulatory Fines
- Productivity Lost

Each of these categories has a low, medium, and high value for financial impact. The inclusion of these levels is strictly for the convenience of the practitioner, and may or may not be available in all organizations. All the values are ultimately averaged and used

to calculate the total financial impact of system down time per "system code" per minute. A portion of the financial impact table is depicted in Figure 9.

System Name	System Code	Average Financial Impact			Service Level Agreements by Contract (1 Day)				Sales Revenue Lost/Incident (1 Day)			
		Day	Hour	Minute	Low	Medium	High	Average	Low	Medium	High	Average
eCommerce	ECOM	3,833.33	159.72	2.66	500.00	1,000.00	10,000.00	3,833.33	0.00	0.00	0.00	0.00
Network	NET	20,000.00	833.33	13.89	0.00	0.00	0.00	0.00	10,000.00	0.00	50,000.00	20,000.00
Security	SEC	1,000.00	41.67	0.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1,000.00
		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 9: Financial Impact Values

4.5.1 Report Dashboard: ALE

To arrive at the ALE for a "system code", which can represent one or more physical devices or applications, this method uses the widely accepted equation of single loss expectancy (SLE) multiplied by the Annual Rate of Occurrence (ARO).

$$ALE = SLE \times ARO$$

The report dashboard produces the ALE for the sum of all vulnerabilities in a particular "system code". There are two components used to calculate the direct financial impact. First, it is assumed that all successfully exploited vulnerabilities will result in system downtime. Therefore, the model calculates minutes of down time from "Impact Start Time" and "Impact Stop Time". This data is multiplied by the total "Financial Impact/Minute" and the total ARO (average of all vulnerabilities in a "system code").

Next, the CVSS Base sub-score for confidentially (B_C) is used to determine if there is a risk for data loss. Security industry analysts typically rely on an annual report published by the Ponemon Institute to determine the cost of losing customer privacy information. The most current report estimates this value at $202 USD (Ponemon 2). If known, the number of privacy records that could be compromised in a system can be entered. The end result is that both system downtime, as well as, data loss is represented in ALE.

System Code	Impact Start Time	Impact Stop Time	Impact Duration	Financial Impact/Minute	Risk of Data Loss?	Number of Records at Risk	Single Loss Expectancy (SLE)	Annual Rate of Occurance (ARO)	Annual Loss Expectancy (ALE)	Cost Benefit
ECOM	12/1/09 3:00	12/1/09 7:00	240	$2.66	True	200	$41,038.89	77.10%	$31,642.04	$0.73
NET	1/1/10 0:00	1/3/10 0:00	2880	$13.89	True	8,888	$1,835,376.00	78.18%	$1,434,870.00	$173.71
SEC	1/1/10 0:00	1/2/10 0:00	1440	$0.69	------		$1,000.00	75.32%	$753.24	$0.15

Figure 10: Dashboard

4.5.2 Cost-benefit

The vulnerabilities classes described in section 4.2 are largely mitigated simply by applying vendor supplied security patches or by altering some system configuration that increases the security posture of a system. Given that the IT labor for these mitigation activities is known by most firms, cost-benefit analysis over a single period is easily calculated as:

$$Benefit = ALE - CostToMitigate$$

Cost-benefit in this model is produced simply by summing the labor estimates required to mitigate all the vulnerabilities in a particular "system code" and subtracting that value from the ALE. For the purposes of demonstration, the cost of remediation for a single vulnerability is calculated at a labor rate of $70 USD. It is estimated that the

mitigation of each vulnerability requires two hours of labor. This block of time includes not only the time necessary to perform the technical work, but also takes into account the time a technical resource would spend completing change management documentation and scheduling maintenance windows with stakeholders in the business for a planned outage.

4.6 Abbreviated Example

The following abbreviated example calculates the ALE and cost-benefit for a single vulnerability on a single host utilizing the primary equations described in this research. The vulnerability under review has the properties listed in Table 3.

Table 3: Vulnerability Properties

Vulnerability Name	Apache Tomcat Directory Traversal Vulnerability	
Common Vulnerability and Exposure ID	CVE-2007-0450	
CVSS Base Vectors		**Score**
Access Vector (B_AV)	Network (N)	1.000
Attack Complexity (B_AC)	Low (L)	0.710
Authentication Instances (B_Au)	None (N)	0.704
Confidentiality (B_C)	Complete (C)	N/A
CVSS Temporal Vectors		
Exploitability Tools & Techniques (T_E)	Functional (F)	0.950
Remediation Level (T_RL)	Official Fix (OF)	0.870
Report Confidence (T_RC)	Confirmed (C)	1.000

To determine the frequency for the vulnerability described in Table 3, the following calculation is used:

$$MF_{init} = (1.00 + 0.71 + 0.704) \div 3 = 0.8047$$
$$MF_{uFac} = (0.95 + 0.87 + 1.00) \div 3 = 0.94$$
$$MF = (0.8047 + 0.94) \div 2 = 0.8723$$
$$MF = ARO = 0.8723$$

The CVSS base vector for confidentiality indicates that this vulnerability would allow for the complete loss of confidentiality of data. For this example, let us assume the data at risk is customer information that could be used by criminals for identity theft and that no other impact to the system occurs. Furthermore, let us assume that there are 100,000 customer records in the data store and that each record lost causes $202 USD impact to the firm. To determine the SLE, the following calculation is preformed:

$$SLE = 202 \times 100{,}000 = 202{,}00{,}000$$

The ALE can be derived as:

$$ALE = 20{,}200{,}000 \times 0.8723 = 17{,}620{,}460$$

If the labor rate for remediation of this vulnerability is $70 USD per hour, and 2 hours of labor are required, the cost-benefit is calculated as:

$$Benefit = 17{,}620{,}460 - 140 = 17{,}620{,}320$$

In this example, the firm could spend $140 USD to prevent a potential loss of $17,620,460 USD.

4.7 Summary

This research provides a simple method by which mid-level managers may produce the necessary financial calculations to support a business case for the remediation of security vulnerabilities. At the core of the method is a novel use of data extracted from CVSS sub-scores to estimate the frequency of security vulnerability exploitation. It is this calibrated estimation of frequency that provides the foundation for all other financial calculations. Ultimately, the total financial impact of the exploitation of security vulnerabilities in a collection of "system codes" may be viewed in a simple dashboard style report.

5. Suggestions for Additional Work

The model presented in this research is likely more sophisticated than what is currently used by most firms for the calculation of security expenditures. However, the simplicity of the approach also presents many opportunities for improvement. Some areas that could be further developed include:

1. Logic could be introduced into the model to programmatically calculate the expected time of system downtime of a single vulnerability. This calculation would rely on expert knowledge internal to the organization.
2. The potential for collateral damage to systems in adjacent "system codes" could be automatically calculated based on "system code" relationships. In the current iteration of the model, "system code" relationships must be known and collateral damage must manually estimated by the practitioner.
3. The calculation for MF in this model is based solely on the weights provided by the CVSS sub-scores in Table 1. A more robust approach might combine these weights with actual threat data from intrusion detection or intrusion prevention systems. Thus, the MF value could be increased or decreased in concert with the current threat landscape.

References / Bibliography

Baker, W. (2009). 2009 Data Breach Investigations Report, Verizon Business**:** 50.

Bompard, E., G. Ciwei, et al. (2009). "Risk Assessment of Malicious Attacks Against Power Systems." Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on **39**(5): 1074-1085.

Brotby, W. K. (2009). Information security management metrics : a definitive guide to effective security monitoring and measurement. Boca Raton, Auerbach Publications.

Chernov, S. (2008). "IT checkup: change control." Potentials, IEEE **27**(5): 9-10.

Congress, U. S. (19996). HEALTH INSURANCE PORTABILITY AND ACCOUNTABILITY ACT OF 1996. t. Congress, U.S. Department of Health & Human Services. **Public Law 104-191:** Department

Conrad, J. R. (2005). Analyzing the Risks of Information Security Investments with Monte-Carlo Simulations. IEEE Workshop on the Economies of Information Security, Boston.

Gale (2009). Business Continuity and Disaster Recovery Planning. Encyclopedia of Emerging Industries. . Farmington Hills, Gale Group.

Gordon, L. A. and M. P. Loeb (2006). "Budgeting process for information security expenditures." Association for Computing Machinery. Communications of the ACM **49**(1): 121.

Houmb, S. H. and V. N. L. Franqueira (2009). Estimating ToE Risk Level Using CVSS. Availability, Reliability and Security, 2009. ARES '09. International Conference on.

Houmb, S. H., V. N. L. Franqueira, et al. (2009). "Quantifying security risk level from CVSS estimates of frequency and impact." Journal of Systems and Software **In Press, Corrected Proof**.

Lejeune, M. A. (2002). "Awareness of Distributed Denial of Service Attacks' Dangers: Role of Internet Pricing Mechanisms." Netnomics **4**(2): 145-162.

Mell, P., K. Scarfone, et al. (2007). "A Complete Guide to the Common Vulnerability Scoring System Version 2.0." Retrieved 06/20, 2009, from http://www.first.org/cvss/cvss-guide.html.

Microsoft (2009). "Microsoft Security Development Lifecycle (SDL)." Retrieved 12/09/09, 2009, from http://msdn.microsoft.com/en-us/security/sdl.aspx.

NIST (2007). "NVD Common Vulnerability Scoring System Support v2." Retrieved 06/20, 2009, from http://nvd.nist.gov/cvss.cfm.

Peace, S. (2002). Senate Bill No. 1386. 1386. S. o. California. California.

Ponemon, L. (2009). "Fourth Annual US Cost of Data Breach Study: Benchmark Study of Companies." Retrieved 02/28/2010, 2010, from http://www.ponemon.org/local/upload/fckjail/generalcontent/18/file/2008-2009%20US%20Cost%20of%20Data%20Breach%20Report%20Final.pdf.

Prince, B. (2009). "Microsoft Fixes Critical IE Security Vulnerabilities on Final Patch Tuesday for 2009." eWeek. Retrieved 12/09/09, 2009, from http://www.eweek.com/c/a/Security/Microsoft-Fixes-Critical-IE-Security-Vulnerabilities-in-2009s-Final-Patch-Tuesday-896233/.

Sonnenreich, W., J. Albanese, et al. (2006). "Return On Security Investment (ROSI) - A Practical Quantitative Modell." Journal of Research and Practice in Information Technology **38**(1).

Wiener, V. (2009). Senate Bill No. 227. 227. S. o. Nevada. Nevada**:** 4.