I hereby submit two (2) copies of my thesis, **"Quantitative Risk Assessment of Vulnerabilities and Threats"** for inclusion into the AUM Library. One copy will be bound and placed in the circulating collection; the remaining copy will be retained as a preservation copy. I hereby give the library permission to store, preserve, and make accessible a digital copy of my theses within the context of an institutional repository. I further give permission for the library to catalog and to make available to researchers the images of my theses, without restriction. I also give permission to the Library to make copies of this thesis for preservation purposes.

Sharmila A

Sharmila Ashokan

May | 14 | 2015

Date

Phill Johnson/ Dean of the AUM Library

6-5-15

Date

QUANTITATIVE RISK ASSESSMENT OF VULNERABILITIES AND THREATS

Sharmila Ashokan

Certificate of Approval:

Mehmet Sahinoglu PhD, Chair
Director of Informatics Institute, Office of the Provost

Luis Cueva-Parra, PhD
Associate Professor, Department of Mathematics and Computer Science

Robert Underwood, PhD
Professor, Department of Mathematics and Computer Science

Matthew Ragland, PhD
Associate Provost for Graduate Studies and Faculty Services

Joe King, PhD
Provost

# Quantitative Risk Assessment of Vulnerabilities and Threats

by

Sharmila Ashokan

A thesis submitted to the Graduate Faculty of
Auburn University at Montgomery
in partial fulfillment of the
requirements for the Degree of
Master of Science

Montgomery, Alabama
May 16, 2015

Keywords: Risk Assessment, Vulnerability, Threat, Counter Measure, Risk management



Approved by

Mehmet Sahinoglu PhD, Chair
Director of Informatics Institute, Office of the Provost
Luis Cueva-Parra, PhD
Associate Professor, Department of Mathematics and Computer Science
Robert Underwood, PhD
Professor, Department of Mathematics and Computer Science
Matthew Ragland, PhD
Associate Provost for Graduate Studies and Faculty Services
Joe King, PhD
Provost

## Abstract

Information security risk assessment has become an essential component of an organizations' operations. Vulnerabilities and threats pose many challenges to the security of any system. Without vulnerability and threat management process in place, organizations are blind to those risks related to the security of their Information Technology (IT) infrastructure. Implementing a vulnerability and threat management process is all about managing risk. By having a well defined process in place, an organization can obtain a continuous view of the risk associated with the presence of security weakening vulnerabilities in its IT systems. This allows organizations to take well advised decisions with regards to remediating actions that could be implemented to mitigate the risks within a cost effective roadmap.

Since the potential risks from different cyber-attacks are increasing, the damage due to lack of cyber security is growing and becoming a serious economic concern to many organizations. Any organization that desire to obtain an understanding of the security risks that they are facing due to the technology, which they are using; should implement a vulnerability and threat management process. This thesis presents a quantitative risk assessment of different security measures followed up by mitigating the unfavorable risk percentage to a tolerable minimal value through game-theoretic optimization using linear programming. Economic metrics are applied for the efficiency assessment and comparative analysis of different protection scenarios.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Dr. Mehmet Sahinoglu, for his excellent guidance, patience, and providing me with an excellent academic atmosphere for accomplishing this truly challenging work. He has been a true mentor and advisor in every sense of the term. I could not have imagined having a better advisor and mentor for my Masters Thesis studies.

I would also like to thank Dr. Luis Cueva-Parra and Dr. Robert Underwood for serving as members of my thesis advisory committee, for their consistent encouragement and insightful comments. I would like to thank Dr. Enoch Lee, Head of the Department of Mathematics and Computer Science, for his departmental support and encouragement. I would also like to thank Dr. Mathew Ragland, Associate Provost for the financial support throughout my Masters studies.

I also owe many thanks to Mr. David. J. Tyson, for his constant guidance and support in coding related challenges. In addition to it, I would also like to extend my thanks to Mr. Joel Junker and Mr. Anthony Buenger who served as my resourceful instructors at the CSIS (Cyber Systems and Information Security) graduate program. I would equally like to thank Mr. Robert Barclay, CSIS Cybersecurity laboratory assistant for discussing with me practical security issues. In addition to all these, I would like to thank all the faculty and staff from the Department of Mathematics and Computer Science, and many others whom I have inadvertently left out during the writing of this thesis.

Last but certainly not the least I would like to express my deepest appreciation to my beloved parents for their relentless support and love during my Master of Science studies.

Table of Contents

List of Figures

# Chapter 1

# Introduction

According to SANS(deriving from SysAdmin, Audit, Networking, and Security) Information Security Resources, "the Information Security refers to the processes and methodologies which are designed and implemented to protect printed, electronic or any other form of confidential, private and sensitive information or data from unauthorized access, use, misuse, disclosure, destruction, modification, or disruption" [1]. As Internet is easily available and accessible, it is increasingly becoming a hunting ground for criminals, activists and terrorists motivated to steal money, get noticed, compromise trust and cause disruption or even bring down corporations and governments through online attacks.

Cyber crime has become a major problem with so many new forms of virus, malware, security breaches and also with many different hacking techniques. With the increasing threat of cyber security breaches and information leaks, the importance of Information Security in organizations is growing rapidly, where organizations are investing significant resources to prevent and safeguard their system from attacks. Organizations are looking for actionable information and substantial counter measures which would help them to prevent cyber attacks.

The security risk assessment model which we have developed is based on a quantitative analysis of the security risks and enables organizations to introduce optimum security solutions. In analyzing the security risks, the model quantitatively evaluates the information assets and their vulnerabilities and threats. The model is designed following structured steps, from the initial selection of input data to the final recommendations selecting the appropriate solutions. The values of the risk parameters are used as a basis for choosing the appropriate counter measures that reduce security risks.

## 1.1 Risk

Risk is defined as "the potential harm that may arise from some current process or from some future event". IT security risk is defined as "the harm to a process or the related information resulting from some purposeful or accidental event that negatively impacts the process or related information" [2]. The management of Information security essentially comes down to mitigating that risk wisely with a planned purpose. The purpose of this thesis is to build a model by performing vulnerability and threat analysis towards risk assessment and risk mitigation using game-theoretic optimization.

### 1.1.1 Risk Management

According to Information Systems Audit and Control Association (ISACA), " Information risk management defines the areas of an organizations information infrastructure and identifies what information to protect and the degree of protection needed to align with the organizations tolerance for risk. It identifies the business value, business impact, compliance requirements and overall alignment to the organizations business strategy" [3].

If the stakes are high enough, we can and should deal with risk explicitly, with the aid of a quantitative model. As humans, we use "heuristics" or "rules of thumb" for dealing with risk, but these don't serve us very well in many business and public policy situations and are frequently deceptive. In fact, much research shows that we have cognitive biases, such as over-weighing or exaggerating the most recent adverse event and projecting current good or bad outcomes too far into the future, all of which work against our desire to make the best decisions. Quantitative risk analysis can help us escape these biases, and make better decisions such as we practice using thermostats in our dwellings rather than sufficing with mild-severe-low temperature installments.

It helps to recognize up front that when uncertainty is a major factor, the best decision does not always lead to the best outcome. However, risk analysis can help us analyze, document, and communicate to senior decision makers and stakeholders the extent of uncertainty, the limits of our knowledge, and the reasons for taking a course of action to remediate.

### 1.1.2 Risk Assessment

Once risks have been identified, they must then be assessed as to their potential severity of impact and to the probability of their likelihood of occurrence. These quantities can be either simple to measure, in the case of the value of a lost building, or impossible to know for sure in the case of the probability of an unlikely event occurring. In order to properly prioritize the implementation of the risk management plan, it is very important to make the best decisions in the risk assessment process [4]. Risk Assessment allows organizations to determine the level of security controls required and allows us to demonstrably justify the decisions we have taken. To determine the level of risk assessment required, it is important to evaluate the security requirements of the unit, legal and regulatory requirements and the nature and criticality of the asset needing protection.

### 1.1.3 Quantitative and Qualitative Risk Analysis

Risk analysis is the basis of information protection, risk assessment and risk management in the process of information protection. Risk analysis includes process such as identification of activity, threat analysis and vulnerability analysis. This method is usually called matrix-based approach. There are two fundamental types of risk analysis, they are Quantitative risk analysis and Qualitative risk analysis.

Qualitative risk analysis does not involve numerical probabilities or predictions of loss. They are usually represented by non-numerical label such as "High", "Medium", "Low". Qualitative risk analysis involve numerical probabilities of various adverse events, and also determines the extent of loses if a particular event occur. Quantitative approach creates a very precise analytical interpretation that can clearly represent which risk-resolving measures have been most well-suited. This makes the quantitative approach favored by many organizations since risk assessments can be clearly represented in the empirical forms like percentages or cost.

## 1.2 Terminology

### 1.2.1 Assets

Assets are the resources that generate and keep information. All information assets should have a clearly defined owner. The process of identifying and valuing assets should include both owners of the asset and operational managers. When placing a value on information assets it should be done assuming that no controls are currently in place and should consider, for example, loss of information, loss of availability, disclosure of information, destruction of information and interference with communications. We can broadly classify assets in the following categories:

**1. Information assets**

Every piece of information about an organization falls in this category. This information has been collected, classified, organized and stored in various forms.

Databases: Information about customers, personnel, production, sales, marketing, finances. This information is critical for businesses. Its confidentiality, integrity and availability is of utmost importance.

Data files: Transactional data giving up-to-date information about each event.

Operational and support procedures: These have been developed over the years and provide detailed instructions on how to perform various activities.

Archived information: Old information that may be required to be maintained by law or convenience.

**2. Software assets**

These can be divided into two categories: Application software: Application software implements business rules of the organization by allowing end users to perform coordinated functions, tasks and actions. Creation of application software is a time consuming task. Integrity of application software is very important. Any flaw in the application software could impact the business adversely.

System software: An organization would invest in various packaged software programs like operating systems, development tools and utilities etc [5]

.

**3. Physical assets**

These are the visible and tangible equipment and could comprise of:

a) Computer equipment: Servers, desktop, mainframe and notebook computers.

b) Communication equipment: Modems, routers and fax machines.

c) Storage media: Magnetic tapes, disks and CDs.

d) Technical equipment: Power supplies, air conditioners.

e) Furniture and fixtures [5]

.

**4. Services**

a) Computing services that the organization has outsourced.

b) Communication services like voice communication, data communication, value added services, wide area network etc.

c) Environmental conditioning services like heating, lighting, air conditioning and power [5].

### 1.2.2 Vulnerability

Vulnerability is defined as "a weakness of an asset or group of assets that can be exploited by one or more threats" [6]. Threats go hand in hand with vulnerabilities and can be graded in a similar manner, measured in terms of motivation and capability. Management is better able to understand the implications of the threat and vulnerabilities when they are quantifiable and measurable.

Examples of vulnerabilities include

World-writeable password files (modification of system-critical data).

Default password (remote command execution or other access).

Denial of service problems that allow an attacker to cause a Blue Screen of Death.

Smurf (denial of service by flooding a network) [7].

### 1.2.3 Threat

Threat is defined as "a circumstance or event with the potential to adversely impact an asset through unauthorized access, destruction, disclosure, modification of data, and/or denial of service" [8]. Threats that are identified must be considered in relation to the business environment and what affect they will have on the organization.

Once the assets have been identified, the threats projected toward those assets should be established. Threats are essentially all things that have the potential to exploit a weakness to result in some form of temporary or permanent damage. Threats can be environmental, deliberate, accidental, logical or technical and should be identified and classified according to their potential impact. When evaluating threats, their extent and likely frequency should be measured. Other factors that may be considered include:

1. The motivation behind the threat
2. The opportunity for the threat to be realized
3. The capability and resources of attackers
4. The attractiveness of the target.

Examples of threats include:

i) Physical threats: natural disasters, such as earthquakes, tsunami, flood, fire etc.

ii) Logical threats: bugs in hardware and power failures.

iii) Human threats: non-malicious and malicious threats, such as disgruntled employees and hackers.

### 1.2.4 Counter Measure (CM)

In Computer Security, a countermeasure is "an action, device, procedure, or technique that reduces a threat, a vulnerability, or an attack by eliminating or preventing it, by minimizing the harm it can cause, or by discovering and reporting it so that corrective action can be taken" [13]. Each threat has a CM value that ranges between 0 and 1 and whose complement gives Lack of Counter Measure (LCM). That is $LCM = 1 - CM$.

### 1.2.5 Residual Risk (RR)

Residual risk is defined as "the portion of risk that remains after countermeasures are applied "[15]. If countermeasures are applied properly in the organization there should be no RR.

### 1.2.6 Non Residual Risk (NRR)

Non Residual Risk is "the risk that an activity would pose if no controls or other mitigating factors were in place (the gross risk or risk before controls)" [14].

### 1.2.7 Capital Cost (CC)

Capital (investment) cost is "the total expected loss in monetary units (e.g. dollars or euros) for the particular system if it is completely destroyed and can no longer be utilized, excluding the shadow costs, had the system continued to generate added value for the system" [15].

## 1.3 National Vulnerability Database (NVD)

"NVD is the U.S. government repository of standards-based vulnerability management data presented using the Security Content Automation Protocol (SCAP). This data enables automation of vulnerability management, security measurement, and compliance. NVD includes databases of security checklists, security related software flaws, misconfigurations, product names, and impact metrics" [9]. In particular, NVD supports the Common Vulnerability Scoring System (CVSS) version 2 standards for all Common Vulnerabilities and Exposures (CVE) vulnerabilities . NVD provides CVSS base scores which represent the innate vulnerability characteristics. It does not currently provide 'temporal scores' (scores that change over time due to events external to the vulnerability).

However, NVD does provide a CVSS score calculator to allow you to add temporal data and to even calculate environmental scores (scores customized to reflect the impact of the vulnerability on your organization). This calculator contains support for U.S. government agencies to customize vulnerability impact scores based on Federal Information Processing Standards (FIPS) 199 System ratings. See Figure 1.1 and 1.2. [15]

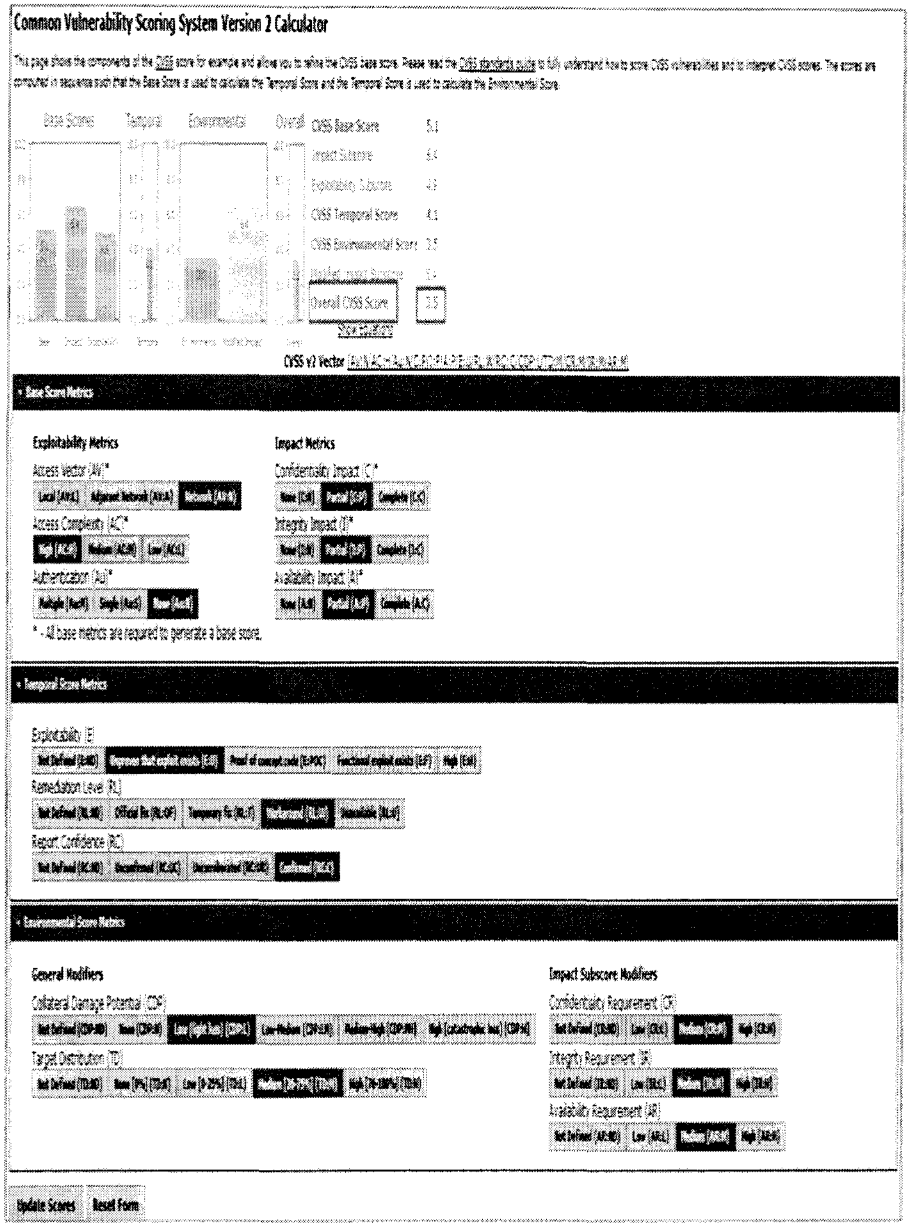



Figure 1.1: CVSS Calculator example screenshot entered by the author.

## 1.4 Common Vulnerabilities and Exposures (CVE)

"The CVE system provides a reference-method for publicly known information-security vulnerabilities and exposures.

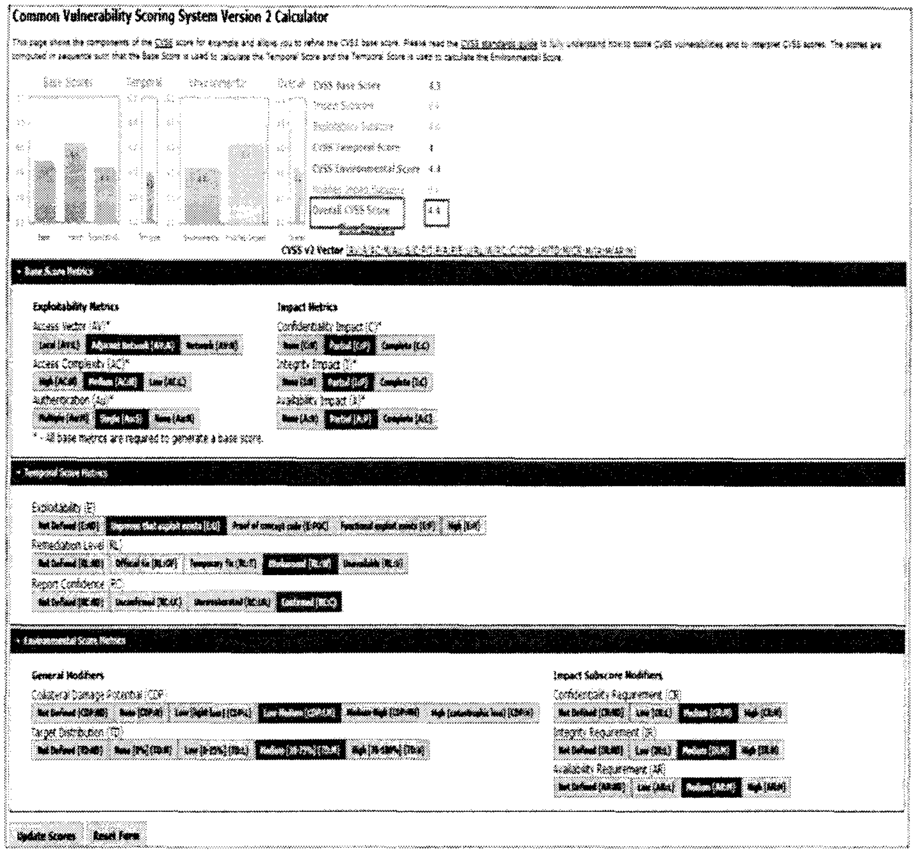



Figure 1.2: NVD Calculator example screenshot entered by the author.

MITRE Corporation maintains the system, with funding from the National Cyber Security Division of the United States Department of Homeland Security. CVE is used by the Security Content Automation Protocol, and CVE IDs are listed on MITRE's system as well as the US National Vulnerability Database" [10].

CVE provides an easy to use web interface to CVE vulnerability data. We can browse for vendors, products and versions and view CVE entries, vulnerabilities, related to them. We can view statistics about vendors, products and versions of products as shown below in Figure 1.3.

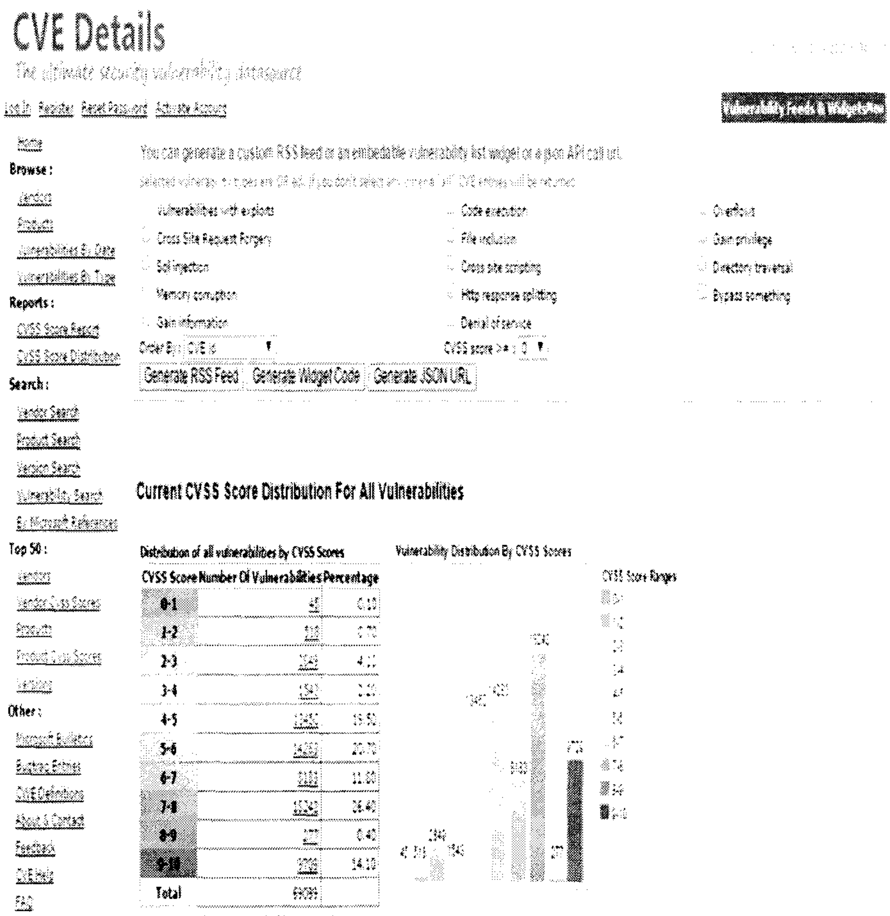



Figure 1.3: CVE Products and Statistics Screenshot

Also CVE details are displayed in a single, easy-to-use page as illustrated below in Figure 1.4.

CVE vulnerability data are taken from National Vulnerability Database (NVD) xml feeds provided by National Institute of Standards and Technology (NIST). Additional data from several sources like exploits, vendor statements and additional vendor supplied dat, Metasploit modules are also published in addition to NVD CVE data. Vulnerabilities are classified by cvedetails.com using keyword matching and CVE numbers if possible, but they are mostly based on keywords. Unless otherwise

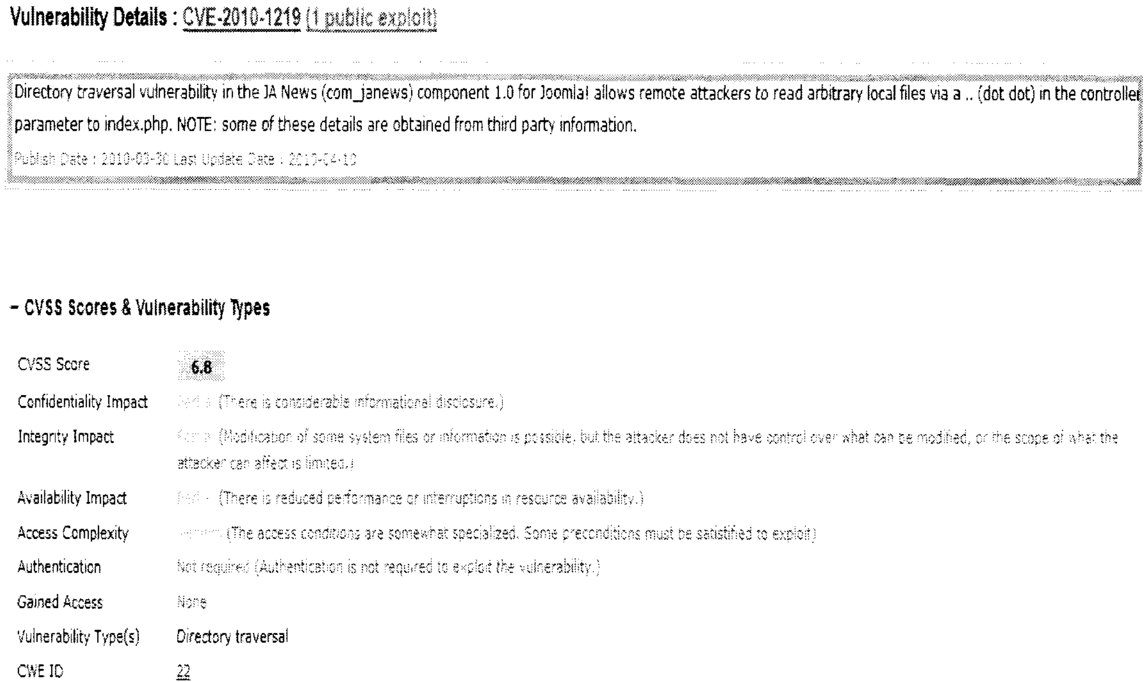



Figure 1.4: CVE details example screenshot

stated CVSS scores listed on this site are "CVSS Base Scores" provided in NVD feeds. Vulnerability data are updated daily using NVD feeds [11].

## 1.5 Purpose

The purpose of this Security Risk Assessment approach is to provide the organizations with a more approachable view of risk assessment regarding the system. The data present in CVE website provides the details of the vulnerability and threats. By doing analysis and segregating threat and vulnerability data, we can predict the probability of occurrence of each threat and vulnerability. We can also use this data to analyze the most occurring vulnerability and threats in a system. Using these data, we are able to successfully calculate the optimized counter measure and residual risk in a system which would help us to take preventive action against any system exploitation in an organization.

## 1.6 System Description

The risk assessment system has been determined to be a Security assessment application which provides users with an option to perform threat and vulnerability analysis. The periodic assessment of risk to organizations operations or assets resulting from the operation of an information system is an important activity required by Federal Information Security Management Act (FISMA). The application is used for risk assessment in accordance with NIST and Risk Management Guide for Information Technology Systems vulnerability and threats. By doing analysis and segregating threat and vulnerability data, we can predict the probability of occurrence of each threat and vulnerability. We can also use this data to analyze the most occurring vulnerability and threats in a system. Using these data, we are able to successfully calculate the optimized counter measure and residual risk in a system which would help us to take preventive action against any system exploitation in an organization.

## 1.7 Scope

To build a practical and accurate quantitative model, we will initially collect the data from different sources and then the models risk analysis probabilities will be estimated using the equations that were developed. The system also provides a method for performing threat and vulnerability analysis. The level of risk assessment required will depend on the security requirements of the unit, legal and regulatory requirements and the nature and criticality of the asset needing protection.

## 1.8 Hardware and Software Requirements for the Application

Based on the business need and the potential usage, the MySQL Database Management System will be utilized. The basic hardware and software requirements for the thesis is as shown in below Table 1.1.

| **Component** | **Requirement** |
|---|---|
| Computer and processor | 500-megahertz (MHz) processor or higher |
| Memory | 256 megabytes (MB) of RAM or higher |
| Hard Disk | 2 gigabyte (GB) available disk space |
| Display | 1024 768 or higher resolution monitor |
| Operating System | Windows XP with Service Pack (SP) 3 (32-bit), Windows Vista with SP1, Windows Server 2003 R2 with MSXML 6.0, Windows Server 2008 or later (32-bit or 64-bit), Windows 7 or later operating systems. |
| Software or higher resolution monitor | Microsoft Web Matrix will be used as a tool for developing the front end. PHP, HTML, CSS and Java script will be used for front end design of webpage forms and reports. |
| Database | MySQL database using MySQL Workbench |
| Other | Connectivity with Windows Server 2003 with SP1 or later running Windows SharePoint Services is required for certain advanced collaboration functionality. Use of graphics hardware acceleration requires DirectX 9.0c compatible graphics card with drivers dated 11/1/2004 or later.Internet Explorer 6 or later, 32-bit or 64-bit browser. Internet functionality requires Internet access (fees might apply). |

Table 1.1: Hardware and Software Requirements.

# Chapter 2

# Risk Analysis

Risk analysis is of two types: Quantitative risk analysis and Qualitative risk analysis. In this thesis, we have used Quantitative risk analysis method.

## 2.1 Quantitative Risk Analysis Method

IT risk is most often represented in terms of expected losses. In Quantitative risk analysis method we will evaluate the losses in numerical terms. The losses may include repair costs to information systems or the replacement cost for an asset that is stolen or lost. It assists user in determining the cost-benefit analysis associated with risks. For some resources or assets in an organization it would be difficult to calculate the losses precisely. In order to calculate the intangible losses there should be proper realization of business process, frequency of threat occurrence and probability of incident occurrence causing loss of asset value in a definite period. Probability of a security incident occurrence is defined as the number of times that a particular threat can occur during a period of time. Probability of the incident can be calculated as the product of probability of threat (T) and asset vulnerability (V) [12].

Threat probability is defined as "a probability of an attack on information assets" [12]. It is equal to the number of attacks per unit time. System vulnerability V is defined as "a probability of a threat that is successfully realized in a form of an incident on an informational asset" [12]. If there is any security incident in the organization, there will be a financial expected cost of loss (ECL) incurred in the organization which will be measured in monetary units. Although it is difficult to measure the financial losses accurately, the immediate direct loss due to an incident can be measured easily. The losses can range from losses of productivity, losses of revenue, and increased costs. Indirect losses from a security incident are very difficult to be measured and represent damage to the organization, business processes, legal liabilities, loss of property or reputational losses [12].

The quantitative analysis of risk can be measured through allocation of losses to individual factors. The security risk represents the expected financial loss caused by the security incident measured in the same monetary unit.

## 2.2 How we Measure or Estimate Risk?

Data for malicious attacks that have been prevented or not prevented are collected. The probabilistic inputs are vulnerability, threat, and LCM of all risks whose value range between 0 and 1, and the constants are the capital cost (asset) and criticality constant (between 0 and 1). The residual risk and expected cost of loss are the outputs obtained using equations (2.1) to (2.3) below. The black box in Figure 2.1 leads to the probabilistic tree diagram of Figure 2.2 to do the calculations. In Figure 2.2, V1 and V2 are vulnerabilities, where as T1 and T2 are threats for respective vulnerabilities.

LCM11 is the lack of counter measure for vulnerability V1 and threat T1, LCM12 is the lack of counter measure for vulnerability V1 and threat T2, LCM21 is the lack of counter measure for vulnerability V2 and threat T1, LCM22 is the lack of counter measure for vulnerability V2 and threat T2. Similarly CM11 is the counter measure for vulnerability V1 and threat T1, CM12 is the counter measure for vulnerability V1 and threat T2, CM21 is the counter measure for vulnerability V2 and threat T1, CM22 is the counter measure for vulnerability V2 and threat T2.Equations (2.1)(2.3) summarize Figures 2.1 and 2.2 from input to output. If there is an attack recorded, we need to come up with a percentage of non-attacks and successful attacks. Out of many such attempts, the number of successful attacks will yield the estimate for

the percentage of LCM. We can then trace the root of the cause to the threat level backward from the outcomes in the tree diagram. Let us imagine that the anti-virus (AV) software did not catch it, and a virus attack occurs, which reveals the threat exactly. As a result of this attack, whose root threat is known, the e-mail system may be disabled. Then, the vulnerability comes from the e-mail itself. This way, we have completed the line of attack on the tree diagram, as illustrated in Figure 2.2. The following equation 2.1 computes the RR for each activity in Figure 2.2 for each leg [15]:
newline

$$RR_{i,j} = P(V_i) \times P(T_j|V_i) \times LCM_{i,j} \tag{2.1}$$

Covering all legs in a tree diagram, RRs ($0 < RR < 1$) summed total to Total Residual Risks(TRR), ($0 < TRR < 1$) as shown in Figure 2.2.

$$FR = TRR \times k \tag{2.2}$$

$$ECL = FR \times CC \tag{2.3}$$

Where $P(V_i)$ is the probability of vulnerability $V_i$, $P(T_j|V_i)$ is the probability of threat $T_j$ corresponding to vulnerability $V_i$, $k$ is the criticality constant whose value is between 0 and 1 and FR is the Final Risk. TRR is calculated using equation 2.4.

$$TRR = \sum RR_{i,j} \tag{2.4}$$

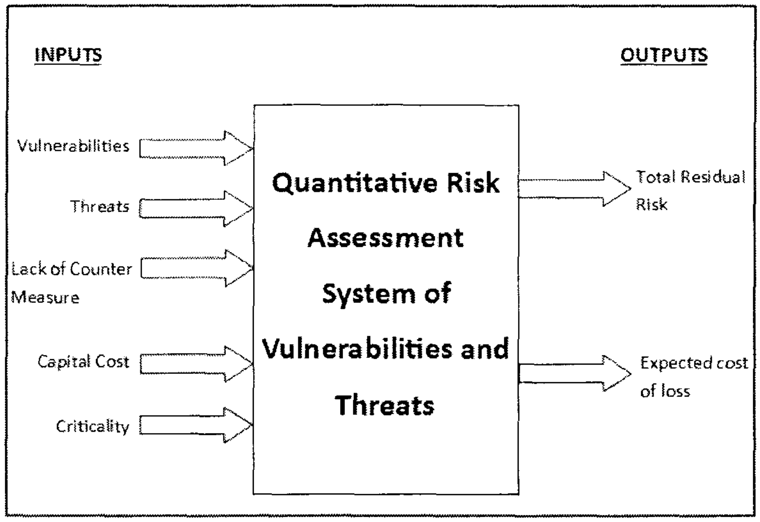



Figure 2.1: Quantitative SM model of probabilistic and deterministic inputs and outputs

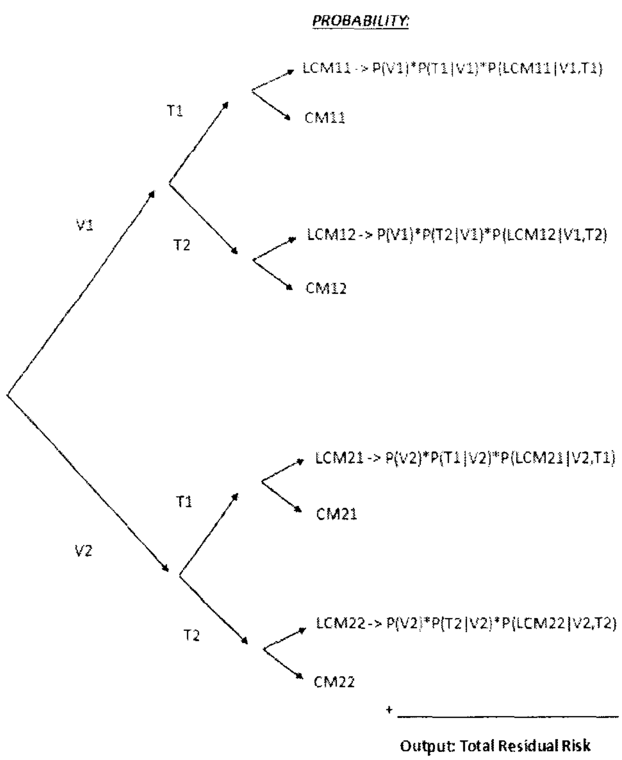



Figure 2.2: Simple tree diagram for two threats per each of the two vulnerabilities

# Chapter 3

# Game Theory and Linear Programming

This chapter reviews the topic of game theoretic optimization using linear programming, and its methods and applications towards the purpose of quantitative risk assessment.

## 3.1 Game Theory

Game Theory is "a decision-making situation in which two or more decision makers compete by each selecting one of several strategies". The value of the game can be provided to decision makers by combining the competing strategies [33].

The usage of game theoretic risk computing is steadily increasing in the world of cyber-risk informatics. Cyber systems security which did not exist when game theory debuted has recently evolved into a complex and challenging problem. The area of cyber network defense mechanism design has been receiving immense attention from the research community for more than two decades ever since the first internet message was delivered. However, the cyber security problem is far from completely solved. Scientists are exploring the applicability of game-theoretic approaches to address the security issues and some of these approaches look promising [17].

Game theory, therefore, is a branch of applied mathematics that attempts to analytically model the rational behavior of intelligent agents in strategic situations, in which an individual's success depends on the decisions of others. While initially developed to analyze competitions in which one individual does better at another's expense, it recently evolved into techniques for modeling a wide class of interactions, characterized by multiple criteria [17].

## 3.2 Linear Programming

Linear programming (also called linear optimization) is "a method to achieve the best outcome (such as maximum profit or lowest cost) in a mathematical model whose requirements are represented by linear relationships. Linear programming is a special case of mathematical programming (mathematical optimization)" [18]. Linear programming problems are optimization problems where the objective function and constraints are all linear. The objective of all linear programming problems is the maximization or minimization of some quantity.

Linear programming is a most significant sector of optimization. In operations research, many practical problems can be expressed as linear programming problems. Historically, " ideas from linear programming have inspired many of the central concepts of optimization theory, such as duality, decomposition, and the importance of convexity and its generalizations". Likewise, linear programming is heavily used in microeconomics and company management, such as planning, production, transportation, technology and other issues. Although the modern management issues are ever-changing, most companies would like to maximize profits or minimize costs with limited resources. Therefore, many issues can be characterized as linear programming problems [18].

## 3.3 Constraints and Variables

Constraint is "an equation or inequality that rules out certain combinations of decision variables as feasible solutions". Decision variable "is a controllable input for a linear programming model " [33].

In this thesis, we consider a probabilistic variable "LOSS" which corresponds to the equilibrium value when solved by Linear Programming problem, where minimax (minimizing the maximum gain from the defender side) will be equal to maximin (maximizing the minimum loss from the offender side) in a zero sum two player game. The optimal equilibrium value obtained with Linear Programming lies between minimax and maximin by Neumann Mixed Strategy [33].

In this thesis, for 'n' number of threats, we consider:

i) $n + 1$ variables where the additional variable "LOSS" is the probabilistic variable which is used to minimize the risk

ii) $3n$ constraints, as each threat requires 3 constraints as in equation 3.1 - 3.3.

1. Nonnegativity constraints is a set of constraints that requires all variables to be nonnegative. $NCM_{i,j}$ in below equation is the variable for optimization problem, that is the new counter measure for vulnerability $V_i$ and threat $T_j$ .

$$0 < NCM_{i,j} \leq 1 \tag{3.1}$$

2. Constraints for the improvement of the counter measure, that is to maximize the current CM value. $CM_{i,j}$ in below equation is the counter measure value for vulnerability $V_i$ and threat $T_j$ before optimization.

$$NCM_{i,j} \geq CM_{i,j} \tag{3.2}$$

3. Game-theoretic constraints is used to minimize the loss. In below equation, $P(V_i)$ and $P(T_j|V_i)$ are the probability of vulnerability and the probability of threat respectively.

$$(P(V_i) \times P(T_j|V_i) \times NCM_{i,j}) - 1 \times LOSS < 0 \tag{3.3}$$

iii) The additional two constraints are the Nonnegativity constraint for the additional variable "LOSS" and a constraint to mitigate the total risk to certain percentage respectively as in equation 3.4 and 3.5.

1. Nonnegativity constraint for the additional variable "LOSS" .

$$0 < LOSS \leq 1 \tag{3.4}$$

2. Constraint to mitigate risk to certain percentage value (expressed in decimal for calculation purpose). The optimized total non residual risk (OTNRR), is the sum of the optimized risk for NCM values. $N$ is the goal risk provided by user.

$$OTNRR > (1 - N) \tag{3.5}$$

These constraints are applied for an example in Section 4.4. The results obtained by Risk Assessment System are then verified with Management Science Linear Programming software, which has the following characteristics:

1. A linear objective function that is to be maximized or minimized
2. A set of linear constraints
3. Variables that are all restricted to nonnegative values [33].

# Chapter 4

# Methodology and Implementation

This section describes the methodology used to conduct the security assessment for the system. The methodology consists of the following stages:

1. Data Collection
2. Data Analysis - Identify threats and vulnerabilities
3. Data Storage
4. Design and implementation of a Web application for user Management
5. Risk Analysis and Calculation
6. Risk Optimization

The major activities following the stages, listed above, are as shown below in a flowchart diagram. See Figure 4.1.

## 4.1 Data Collection

This step begins with collection of data from NVD database. Vulnerability and threat data was collected from the data repository of www.cvedetails.com. The data for 15 years ranging from year 1999 to 2014 was downloaded and saved as a CSV (Comma Separated Value) file. Figure 4.2 shows the screenshot from CVE where the data are organized year by year.

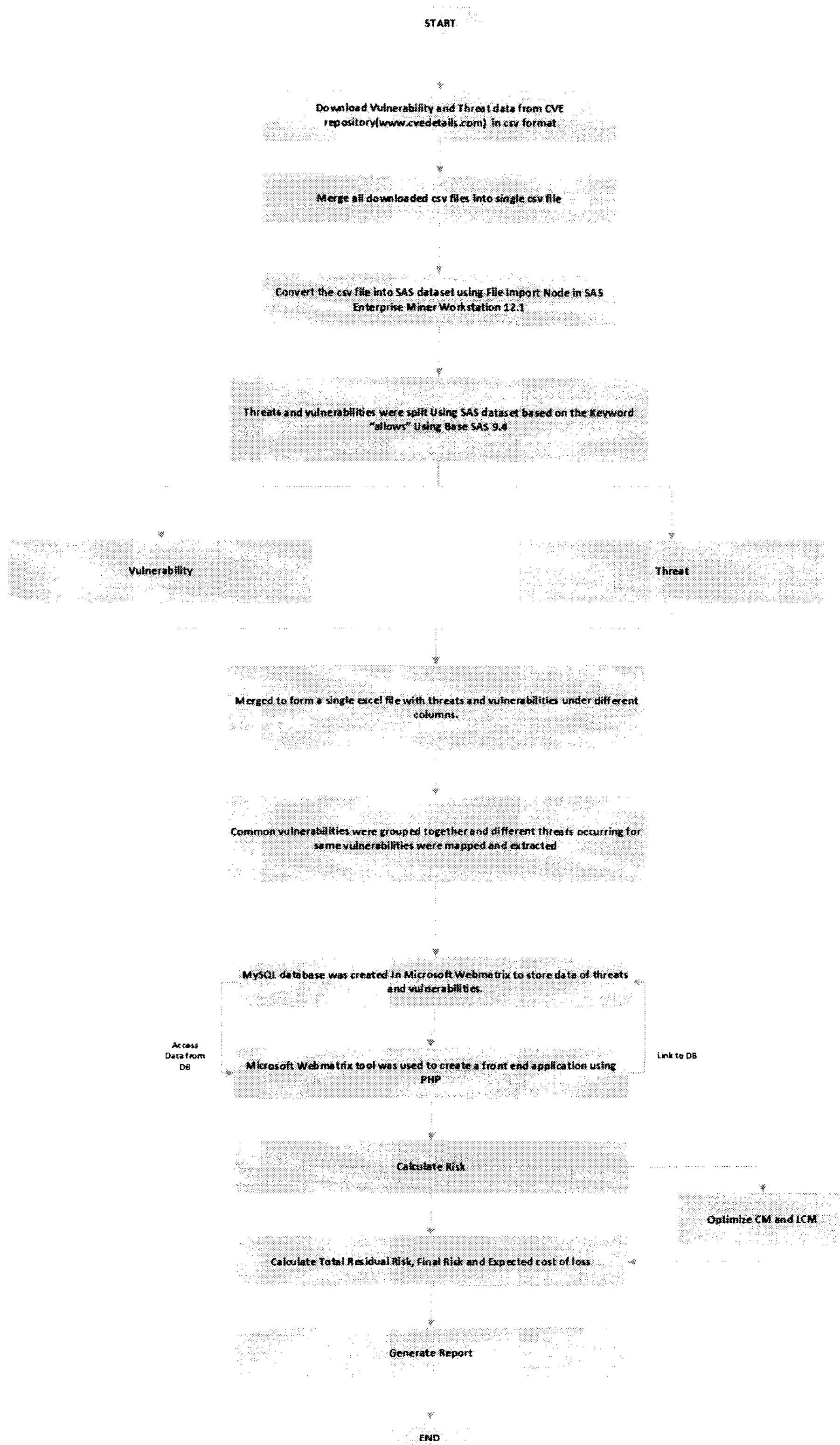



Figure 4.1: Flowchart of activities for developing and implementing the security assessment application

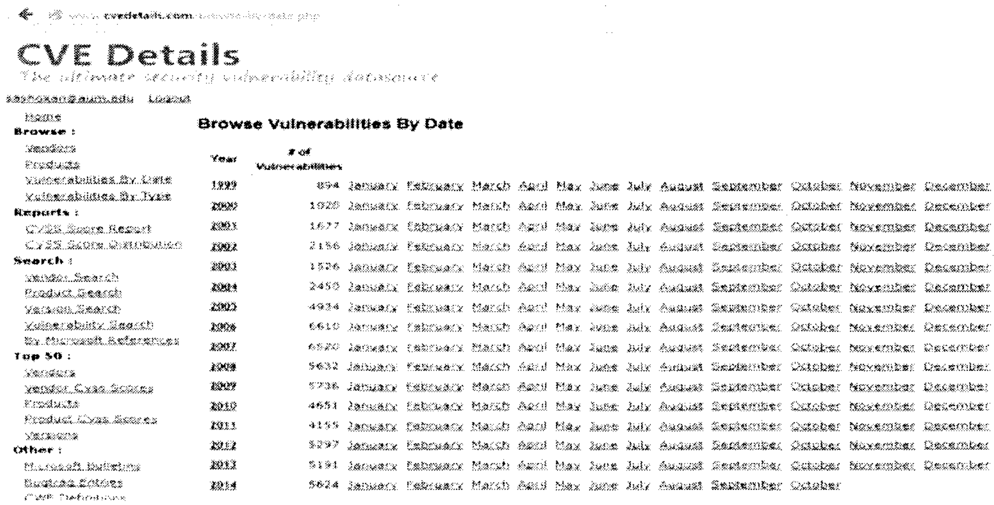
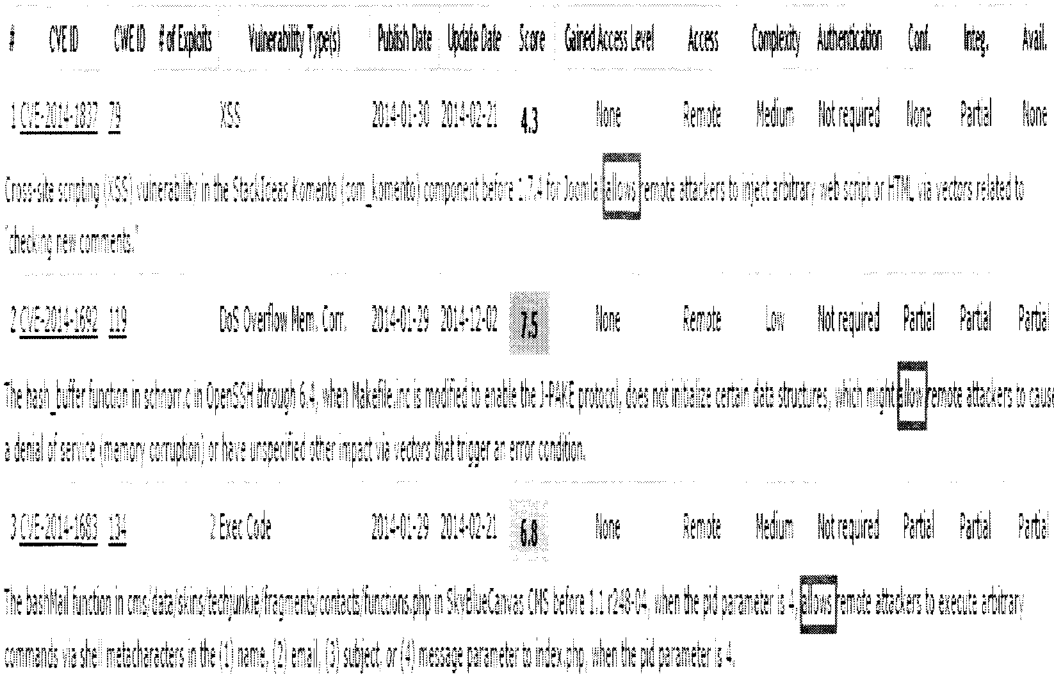



Figure 4.2: CVE Screenshot of vulnerability data according to year wise

The data for all the years was merged into a single CSV file for further analysis. Sample data from CVE website is as shown below in Figure 4.3



| # | CVE ID | CWE ID | # of Exploits | Vulnerability Type(s) | Publish Date | Update Date | Score | Gained Access Level | Access | Complexity | Authentication | Conf. | Integ. | Avail. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | CVE-2014-1837 | 79 | | XSS | 2014-01-30 | 2014-02-21 | 4.3 | None | Remote | Medium | Not required | None | Partial | None |
| 2 | CVE-2014-1692 | 119 | | DoS Overflow Mem. Corr. | 2014-01-29 | 2014-12-02 | 7.5 | None | Remote | Low | Not required | Partial | Partial | Partial |
| 3 | CVE-2014-1683 | 134 | | 2 Exec Code | 2014-01-29 | 2014-02-21 | 6.8 | None | Remote | Medium | Not required | Partial | Partial | Partial |

1: Cross-site scripting (XSS) vulnerability in the StackIdeas Komento (com_komento) component before 1.7.4 for Joomla allows remote attackers to inject arbitrary web script or HTML via vectors related to "checking new comments."

2: The hash_buffer function in schnorr.c in OpenSSH through 6.4, when Makefile.inc is modified to enable the J-PAKE protocol, does not initialize certain data structures, which might allow remote attackers to cause a denial of service (memory corruption) or have unspecified other impact via vectors that trigger an error condition.

3: The bashMail function in cms/data/skins/techjunkie/fragments/contacts/functions.php in SkyBlueCanvas CMS before 1.1 r248-04, when the pid parameter is 4, allows remote attackers to execute arbitrary commands via shell metacharacters in the (1) name, (2) email, (3) subject, or (4) message parameter to index.php, when the pid parameter is 4.

Figure 4.3: Sample vulnerability data from CVE

## 4.2 Data Analysis

The vulnerability and threat data are now cleansed by removing irrelevant data which are not necessary for our analysis. The sample vulnerability and threat data before the data analysis stage is as shown below.

**Sample vulnerability and threat data before data analysis:**

a) Multiple cross-site scripting (XSS) vulnerabilities allow remote attackers to inject arbitrary web script or HTML via the search parameter.

b) Multiple cross-site scripting (XSS) vulnerabilities allow remote attackers to inject arbitrary web script or HTML via the (1) Phone Number field to open.php or (2) Phone number field, (3) passwd1 field, (4) passwd2 field, or (5) do parameter to account.php.

c) Multiple cross-site scripting (XSS) vulnerabilities allow remote attackers to inject arbitrary web script or HTML.

d) Cross-site scripting (XSS) vulnerability allows remote attackers to inject arbitrary web script or HTML via the systemid parameter in a mediaFolder action to index.php.

e) Cross-site scripting (XSS) vulnerability allows remote attackers to inject arbitrary web script or HTML via unspecified vectors, which is not properly handled in an error message.

f) SQL injection vulnerability allows remote attackers to execute arbitrary SQL commands via the article_id parameter in a Submit Comment action.

g) SQL injection vulnerability allows remote attackers to execute arbitrary SQL commands via the catid parameter.

The CSV file containing threats and vulnerabilities was converted into SAS dataset using SAS Enterprise Miner Workstation 12.1. The file import node was used in SAS Enterprise Miner Workstation 12.1 to import the CSV file and convert it into SAS Dataset. In Base SAS 9.3 version, SAS program was written to import the dataset and split the threats and vulnerabilities based on the keyword "allows". The SAS Enterprise Miner screenshot with File Import Node is shown below in Figure 4.4 and screenshot of Base SAS with SAS program is shown in Figure 4.5

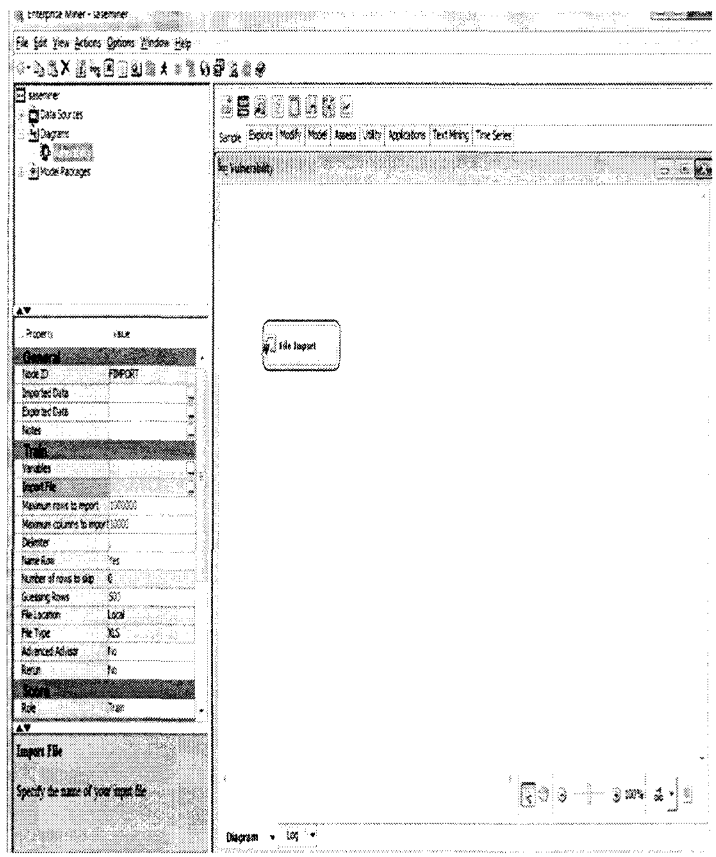



Figure 4.4: SAS Enterprise Miner Screenshot with File Import Node

The resultant data was exported under two different columns as threats and vulnerabilities. The data was merged to form a single excel file with threats and vulnerabilities under different columns. The common vulnerabilities were grouped together and different threats occurring for same vulnerabilities were mapped and extracted under a separate tab in excel file. After the vulnerabilities were classified, we analyzed threats based on keyword and classified unique threats for each vulnerability separately.

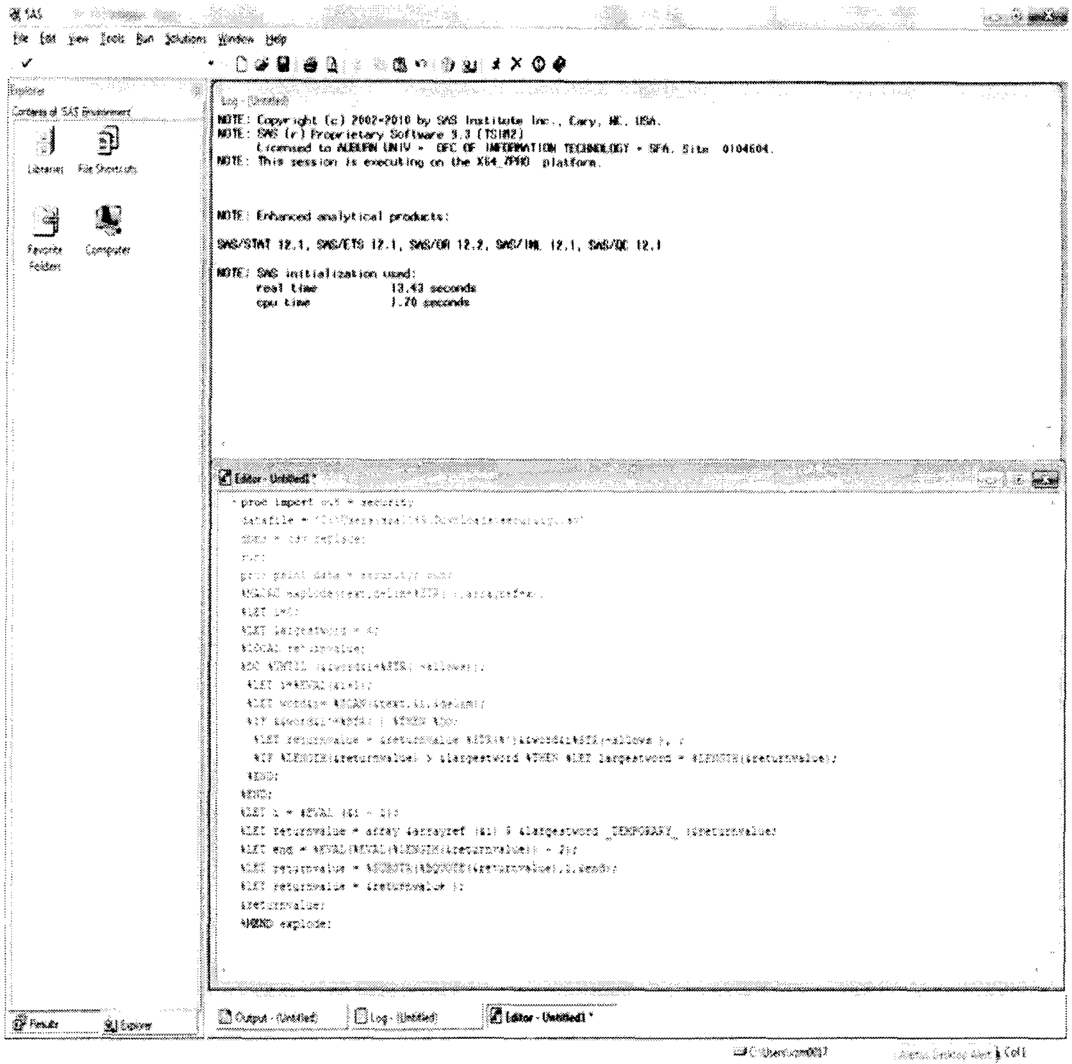



Figure 4.5: Base SAS screenshot with SAS code

The decision to split the threat and vulnerabilities based on "allows" keyword was taken after the analysis of the description of threats and vulnerabilities and finding a pattern in sample data of 2000 rows in excel.

Sample data after Data Analysis is shown in Figure 4.6. After data analysis, below list of vulnerability and threats were found using the data from CVE as a reference as shown in Figure 4.7. The final list correlating vulnerabilities and threats is shown in Table 4.1.

| VULNERABILITIES | THREATS |
|---|---|
| Buffer overflow | allows remote attackers to gain root access using a long PASS command. |
| Buffer overflow | allows root privileges. |
| Buffer overflow | allows local users to execute arbitrary code as root via a long -C (classification) command line option. |
| Buffer overflow | allows local users to execute commands as root. |
| Buffer overflow | allows local users to execute commands with root privileges. |
| Buffer overflow | allow root access. |
| Buffer overflow | allows shell access. |
| Buffer overflow | allows a remote attacker to execute commands. |
| Buffer overflow | allows remote attackers to execute arbitrary code via a UDP packet with a long hostname. |
| Buffer overflow | allow local users to gain root access. |
| Buffer overflow | allows local users to obtain root access. |

| | A | B |
|---|---|---|
| 1 | VULNERABILITIES | THREATS |
| 2 | Denial of Service | allows a remote attacker to cause a crash. |
| 3 | Denial of Service | allows attackers to generate messages. |
| 4 | Denial of Service | allows attackers to register or unregister RPC services or spoof RPC services using a spoofed source IP address such as 127.0.0.1. |
| 5 | Denial of Service | allows attackers to reboot the router using a long URL. |
| 6 | Denial of Service | allows remote attackers to disrupt a user's display. |
| 7 | Denial of Service | allows local users to prevent any server from listening on any non-privileged port. |
| 8 | Denial of Service | allows local users to crash the system. |
| 9 | Denial of Service | allows attackers to cause a Denial of Service (CPU consumption in URC host service). |
| 10 | Denial of Service | allows remote attackers to cause a Denial of Service (possibly CPU consumption) via a SYN flood with malformed TCP packets from multiple connections. |
| 11 | Denial of Service | allows cookie injection. |
| 12 | Denial of Service | allows remote attackers to bypass rulesets and add PHP sequences to a file via unspecified vectors. |

Figure 4.6: Data extracted after Data Analysis

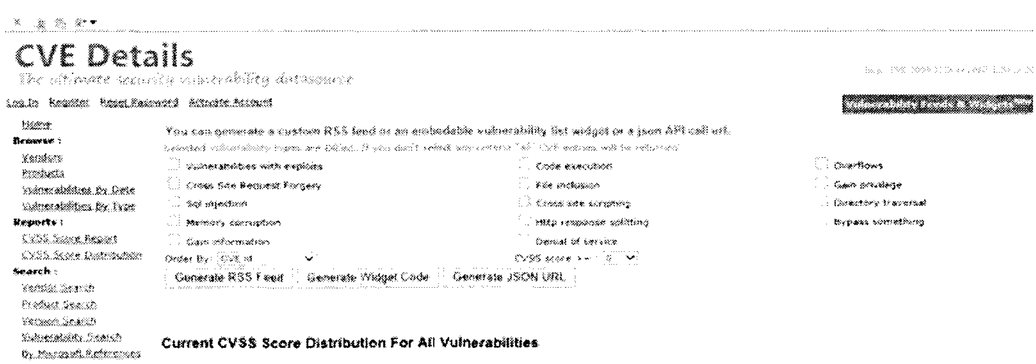



Figure 4.7: CVE Screenshot for vulnerability types

| Vulnerability | Vulnerability Count | Probability of Vulnerability occurrence | Threat | Threat Count | Probability of Threat occurrence |
|---|---|---|---|---|---|
| Buffer Overflow (v1) | 5466 | 0.216999484 | Remote attackers | 4090 | 0.748261983 |
| | | | Local users | 766 | 0.140139041 |
| | | | User-assisted remote attackers | 349 | 0.06384925 |
| | | | Remote authenticated user | 151 | 0.02762532 |
| | | | Context-dependent attackers | 110 | 0.020124405 |
| Total | | | | | 1 |
| Denial of Service(v2) | 10 | 0.000396999 | Attacker | 8 | 0.8 |
| | | | Local users | 2 | 0.2 |
| Total | | | | | 1 |
| Web Server(v3) | 1135 | 0.045059351 | Remote attacker | 877 | 0.772687225 |
| | | | Remote user | 134 | 0.118061674 |
| | | | Local user | 54 | 0.047577093 |
| | | | User-assisted remote attackers | 2 | 0.001762115 |
| | | | Remote authenticated user | 68 | 0.059911894 |
| Total | | | | | 1 |
| JavaScript(v4) | 178 | 0.007066577 | Remote attacker | 169 | 0.949438202 |
| | | | Local user | 1 | 0.005617978 |
| | | | User-assisted remote attacker | 8 | 0.04494382 |
| Total | | | | | 1 |
| Race condition(v5) | 313 | 0.012426059 | Remote attacker | 87 | 0.277955272 |
| | | | Local users | 214 | 0.68370607 |
| | | | Physically proximate attacker | 7 | 0.022364217 |
| | | | Remote authenticated user | 5 | 0.015974441 |
| Total | | | | | 1 |
| Cross Site(v6) | 7274 | 0.288776847 | Remote attacker | 6798 | 0.934561452 |
| | | | Local users | 4 | 0.000549904 |
| | | | User-assisted remote attackers | 34 | 0.004674182 |
| | | | Remote authenticated user | 438 | 0.060214462 |
| Total | | | | | 1 |
| SQL Injection(V7) | 5564 | 0.220890071 | Remote attacker | 5321 | 0.956326384 |
| | | | Local users | 3 | 0.00053918 |
| | | | User-assisted remote attackers | 2 | 0.000359454 |
| | | | Remote authenticated user | 236 | 0.042415528 |
| | | | Context-dependent attackers | 2 | 0.000359454 |
| Total | | | | | 1 |
| Static code injection(V8 | 97 | 0.003850887 | Remote attacker | 81 | 0.835051546 |
| | | | Remote authenticated user | 16 | 0.164948454 |
| Total | | | | | 1 |
| File Inclusion(V9) | 2060 | 0.08178173 | Remote attacker | 2054 | 0.997087379 |
| | | | Remote authenticated user | 6 | 0.002912621 |
| Total | | | | | 1 |
| Format string(v10) | 458 | 0.01818254 | Remote attacker | 331 | 0.722707424 |
| | | | Local users | 80 | 0.174672489 |
| | | | User-assisted remote attackers | 22 | 0.048034934 |
| | | | Remote authenticated user | 16 | 0.034934498 |
| | | | Context-dependent attackers | 9 | 0.019650655 |
| Total | | | | | 1 |
| Http response splitting(v11) | 9 | 0.000357299 | Remote attacker | 9 | 1 |
| Total | | | | | 1 |
| Memory corruption(v12) | 3 | 0.0001191 | Remote attakers | 1 | 0.333333333 |
| | | | Attackers | 1 | 0.333333333 |
| | | | Local users | 1 | 0.333333333 |
| Total | | | | | 1 |
| Directory traversal(V13) | 2267 | 0.089999603 | Remote attackers | 2092 | 0.92280547 |
| | | | Remote authenticated users | 154 | 0.067931187 |
| | | | Remote authenticated administrators | 21 | 0.009263344 |
| Total | | | | | 1 |
| Untrusted search path(V14) | 355 | 0.014093453 | Remote attacker | 43 | 0.121126761 |
| | | | Local users | 309 | 0.870422535 |
| | | | User-assisted remote attackers | 2 | 0.005633803 |
| | | | Remote authenticated user | 1 | 0.002816901 |
| Total | 25189 | 1 | | | 1 |

Table 4.1: Vulnerability and threat data

### 4.2.1 Vulnerability Descriptions

In this section, we are explaining the vulnerabilities and threats found during our analysis stage.

**1. Buffer Overflow:**

A buffer overflow occurs when a program or process tries to store more data in a buffer (temporary data storage area) than it was intended to hold. A buffer overflow or buffer overrun occurs when more data is put into a fixed-length buffer than the buffer can handle. Adjacent memory space becomes overwritten and corrupted. When this occurs bad things happen. Usually system crashes, but also the opportunity for an attacker to run arbitrary code [20].

**2. Denial of Service:**

Denial-of-service (DoS) or Distributed Denial-of-service (DDoS) attack "is an attempt to make a machine or network resource unavailable to its intended users". Services of a host connected to the Internet can be temporarily or indefinitely interrupted or suspended by these DoS attacks[21].

**3. Web Server:**

Camouflage should be "standard issue" for Web servers. The first task of a Web attacker (a cyber criminal, internal or external) is to determine your operating system, Web server, application server and database platforms. The most successful attacks are often targeted attacks, so removing or obfuscating the signatures of your technology platforms – both obvious ones like the server name header or file extensions in HTTP, or the TCP/IP window size, as well as more subtle signatures, like cookie names, HTTP header order, or services running on IP/port combinations is an important type of countermeasure in itself.

### 4. Java Script:

JavaScript enables malicious actors to deliver scripts over the web and run them on client computers. There are two measures that can be taken to contain this JavaScript security risk. "First is sandboxing, or running scripts separately so that they can only access certain resources and perform specific tasks. The second measure is implementing the same origin policy, which prevents scripts from one site from accessing data that is used by scripts from other sites. Many JavaScript security vulnerabilities are the result of browser authors failing to take these measures to contain DOM-based JavaScript security risks" [22].

### 5. Race Condition:

A race condition or race hazard is "the behavior of software or other system where the output is dependent on the sequence or timing of other uncontrollable events". It becomes a bug when events do not happen in the order the programmer intended [23].

### 6. Cross site scripting:

Cross-site scripting (XSS) is "a type of computer security vulnerability typically found in Web applications". XSS enables attackers to inject client-side script into Web pages viewed by other users. A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same-origin policy [24].

### 7. SQL Injection:

SQL Injection is "the hacking technique which attempts to pass SQL commands (statements) through a web application for execution by the backend database". SQL injection must exploit a security vulnerability in an application's software, for example, when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed. SQL injection is mostly known as an attack vector for websites but can be used to attack any type of SQL database [25].

**8. Static code injection:**

Static Code Injection attack consists of injecting code directly onto the resource used by application while processing a user request. This is normally performed by tampering libraries and template files which are created based on user input without proper data sanitization. Upon a user request to the modified resource, the actions defined in it will be executed at server side in the context of web server process.

**9. File Inclusion:**

File inclusion vulnerability enables an attacker to include a file, usually through a script on the web server. It is the type of vulnerability which is usually found on websites. The vulnerability occurs due to the use of user-supplied input without proper validation. This can lead to something as minimal as outputting the contents of the file or more serious events such as [26]: Code execution on the web server Code execution on the client-side such as JavaScript which can lead to other attacks such as cross site scripting (XSS) Denial of service (DoS)

**10. Format String:**

The Format String exploit occurs when the submitted data of an input string is evaluated as a command by the application. In this way, the attacker could execute code, read the stack, or cause a segmentation fault in the running application, causing new behaviors that could compromise the security or the stability of the system [27].

**11. HTTP Response splitting:**

HTTP response splitting is "a form of web application vulnerability, resulting from the failure of the application or its environment to properly sanitize input values". It can be used to perform cross-site scripting attacks, cross-user defacement, web cache poisoning, and similar exploits [28].

**12. Memory Corruption:**

Memory corruption occurs in a computer program when the contents of a memory location are unintentionally modified due to programming errors; this is termed violating memory safety. When the corrupted memory contents are used later in that program, it leads either to program crash or to strange and bizarre program behavior [29].

**13. Directory Traversal:**

A directory traversal (or path traversal) consists in exploiting insufficient security validation / sanitization of user-supplied input file names, so that characters representing "traverse to parent directory" are passed through to the file APIs. The goal of this attack is to order an application to access a computer file that is not intended to be accessible. This attack exploits a lack of security (the software is acting exactly as it is supposed to) as opposed to exploiting a bug in the code [30].

**14. Untrusted Search Path:**

The application searches for critical resources using an externally-supplied search path that can point to resources that are not under the application's direct control. This might allow attackers to execute their own programs, access unauthorized data files, or modify configuration in unexpected ways. If the application uses a search path to locate critical resources such as programs, then an attacker could modify that search path to point to a malicious program, which the targeted application would then execute. The problem extends to any type of critical resource that the application trusts [31].

## 4.3 Data Storage

An Oracle MySQL database will be created to store data of threats and vulnerabilities. MySQL workbench version 6.2.3 is used to create MySQL database and to connect to the web application. MySQL connectors for ODBC, Python, Java, and PHP were downloaded and installed to support and enable remote connection to database.

A new Table was created in database to store vulnerability name, threat name, probability of vulnerability and threat, count of vulnerability and threat, LCM, CM and NRR, Optimized CM and Optimized residual risk. Microsoft Web matrix tool was used to create a front end application using PHP. Connection to the application and Database was established, to enable end user to add data.

## 4.4 Design and implementation of a Web application for user Management

A front end web application, as shown in Figure 4.8, was developed to enable the user to add any new or existing vulnerability and threats. Microsoft Web matrix tool was used for developing a front end application using PHP, Java script, CSS, HTML and Java. The web application was connected to the MySQL database at the backend to store the data in the database. Also it was developed with the functionality to count the number of vulnerability and threats.

The vulnerability and threat data which was extracted and analyzed was uploaded into the database using Excel import function in MySQL. Once the data has been uploaded to the database it will count the number of vulnerabilities and threats and calculate the corresponding probability of occurrence of vulnerability and threats

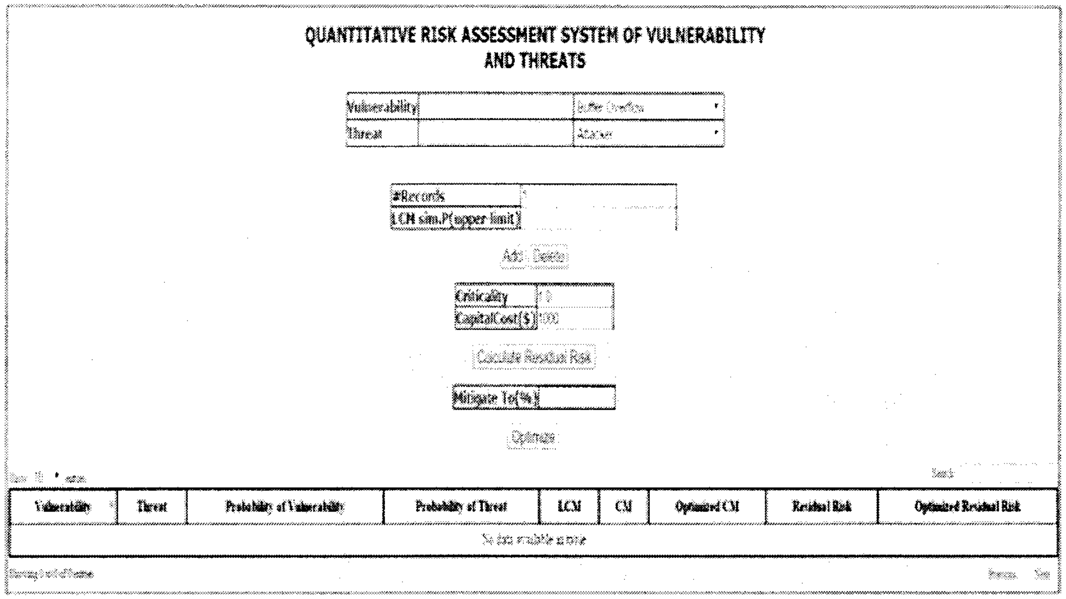



Figure 4.8: Screenshot of Risk Assessment system of vulnerability and threats

as explained in the below scenarios. In the application the data that will be entered by the users are vulnerability and threat name. Description of each field is explained below:

a) #Records field corresponds to the Number of instances users wants to add the vulnerability and threat occurrences to the database.

b) LCM sim.P(upper-limit) field refers to the upper limit of LCM which is defaulted in our application to 0.4 for calculation purpose. Users can change the value as per the requirement. Since we are defaulting the upper limit value to 0.4 the random numbers ranging from 0.0 to 0.4 will be generated as LCM value.

c) Criticality is a constant that indicates the degree of how critical or disruptive the system is in the event of an entire loss. The value can range from 0.0 to 1.0. It is defaulted to 1.0.

d) Capital Cost (Investment Cost) is the total expected loss in monetary units (e.g., dollars) for the particular system. The value is entered by the user.

e) Mitigate To After the countermeasures are generated, users can mention by how much margin they want to mitigate the risk.

f) Add Button: When user enters the new vulnerability and threat and mention the No of Records and click on Add button the corresponding records gets added into the database.

g) Delete Button: User can use the Delete button to, remove any existing vulnerability and threat record from the database.

h) Calculate Button: When user click on Calculate button Total Residual Risk (TRR), Final Risk (FR) and Expected Cost of Loss values are calculated.

i) Optimize Button: When user click on Optimize button by mentioning the Mitigate to value, the Optimized TRR, FR and ECL values are calculated.

The existing vulnerabilities and threats from CVE are displayed in the application in the form of drop down menu as shown below in Figure 4.9 and Figure 4.10.

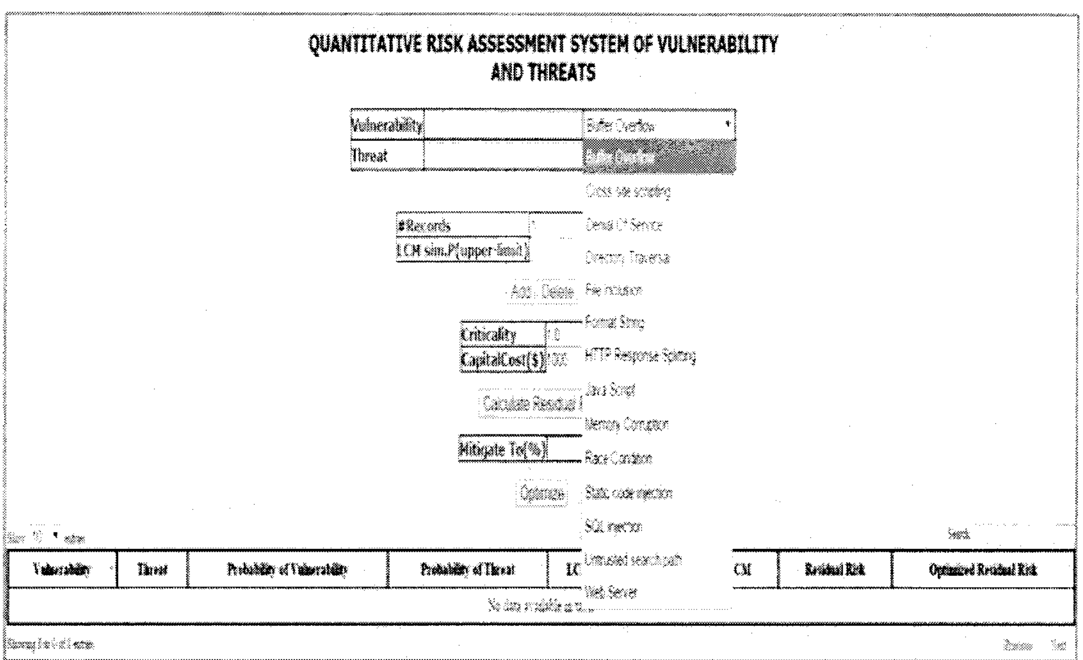



Figure 4.9: Screenshot of dropdown menu of vulnerabilities

We have developed our application to handle the following scenarios.

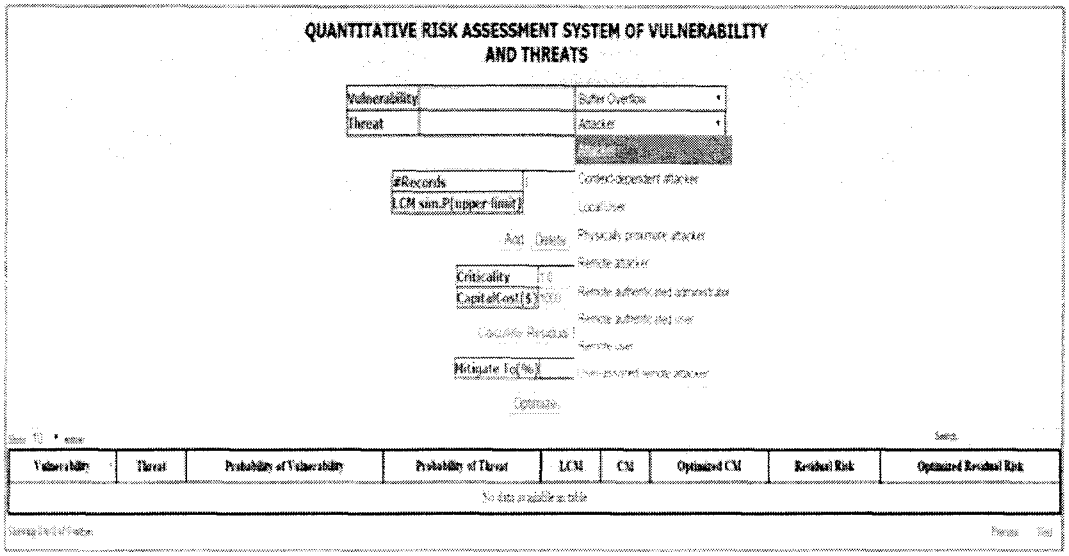


Figure 4.10: Screenshot of dropdown menu of threats

Scenario 1: Users can add any existing vulnerability and existing threat to the database using the dropdown menu and selecting the corresponding vulnerability and threat. When users select the values from dropdown values and click on Add button the vulnerability and threat gets added to the database as shown below in Figure 4.11. When user does not enter any values in vulnerability and threat field, the values selected in the dropdown menu will get added to the database by default.

Scenario 2: Users can add any new vulnerability and new threat or any existing vulnerabilities and threats to the database by entering the values in the vulnerability and threat field. When users enters the values and click on Add button the vulnerability and threat gets added to the database as shown below in Figure 4.12. When user does not enter any values in vulnerability and threat field, the values selected in the dropdown menu will get added to the database by default.

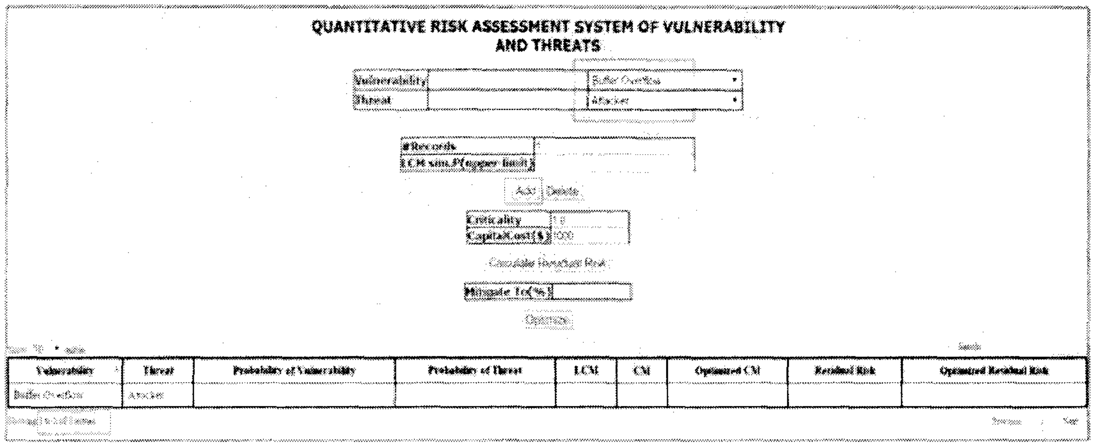



Figure 4.11: Screenshot of Add button function with dropdown value

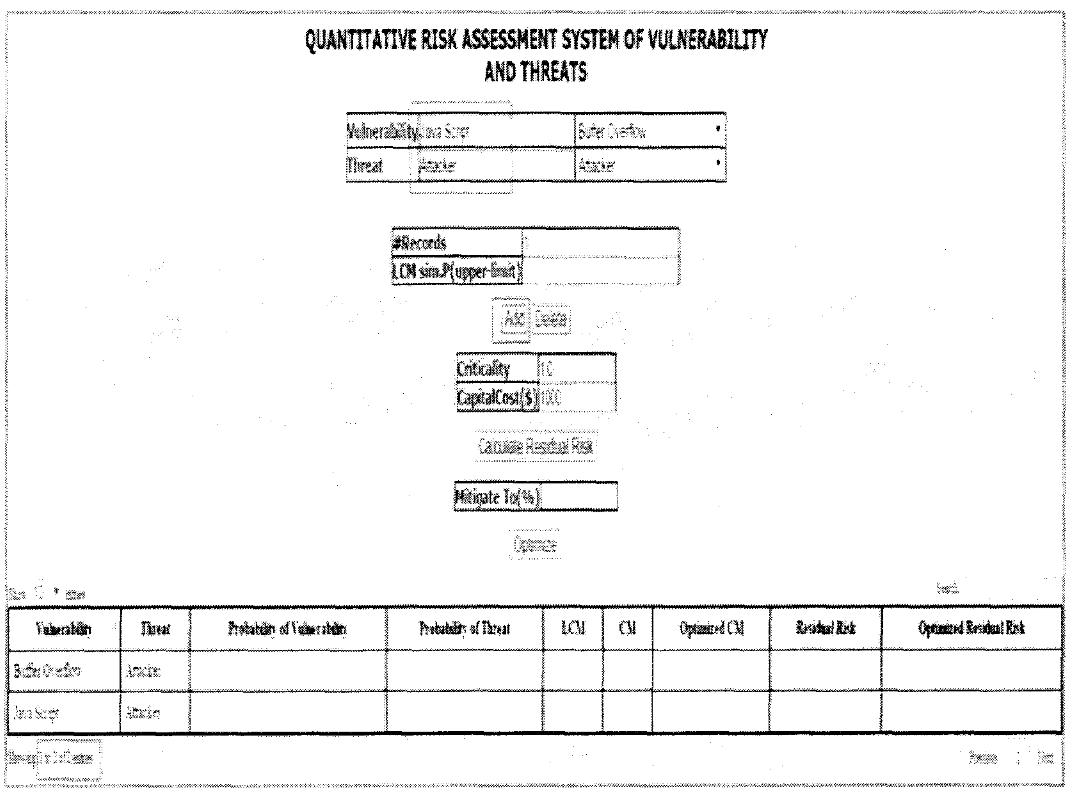



Figure 4.12: Screenshot of Add button function with field entry value

Scenario 3: Users can add many records of vulnerability and threats by mentioning the Number of records they want to add to the database in the field No of

Records. When user select vulnerability and threat from dropdown or by entering the values in the corresponding field and click on Add button by mentioning the number of records, those many records will get added to the database. As shown in the below Figure 4.13 , when user selects number of records as 4, four records gets added to the database and the total count of records is displayed accordingly as shown below. Although 4 records get added only one record will be displayed at the front end application.

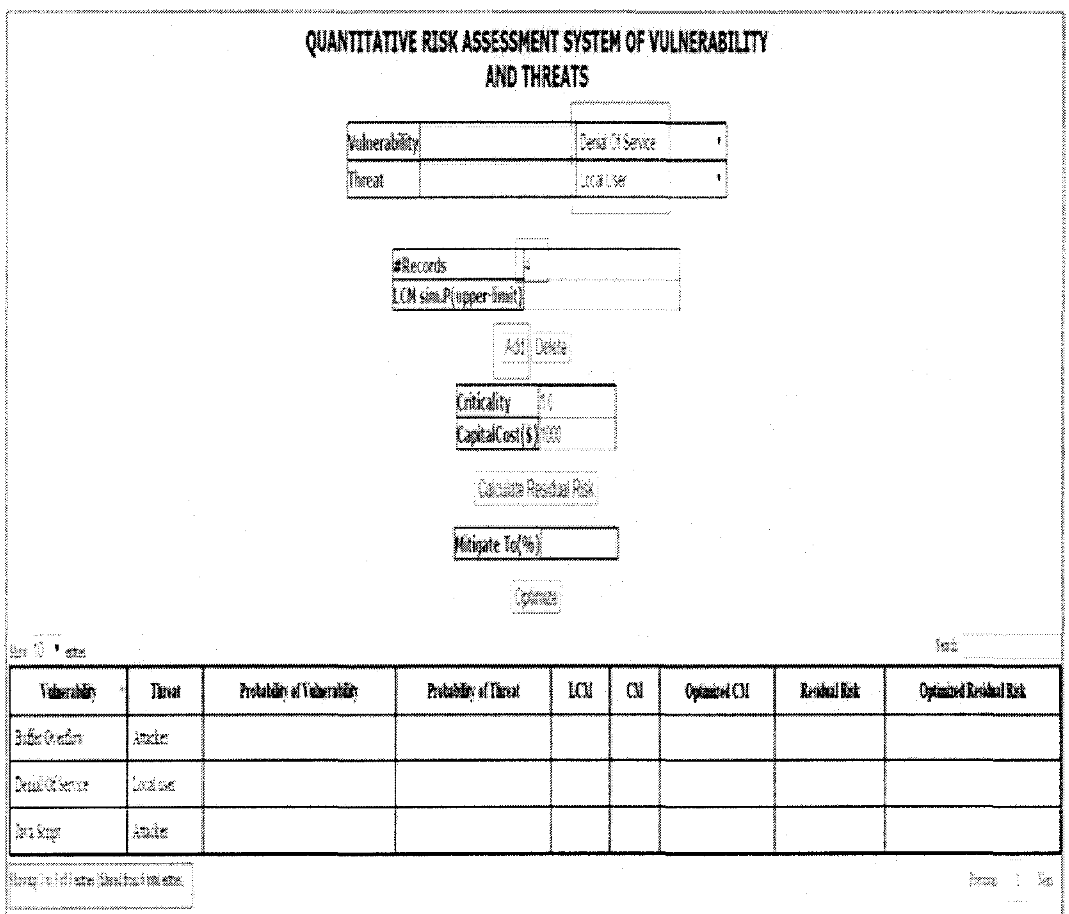



Figure 4.13: Screenshot of Adding multiple number of records

Scenario 4: User can add any new or existing vulnerabilities and threats to the database. When they add an already existing vulnerability and a new threat or any new vulnerability and an existing threat to the database, the data gets inserted in a new row as shown below in the Figure 4.14.

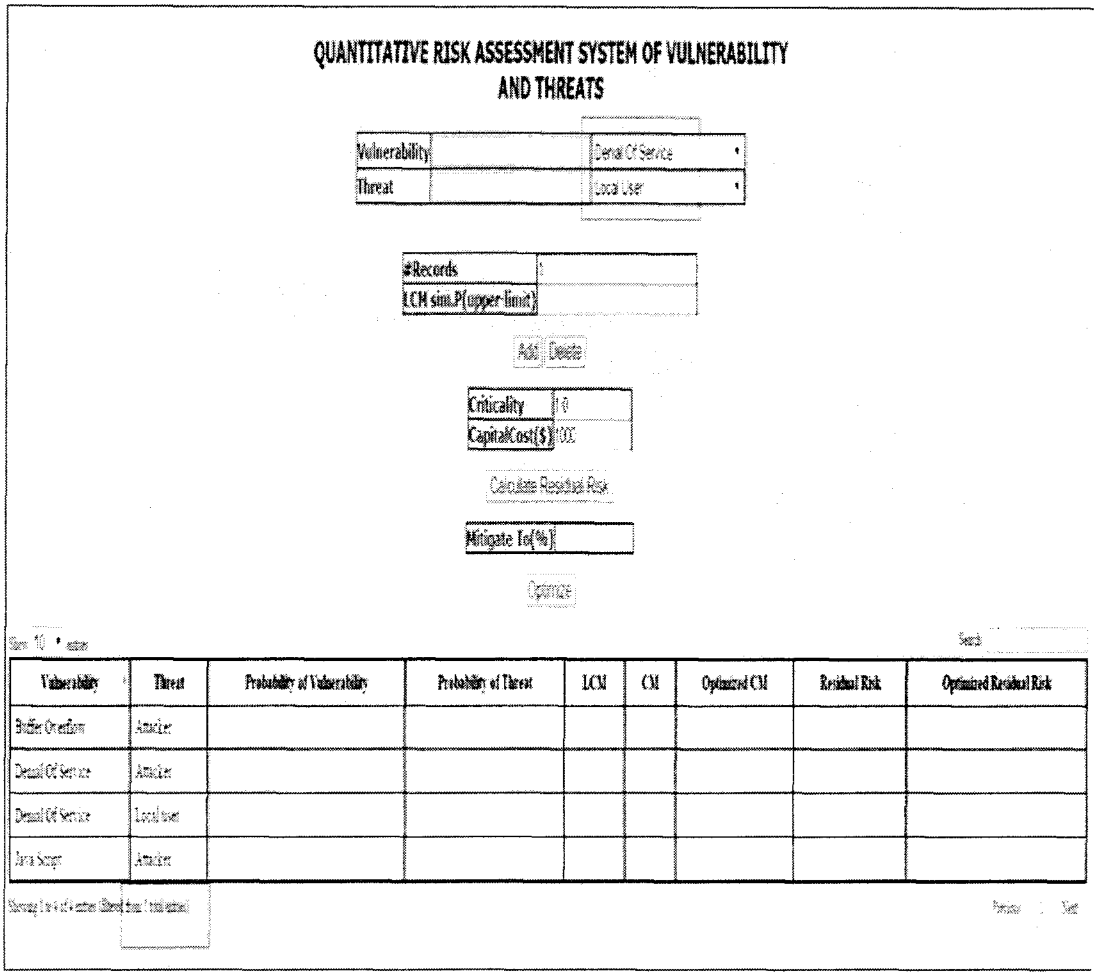



Figure 4.14: Screenshot of Adding existing vulnerability and new threat to the database

Scenario 5: User can delete the records by selecting the vulnerability and threat from dropdown menu or by entering the vulnerability and threat in the corresponding fields. Figure 4.15 and Figure 4.16 shows the records before deletion and after deletion respectively.

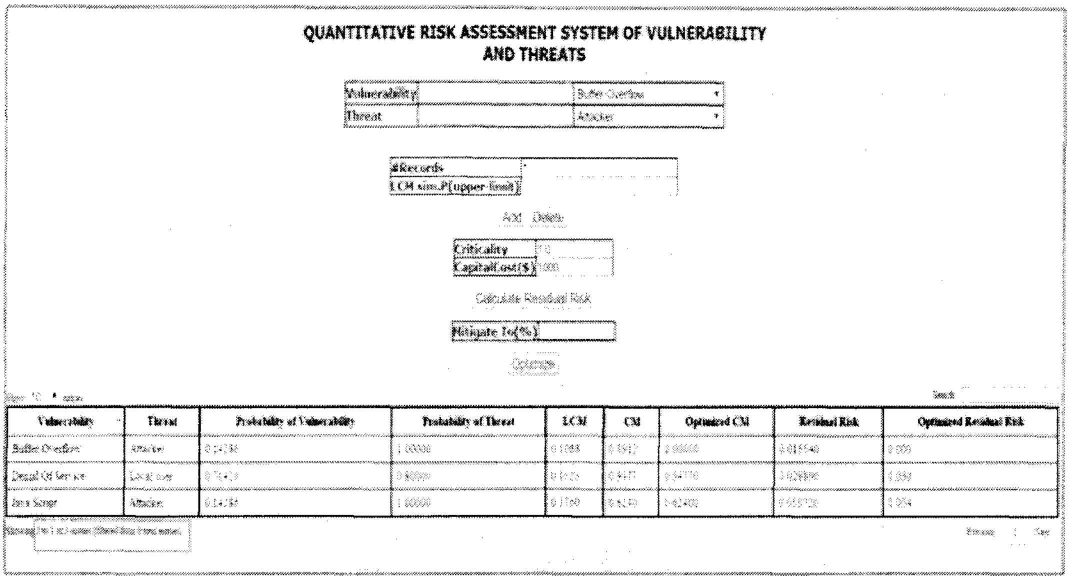



Figure 4.15: Screenshot before deletion of a record

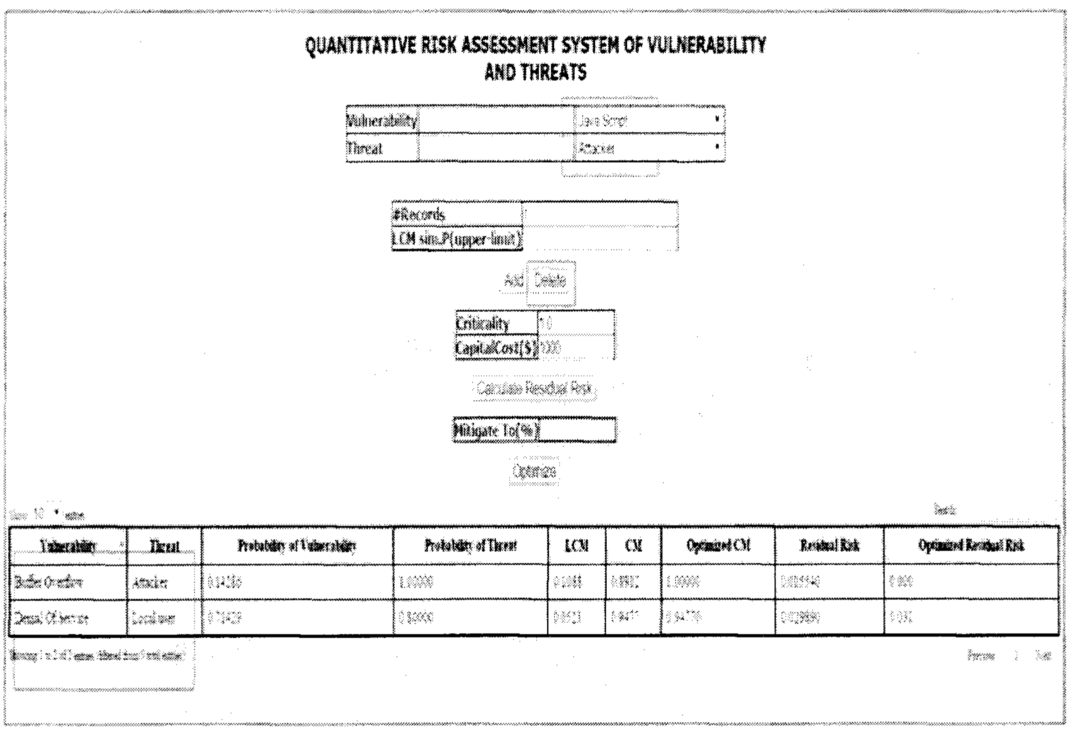



Figure 4.16: Screenshot after deletion of a record

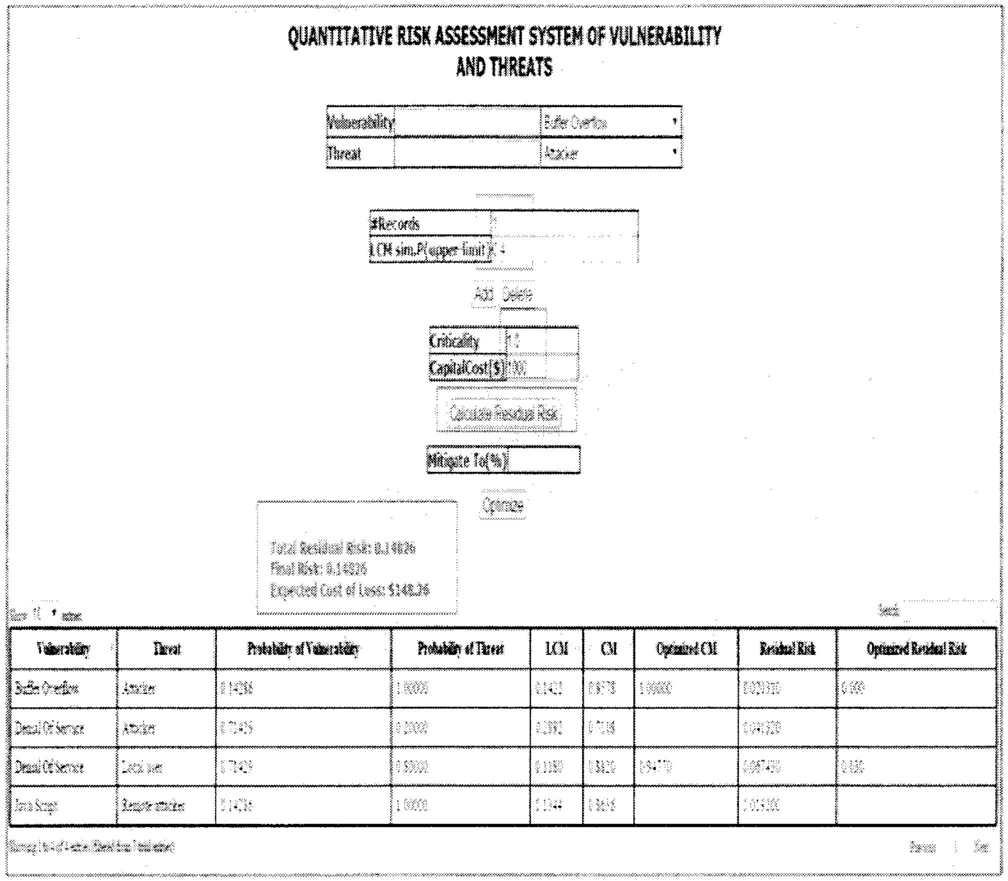



Figure 4.17: Screenshot of calculating residual risk

Scenario 6: After adding the records user can calculate the residual risk values. To calculate the values of residual risk, we are assuming the LCM values to be generated randomly from 0.0 to 0.4 and criticality to be 1.0 and capital cost to be 1000 dollars. After adding the records when user clicks on Calculate Residual Risk button, the following values are calculated and displayed that is Probability of vulnerability, Probability of threat, LCM, CM and RR.

A snapshot of the application after these calculations are done, is shown in Figure 4.17. Using equations (2.1), (2.2) and (2.3) , the below calculations are performed for the vulnerabilities Buffer Overflow(V1) and its threat Attacker(T1), Denial of Service(V2) and its threats Attacker(T1) and Local User(T2), Java Script(V3) and its threats Remote Attacker(T1), as shown in Figure 4.17.

The probabilities of V1, V2 and V3 are calculated as below:

$$P(V1) = \text{Total V1 occurrences / Total no. of vulnerabilities} = 1/7 = 0.14286 \quad (4.1)$$

$$P(V2) = \text{Total V2 occurrences / Total no. of vulnerabilities} = 5/7 = 0.71429 \quad (4.2)$$

$$P(V3) = \text{Total V3 occurrences / Total no. of vulnerabilities} = 1/7 = 0.14286 \quad (4.3)$$

The probabilities of threats are calculated as below:

$$P(T1|V1) = \text{Number of T1|V1 / Total no. of threats under V1} = 1/1 = 1.0 \quad (4.4)$$

$$P(T1|V2) = \text{Number of T1|V2 / Total no. of threats under V2} = 1/5 = 0.2 \quad (4.5)$$

$$P(T2|V2) = \text{Number of T2|V2 / Total no. of threats under V2} = 4/5 = 0.8 \quad (4.6)$$

$$P(T1|V3) = \text{Number of T1|V3 / Total no. of threats under V3} = 1/1 = 1.0 \quad (4.7)$$

RR for all threats are calculated as below:

$$\text{RR for (T1|V1)} = 0.14286 \times 1.0 \times 0.1422 = 0.02031 \quad (4.8)$$

$$\text{RR for (T1|V2)} = 0.71429 \times 0.2 \times 0.2892 = 0.04132 \quad (4.9)$$

$$\text{RR for (T2|V2)} = 0.71429 \times 0.8 \times 0.1180 = 0.06743 \quad (4.10)$$

$$\text{RR for (T1|V3)} = 0.14286 \times 1.0 \times 0.1344 = 0.01920 \quad (4.11)$$

Assuming criticality as 1.0 and capital cost as \$1000, following calculations are performed:

$$TRR = 0.02031 + 0.04132 + 0.06743 + 0.01920 = 0.14826 \quad (4.12)$$

$$FR = 0.14826 \times 1 = 0.14826 \quad (4.13)$$

$$ECL = 0.14826 \times 1000 = 148.26(indollars) \quad (4.14)$$

Scenario 7: Once the residual risk values are calculated, users can optimize the risk by clicking on Mitigate To (%) button. User should enter the percentage margin to which they want to mitigate the risk. This value should be less than the total residual risk calculated previously. Upon clicking on the button the optimized risk values will get calculated as per the equations we have developed and will be displayed. Optimization is done based on the constraints described in equations (3.1)-(3.4). Since there are four threats, 4+1 = 5 variables are used, the additional one is the LOSS and $(4 \times 3) + 2 = 14$ constraints are used, as explained in Section 3.3.

The optimized CM values that is CM11(optimized counter measure for V1 and T1), CM21(optimized counter measure for V2 and T1), CM22(optimized counter measure for V2 and T2), CM31(optimized counter measure for V3 and T1) are obtained for inputs in Figure 4.17, based on the constraints explained in Section 3.3.

Non-negativity constraints:

$$CM11 \leq 1 \quad (4.15)$$

$$CM21 \leq 1 \quad (4.16)$$

$$CM22 \leq 1 \quad (4.17)$$

$$CM31 \leq 1 \tag{4.18}$$

$$LOSS \leq 1 \tag{4.19}$$

Constraints for improvement of the counter measure vector column:

$$CM11 \geq 0.8578 \tag{4.20}$$

$$CM21 \geq 0.7108 \tag{4.21}$$

$$CM22 \geq 0.8820 \tag{4.22}$$

$$CM31 \geq 0.8656 \tag{4.23}$$

Game-theoretic constraints:

$$(0.14286 \times 1.0) \times CM11 - LOSS < 0 \tag{4.24}$$

$$(0.71429 \times 0.2) \times CM21 - LOSS < 0 \tag{4.25}$$

$$(0.71429 \times 0.8) \times CM22 - LOSS < 0 \tag{4.26}$$

$$(0.14286 \times 1.0) \times CM31 - LOSS < 0 \tag{4.27}$$

Constraint for facilitating risk mitigation from 14.83% to 10%:

$$0.14286 \times CM11 + 0.142858 \times CM21 + 0.571432 \times CM22 + 0.14286 \times CM31 > 0.9 \tag{4.28}$$

Optimal Solution: See column 7 in Figure 4.18 of the improved scenario for the newly obtained solution: CM11=1.0, CM21=0.90634, CM22=0.8820, CM31=0.86560.

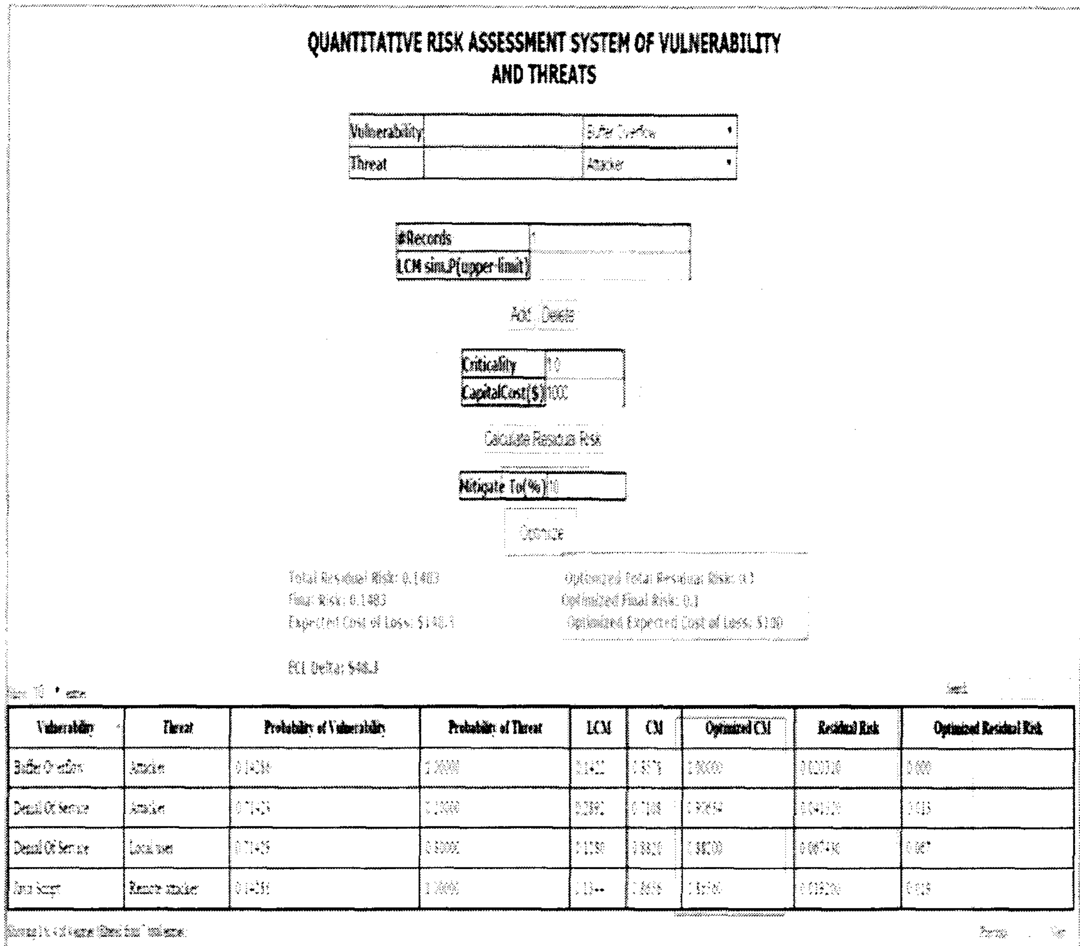



Figure 4.18: Screenshot of Optimizing Risk

Scenario 8: Risk can be optimized to the value not greater than the residual risk percentage. If users enters a value greater than the risk percentage then there will be an validation message or red flag which will be displayed to user as to the value upto which they can optimize the risk as shown in below Figure 4.19.

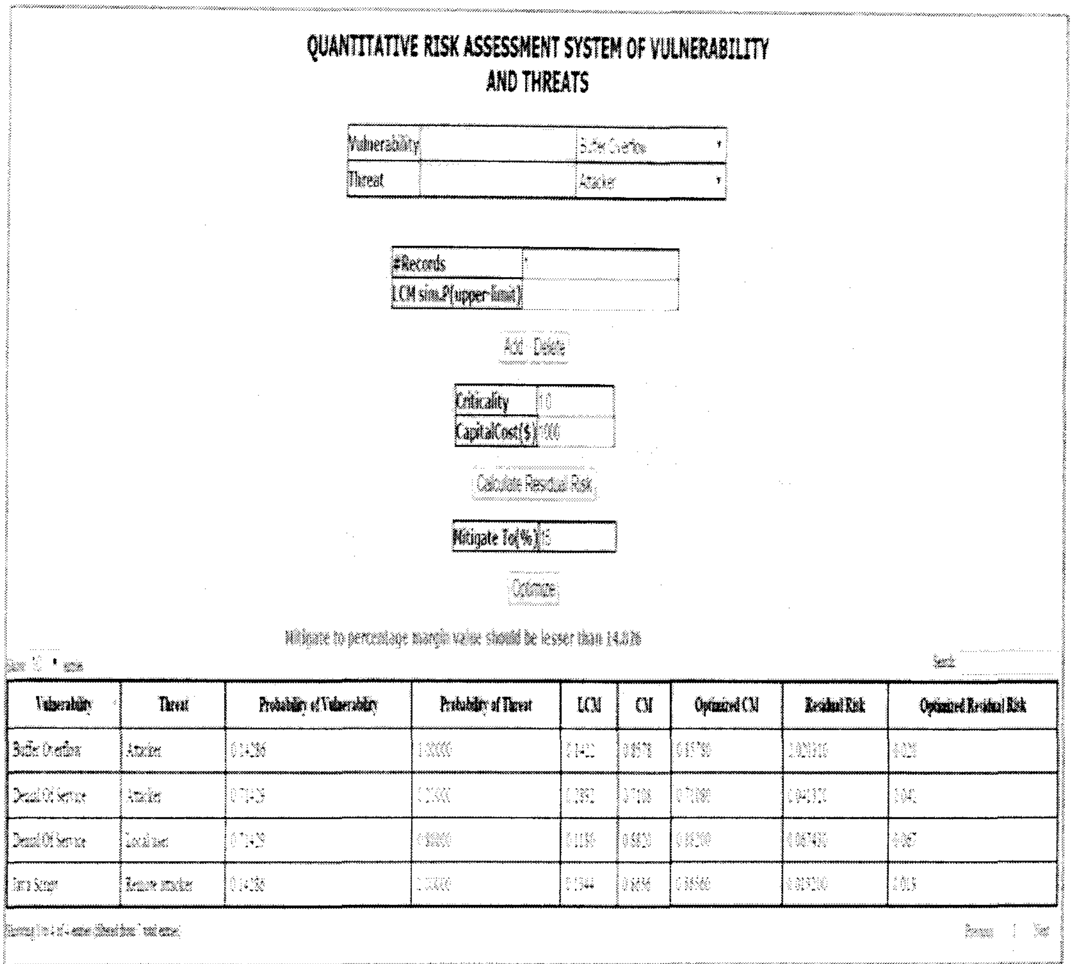



Figure 4.19: Screenshot of Validation message

Chapter 5

# Results and Interpretation

## 5.1 Interpretation

The risk assessment system automatically calculates the RR based on the threat and vulnerability data entered by the user. It also calculates the Expected cost of Loss based on criticality and capital cost entered by user. It also mitigates the RR to percentage entered by user.

## 5.2 Results

We have optimized the risk for large number of records and the results are as shown in the below figures.

Scenario 1: When we optimized the risk for 5000 vulnerabilities, we achieved the below results as shown in Figure 5.1. In this scenario the total residual risk is 0.3378 before optimization. We then chose to optimize the risk to 20%. The counter measures generated after optimization are the OptimizedCM values shown in column 7 in Figure 5.1. As we can see all the CM values are optimized to better values leading to minimal risk. Upon optimizing the risk to 20%, the total residual risk is reduced to 0.2.

If optimized counter measure value is 1, it means that the appropriate countermeasures are in place against a particular vulnerability and threat. In this scenario, the optimized CM value for Buffer Overflow vulnerability and Local User threat is 0.79840 whereas the optimized CM value for Java script vulnerability and Context dependent attacker is 0.80096. Since the optimized CM value is less in case of Buffer

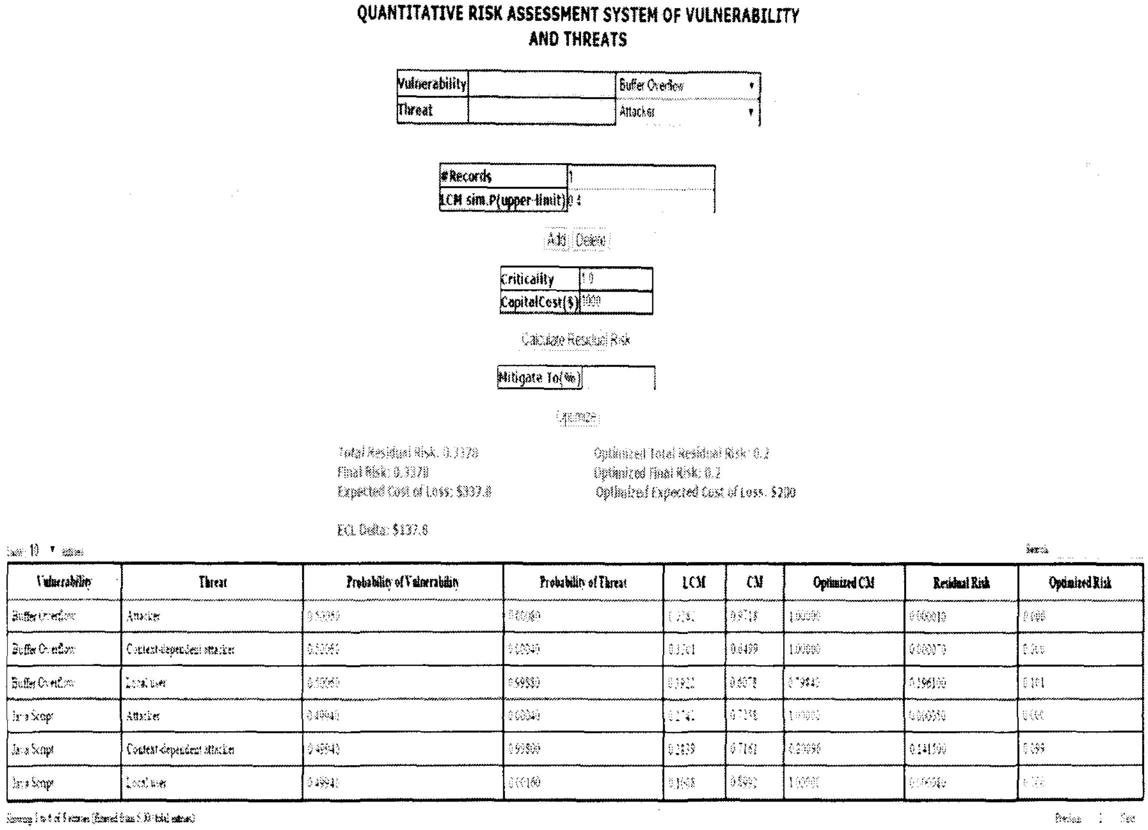



Figure 5.1: Optimized risk for 5000 vulnerabilities

Overflow and Local User, it should be given higher priority over JavaScript and Context dependent attacker while designing countermeasures.

Scenario 2: When we optimized the risk for 8000 vulnerabilities, we achieved the below results as shown in Figure 5.2. In this scenario the total residual risk is 0.329 before optimization. We then chose to optimize the risk to 20%. The counter measures generated after optimization are the OptimizedCM values shown in column 7 in Figure 5.1. As we can see all the CM values are optimized to better values leading to minimal risk. Upon optimizing the risk to 20%, the total residual risk is reduced to 0.2.

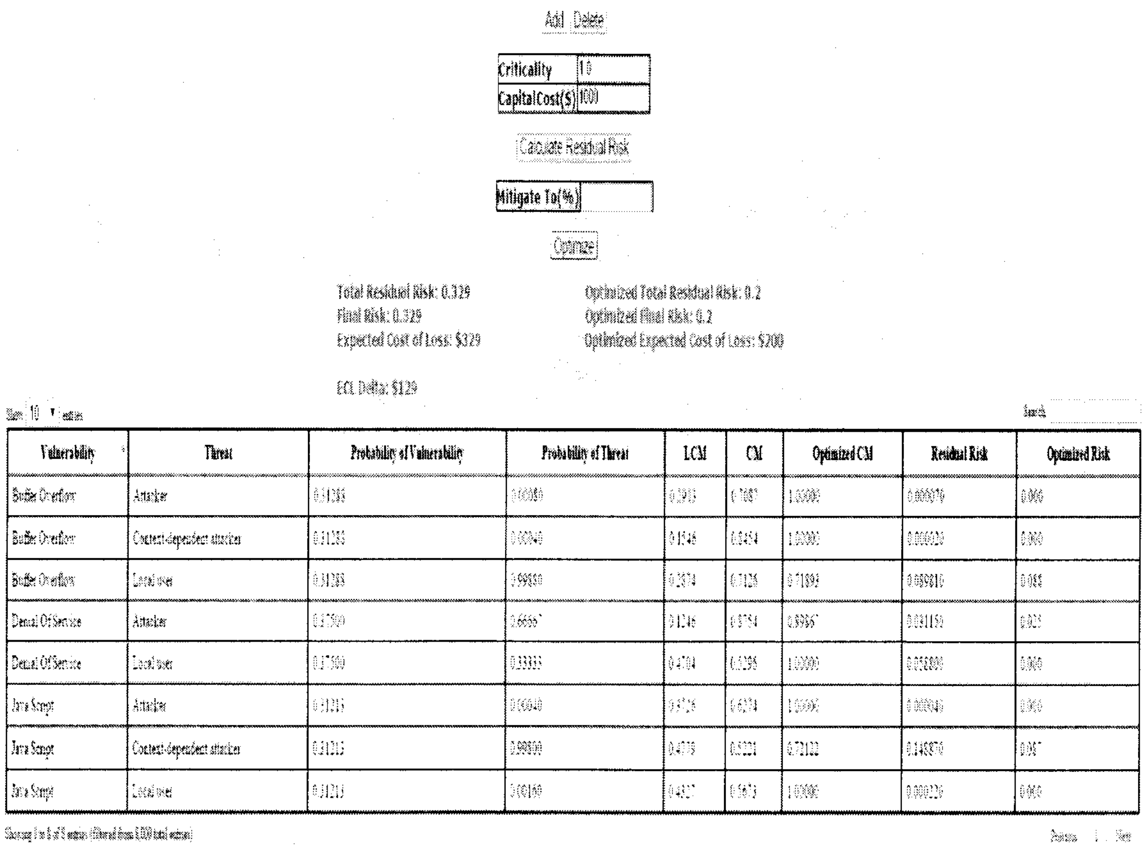



Figure 5.2: Optimized risk for 8000 vulnerabilities

In this scenario, the optimized CM value for Buffer Overflow vulnerability and Local User threat is 0.71893, for Java script vulnerability and Context dependent attacker is 0.72122 and for Denial of Service and Attacker is 0.89867. Since the optimized CM value is less in case of Buffer Overflow and Local User, it should be given higher priority over JavaScript and Context dependent attacker, and Denial of Service and Attacker while designing countermeasures.

## 5.3 Risk Assessment System Results Validation

To validate our system, we have used the software which was openly provided by the Management Science book, An Introduction to Management Science, Quantitative

Approaches to Decision Making, 13e [33]. For this we have considered the below example, as shown in Figure 5.3.

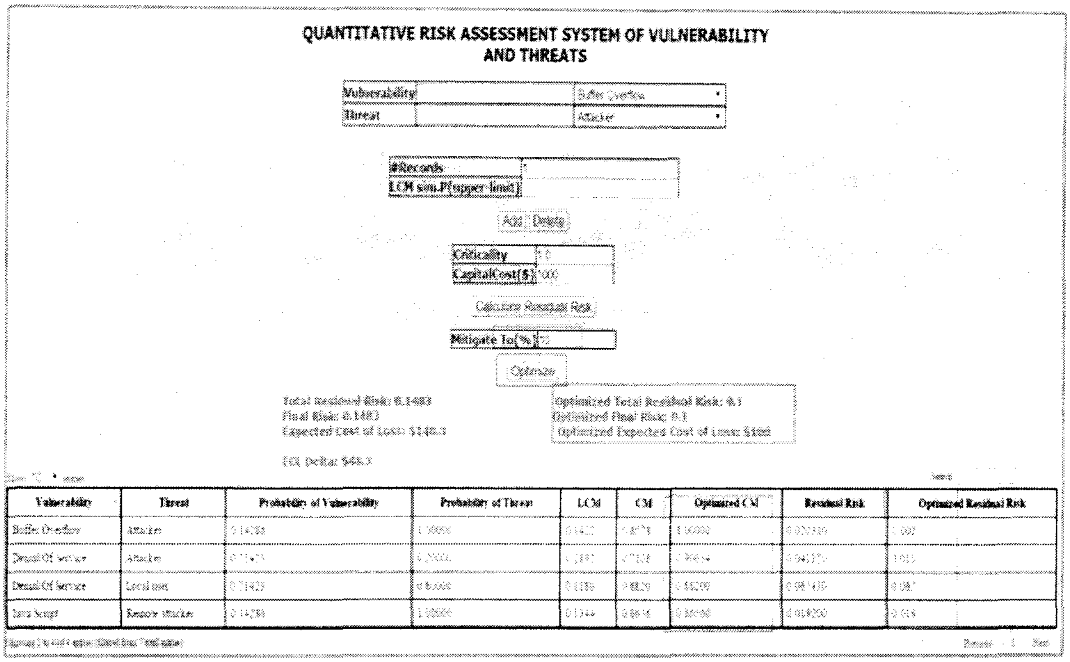
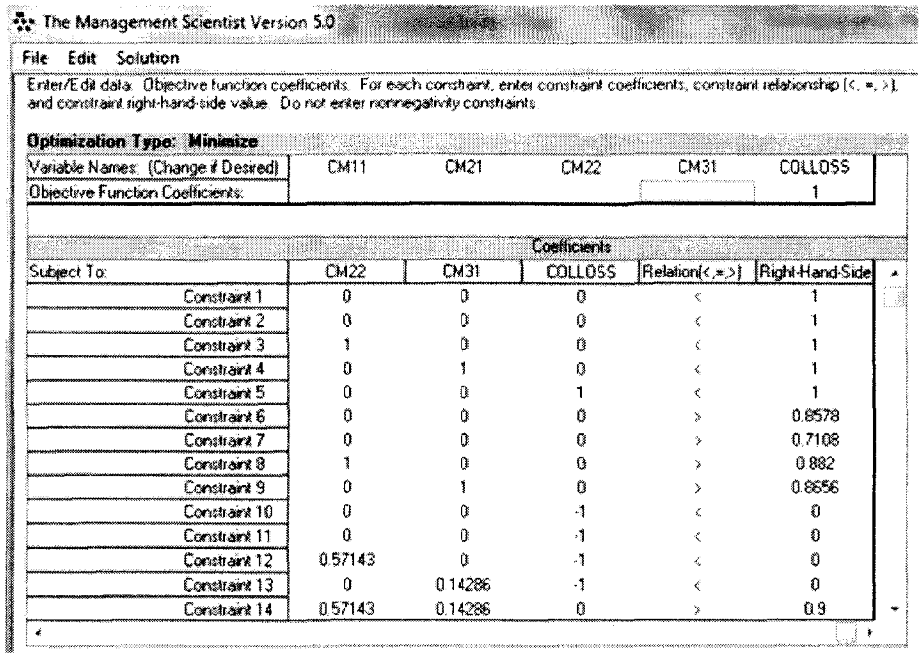


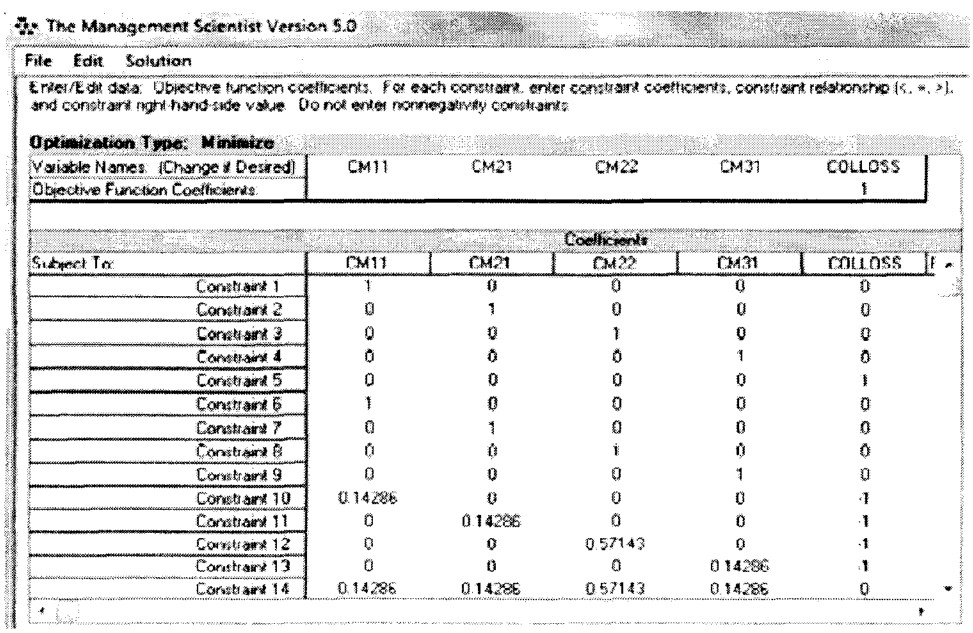

Figure 5.3: Risk Optimization calculation in Risk Assessment System

The same input values, as in Figure 5.3, were entered in Management Science software. The output of Management Science software and Risk Management system remains same after the calculation. Figure 5.4 shows the right part of input values in Management Science Software. Figure 5.5 shows the right part of input values in Management Science Software.




| Subject To: | CM22 | CM31 | COLLOSS | Relation(<,=,>) | Right-Hand-Side |
|---|---|---|---|---|---|
| Constraint 1 | 0 | 0 | 0 | < | 1 |
| Constraint 2 | 0 | 0 | 0 | < | 1 |
| Constraint 3 | 1 | 0 | 0 | < | 1 |
| Constraint 4 | 0 | 1 | 0 | < | 1 |
| Constraint 5 | 0 | 0 | 1 | < | 1 |
| Constraint 6 | 0 | 0 | 0 | > | 0.8578 |
| Constraint 7 | 0 | 0 | 0 | > | 0.7108 |
| Constraint 8 | 1 | 0 | 0 | > | 0.882 |
| Constraint 9 | 0 | 1 | 0 | > | 0.8656 |
| Constraint 10 | 0 | 0 | -1 | < | 0 |
| Constraint 11 | 0 | 0 | -1 | < | 0 |
| Constraint 12 | 0.57143 | 0 | -1 | < | 0 |
| Constraint 13 | 0 | 0.14286 | -1 | < | 0 |
| Constraint 14 | 0.57143 | 0.14286 | 0 | > | 0.9 |

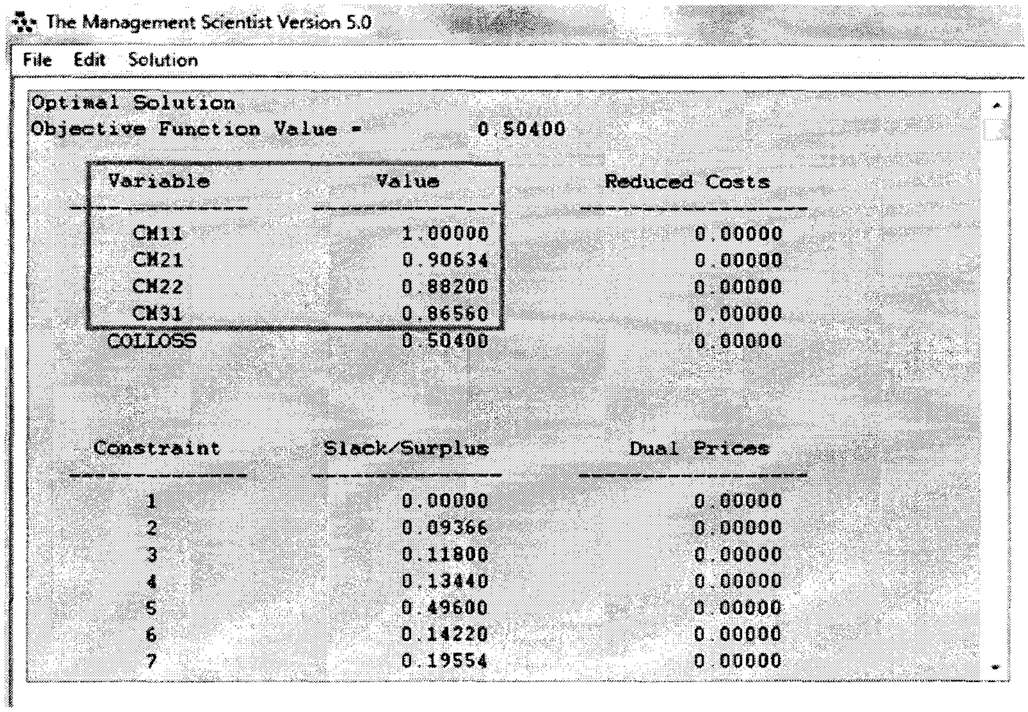
Figure 5.4: Right part of input values in Management Science Software




| Subject To: | CM11 | CM21 | CM22 | CM31 | COLLOSS |
|---|---|---|---|---|---|
| Constraint 1 | 1 | 0 | 0 | 0 | 0 |
| Constraint 2 | 0 | 1 | 0 | 0 | 0 |
| Constraint 3 | 0 | 0 | 1 | 0 | 0 |
| Constraint 4 | 0 | 0 | 0 | 1 | 0 |
| Constraint 5 | 0 | 0 | 0 | 0 | 1 |
| Constraint 6 | 1 | 0 | 0 | 0 | 0 |
| Constraint 7 | 0 | 1 | 0 | 0 | 0 |
| Constraint 8 | 0 | 0 | 1 | 0 | 0 |
| Constraint 9 | 0 | 0 | 0 | 1 | 0 |
| Constraint 10 | 0.14286 | 0 | 0 | 0 | -1 |
| Constraint 11 | 0 | 0.14286 | 0 | 0 | -1 |
| Constraint 12 | 0 | 0 | 0.57143 | 0 | -1 |
| Constraint 13 | 0 | 0 | 0 | 0.14286 | -1 |
| Constraint 14 | 0.14286 | 0.14286 | 0.57143 | 0.14286 | 0 |

Figure 5.5: Left part of input values in Management Science Software

The Optimized CM values from Figure 5.3 is same as the CM11, CM21, CM22, CM31 values from Figure 5.6.



The Management Scientist Version 5.0

File Edit Solution

Optimal Solution

Objective Function Value = 0.50400

| Variable | Value | Reduced Costs |
|---|---|---|
| CM11 | 1.00000 | 0.00000 |
| CM21 | 0.90634 | 0.00000 |
| CM22 | 0.88200 | 0.00000 |
| CM31 | 0.86560 | 0.00000 |
| COLLOSS | 0.50400 | 0.00000 |

| Constraint | Slack/Surplus | Dual Prices |
|---|---|---|
| 1 | 0.00000 | 0.00000 |
| 2 | 0.09366 | 0.00000 |
| 3 | 0.11800 | 0.00000 |
| 4 | 0.13440 | 0.00000 |
| 5 | 0.49600 | 0.00000 |
| 6 | 0.14220 | 0.00000 |
| 7 | 0.19554 | 0.00000 |

Figure 5.6: Output values in Management Science Software

Chapter 6

# Conclusion and Future Work

## 6.1 Conclusion

The risk assessment system helps organizations decide on the necessary security investments in security measures that are most effective for the organization. The basic risk management strategy is to reduce the risk by introducing appropriate technologies, tools and procedures. This reduces the probability of a security incident or damage caused by the incident. The risk assessment system addresses all these issues.

The application we have developed is based on a Quantitative analysis of security risks and it allows for evaluation of different investment options in information security. Risk assessment system leads an organization from the initial input of data to final recommendations for the selection of optimum measure that reduces a certain security risk.

By using risk assessment system, enterprises can easily track any new or existing threats and vulnerabilities that can pose risk to the organization. Based on the risk calculations, users can easily classify the threats and vulnerabilities as high, medium and low. It helps organizations to prioritize the threats and vulnerabilities that are of high importance from the organizations perspective. This would assist organizations to decide on the necessary investments in security measures that are most effective to the organization.

In the process of evaluating the optimal level of an investment in information security it is necessary to quantitatively evaluate the threats and vulnerabilities that are related to an information asset as well as measures to reduce these risks. By using quantitative analysis method for evaluation of vulnerabilities and threats, we are able to calculate the best optimal solutions using the indicators ECL, NRR, RR.

Threat and vulnerability assessment and risk analysis can be applied to any organization. The application software we developed assists in performing threat/vulnerability assessments and risk analysis. It will enable the user to input threats and vulnerabilities and calculate the probability of occurrence of each threat and vulnerability and determine the risk level for each vulnerability and threat based on current or existing countermeasure.

However, introducing a new vulnerability and threat management process within an organization can also be challenging. In order to ensure a successful vulnerability management process, attention should be paid to a number of aspects. First, all roles and responsibilities should be clearly assigned. Ensure all stakeholders within the organization know what to expect. Then the organization needs to filter vulnerabilities and threats that suit the needs of the organization. Sufficient attention should be paid to the configuration and fine tuning of the vulnerability scanner and threat filtering. Finally, when starting out with vulnerability and threat assessment, it is important to limit the scope of the initial vulnerability and threats filtering, to avoid the collection of thousands of vulnerabilities and threats.

## 6.2 Limitation

As the system is being developed to assess the risk using threat and vulnerability data, it is imperative to be accurate while analyzing the threats and vulnerabilities. The data collected from the CVE website was analyzed for 2000 rows of sample data and it was decided to split threats and vulnerabilities based on allows keyword. This has lead to some inconsistency in the threat and vulnerability data. Also the organization wants to store all the data in the database as they want to analyze the data in future. This has resulted in duplicity of data in the database and has resulted in more memory capability and cost of database. Since, the NIST couldn't provide us with the details of the particular threat being intercepted or not, we are assuming the CM and LCM value to be randomly generated between 0 and 1 for our calculation purposes.

## 6.3 Future Work

In the risk assessment system of vulnerabilities and threats, we are calculating the risk assuming the random values for CM and LCM probability (likelihood) values. In order for accurate calculation of risk in accordance to the industry standards laid out by the NIST as created by US-Computer Emergency Readiness Team (CERT), we need to obtain CM and LCM values from the respective organizations. In addition to it we should also obtain weight of the threats so that a particular threat may be more influential than the others in the pool.

As the data keeps updating, we need to obtain these values from organizations in real time for accurate calculation of risk. Also during data cleaning process in Data Analysis stage, we are losing some of the valuable information. In future we should be able to allow the users to enter and store raw data from NVD database and extract information regarding threat and vulnerability from the raw data. To enable storing of raw data, we should implement a non relational database instead of the present relational database. This would allow us to store huge amount of data in our database. We would be enhancing our front end application to allow users to view graphical visualizations and different types of reports. We should also ask for the weight of the threats so that a particular threat may be more influential than the others in the pool. In this thesis we are assuming that all the threats are having equal weight.

To allow different enterprises or organizations and users to store the vulnerability and threat data according to the specifications and standards established by their organization, a new table will be created to store the data as per their needs. This will allow them customize their security needs, by storing data relevant to them and performing risk assessment in an effective manner.

# Bibliography

[1] SANS Information Security Resources.,http://www.sans.org/information-security/. Retrieved April 15, 2015.

[2] Steve Elky., An Introduction to Information System Risk Management.May 31 2006. http://www.sans.org/reading-room/whitepapers/auditing/introduction-information-system-risk-management-1204.

[3] John P. Pironti, Key Elements of an Information Risk Management Program .,Information Systems Control Journal, Volume 2, 2008. http://www.isaca.org/about-isaca/Pages/default.aspx.

[4] Risk Management Standards: High-impact Strategies - What You Need to Know: Definitions,Adoptions, Impact, Benefits, Maturity, Vendors., 24 October 2012.

[5] Avinash Kadam.,Identifying and classifying assets. Issue December 2002 .http://www.networkmagazineindia.com/200212/security2.shtml.

[6] Vulnerability Definitions.,http://en.wikipedia.org/wiki/Vulnerability_(computing). Retrieved April 15, 2015.

[7] CVE. Terminology.,https://cve.mitre.org/about/terminology.html. Retrieved April 15, 2015.

[8] Threat. http://en.wikipedia.org/wiki/Threat_computer). Retrieved April 15, 2015.

[9] National Vulnerability Database.,https://nvd.nist.gov/. Retrieved March 10, 2015.

[10] Common Vulnerabilities and Exposures.,http://en.wikipedia.org/wiki/Common_Vulnerabilities_and_Exposures. Retrieved March 10, 2015.

[11] Common Vulnerabilities and Exposures. CVE Details.http://www.cvedetails.com/. Retrieved April 16, 2015.

[12] Rok Bojanc. Quantitative Model for Information Security Risk Management., Management knowledge and Learning International Conference, 20-22 June 2012 Celje, Slovenia.

[13] Counter Measure: http://en.wikipedia.org/wiki/Countermeasure_(computer). Retrieved April 16, 2015.

[14] Information Security Handbook.,http://ishandbook.bsewall.com/risk/Assess/Risk/inherent_risk.html. Retrieved April 16, 2015.

[15] M Sahinoglu, Trustworthy Computing: Analytical and Quantitative Engineering Evaluation.,Chapter 3. Security Risk Assessment and Management. Wiley Inc. 2007

[16] Risk Optimization.,http://www.iadclexicon.org/risk-optimization/. Retrieved April 16, 2015.

[17] M Sahinoglu Cyber-risk informatics.,Chapter 4. Security Assessment and Management.,Chapter 5.Game-Theoretic Computing, Wiley Inc.[To be published in August 2015].

[18] Linear Programming.http://en.wikipedia.org/wiki/Linear_programming. Retrieved April 16, 2015.

[19] George B. Dantzig, Linear Programming, article originally appeared in 1991. Department of Management Science and Engineering, Stanford University, Stanford, California 94305-4023.

[20] Buffer Overrun Vulnerabilities, Exploits and Attacks.https://www.veracode.com/blog/2012/04/what-is-a-buffer-overflow-learn-about-buffer-overrun-vulnerabilities-exploits-attacks. Retrieved April 15, 2015.

[21] Denial of Service. http://en.wikipedia.org/wiki/Denial-of-service_attack. Retrieved April 17, 2015.

[22] Java Script. http://www.veracode.com/security/javascript-security. Retrieved April 17, 2015.

[23] Race Condition. http://en.wikipedia.org/wiki/Race_condition. Retrieved April 17, 2015.

[24] Cross-site scripting. http://en.wikipedia.org/wiki/Cross-site_scripting. Retrieved April 17, 2015.

[25] SQL Injection. http://en.wikipedia.org/wiki/SQL_injection. Retrieved April 18, 2015.

[26] File Inclusion Vulnerability. http://en.wikipedia.org/wiki/File_inclusion_vulnerability. Retrieved April 18, 2015.

[27] Format String Attack. https://www.owasp.org/index.php/Format_string_attack. Retrieved April 16, 2015.

[28] HTTP Response Splitting. http://en.wikipedia.org/wiki/HTTP_response_splitting. Retrieved April 17, 2015.

[29] Memory corruption. http://en.wikipedia.org/wiki/Memory_corruption. Retrieved April 17, 2015.

[30] Directory Traversal. http://en.wikipedia.org/wiki/Directory_traversal_attack. Retrieved April 18, 2015.

[31] Untrusted Search Paths.https://cwe.mitre.org/data/definitions/426.html. Retrieved April 18, 2015.

[32] M Sahinoglu, L Cueva-Parra, D Ang, Game-theoretic computing in risk analysis, WIREs Comput. Stat, doi: 10.1002/wics, 1205, 2012. http://authorservices.wiley.com/bauthor/onlineLibraryTPS.asp?DOI=10.1002/wics.1205&ArticleID=961931

[33] David Anderson, Dennis Sweeney, Thomas Williams, Kipp Martin. An Introduction to Management Science. Quantitative Approaches to Decision Making. 13th edition, 2012.

[34] M Sahinoglu, An Input-Output Measurable Design for the Security Meter Model to Quantify and Manage Software Security Risk, IEEE Transactions on Instrumentation and Measurement,Vol. 57, No. 6, pp. 1251-1260, June 2008.

Appendix A

PHP code for Risk Assessment System

Main code in PHP used to display a web based user interface:

mainPage.php

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang
   ="en">
<head>
<title>Add Records Form</title>
<meta http-equiv="Content-Type" content="text/html; charset=
   iso-8859-1" />
<link rel="stylesheet" type="text/css" href="//cdn.datatables
   .net/1.10.5/css/jquery.dataTables.css">
<style type="text/css">
#wrapper {
    width: 1000px;
    margin: 20px auto 0;
    font: 1.2em Verdana, Arial, sans-serif;
}
input {
    font-size: 1em;
}
#submit {
    padding: 4px 8px;
```

```
}
h2 {font-size: 26px;
    color: #4d338f;
    margin-left: 0px;
     }
body {
        background-color: rgba(128, 128, 128, 0.23)
    }
table {
        border-collapse: collapse;
        width:  60%;
        height: 20px;
    }
table, th, td{
        border:  2px solid black;
        width: auto;
        font-size: 20px;
    }
select {
  font-size: 20px;
}
</style>
</head>

<body>

<div id="wrapper">
```

```
<form action="newFile.php" method="post">
   <h2><p ALIGN = "center">QUANTITATIVE RISK ASSESSMENT
      SYSTEM OF VULNERABILITY AND THREATS</p></h2>
   <table ALIGN="center">
      <tr style="color: #4d338f">
         <td><b>Vulnerability</b></td>
         <td><input type="text" name="Vulnerability" id="
            Vulnerability" value="" size="25" tabindex="5
            " /></td>
         <td><b><select name="Vulnerabilities" id="
            Vulnerabilities" style="width: 300px;height:
            32px;"><option value="Buffer Overflow">Buffer
            Overflow</option>
             <option value="Cross site scripting">Cross
                site scripting</option>
             <option value="Denial Of Service">Denial Of
                Service</option>
             <option value="Directory Traversal">Directory
                 Traversal</option>
             <option value="File Inclusion">File Inclusion
                </option>
             <option value="Format String">Format String</
                option>
             <option value="HTTP Response Splitting">HTTP
                Response Splitting</option>
             <option value="Java Script">Java Script</
                option>
```

```
        <option value="Memory Corruption">Memory
          Corruption</option>
        <option value="Race Condition">Race Condition
          </option>
        <option value="Static code injection">Static
          code injection</option>
        <option value="SQL injection">SQL injection</
          option>
        <option value="Untrusted search path">
          Untrusted search path</option>
        <option value="Web Server">Web Server</option
          ></select></b>
    </td>
</tr>
<tr style="color: #4d338f">
    <td><b>Threat</b></td>
    <td><input type="text" name="Threat" id="Threat"
       value="" size="25" tabindex="5" /></td>
    <td><b><select name ="Threats" id="Threats" style
       ="width: 300px;height: 32px;"><option value="
       Attacker">Attacker</option>
        <option value="Context-dependent attacker">
          Context-dependent attacker</option>
        <option value="Local user">Local User</option
          >
        <option value="Physically proximate attacker"
          >Physically proximate attacker</option>
```

```
            <option value="Remote attacker">Remote
               attacker</option>
            <option value="Remote authenticated
               administrator">Remote authenticated
               administrator</option>
            <option value="Remote authenticated user">
               Remote authenticated user</option>
            <option value="Remote user">Remote user</
               option>
            <option value="User-assisted remote attacker"
               >User-assisted remote attacker</option></
               select></b>
        </td>
    </tr>
</table>
<br /><br />
<table ALIGN="center">
    <tr style="color: #4d338f">
        <td><b>#Records</b></td>
        <td><input type="text" name="NRecords" id="
           NRecords" value="1" size="25"  tabindex="5"
           /></td>
    </tr>
    <tr style="color: #4d338f">
        <td><b>LCM sim.P(upper-limit)</b></td>
```

```
            <td><input type="text" name="LCMpoint" id="
                LCMpoint" value="" size="25" tabindex="5" /></
                td>
        </tr>
</table>
<P ALIGN ="center"><input type="submit" name="submit"
    value="Add" tabindex="2" style="color: #4d338f;font-
    size: 22px"/>
<input type="submit" name="submit3" value="Delete"
    tabindex="2" style="color: #4d338f;font-size: 22px"/></
    P>
<table ALIGN="center">
        <tr style="color: #4d338f">
            <td><b>Criticality </b></td>
            <td><input type="text" name="Criticality" id="
                Criticality" value="1.0" size="10"  tabindex="
                5"  /></td>
        </tr>
        <tr style="color: #4d338f">
            <td><b>CapitalCost($)</b></td>
            <td><input type="text" name="CapitalCost" id="
                CapitalCost" value="1000" size="10" tabindex="
                5" /></td>
         </tr>
</table>
<P ALIGN ="center">
```

```
<input type="submit" name="submit1" value="Calculate
   Residual Risk" tabindex="2" style="color: #4d338f;font-
   size: 22px"/></P>
<table ALIGN="center">
     <tr style="color: #4d338f">
         <td><b>Mitigate To(%)</b></td>
         <td><input type="text" name="MitigateTo" id="
            MitigateTo" value="" size="10" tabindex="5"
            /></td>
     </tr>
</table>
<P ALIGN ="center">
<input type="submit" name="submit2" value="Optimize"
   tabindex="2" style="color: #4d338f;font-size: 22px"/></
   P>

<?php

$link = mysqli_connect("localhost", "root", "sharmi@08", "
   MySQL56");

// Check connection
if ($link === false) {
    die("ERROR: Could not connect. " . mysqli_connect_error()
       );
}
```

```
// attempt insert query execution

if (isset($_POST['submit'])) {

    // Escape user inputs for security
    $Vulnerability = mysqli_real_escape_string($link, $_POST[
        'Vulnerability']);
    $Threat = mysqli_real_escape_string($link, $_POST['Threat
        ']);

    if ($Vulnerability == null && $Threat == null) {
        $Vulnerability = $_POST['Vulnerabilities'];
        $Threat = $_POST['Threats'];
    }

    $table = mysqli_real_escape_string($link, $_POST['Vul']);
    $nrec = mysqli_real_escape_string($link, $_POST['NRecords
        ']);

    for ($i=0;$i<$nrec;$i++) {
        $sql1 = "SELECT * FROM VulTab WHERE Vulnerability ='
            $Vulnerability'";
        $result1 = mysqli_query($link, $sql1);
        $count1 =  1 + mysqli_num_rows($result1);

        $sql4 = "SELECT * FROM VulTab WHERE Vulnerability = '
            $Vulnerability' AND Threat = '$Threat'";
```

```
    $result = mysqli_query($link, $sql4);
    $count =  1 + mysqli_num_rows($result);

    if ($count1 == 1) {
        $sql2 = "INSERT INTO VulTab(ID,Vulnerability,
           Threat,Vulnerability_Count,Threat_count)
           VALUES ('1','$Vulnerability', '$Threat',1,1)";
        $result1 = mysqli_query($link, $sql2);

    } else if ($count1 > 1) {

        $sql2 = "INSERT INTO VulTab(ID,Vulnerability,
           Threat) VALUES ('2','$Vulnerability', '$Threat
           ')";
        $result1 = mysqli_query($link, $sql2);

        $sql10 = "UPDATE VulTab SET Vulnerability_Count =
           ".$count1." WHERE Vulnerability = '
           $Vulnerability'";
        $result10 = mysqli_query($link, $sql10);

        $sql11 = "UPDATE VulTab SET Threat_Count=".$count
           ." WHERE Vulnerability = '$Vulnerability' AND
           Threat = '$Threat'";
        $result11 = mysqli_query($link, $sql11);
    }
}
```

```
    if ( mysqli_query($link, $sql, $query)) {
        echo "New Records added successfully." ;

    } else {
        echo " " ;
    }
}
if (isset($_POST['submit3'])) {

    // Escape user inputs for security
    $Vulnerability = mysqli_real_escape_string($link, $_POST[
       'Vulnerability']);
    $Threat = mysqli_real_escape_string($link, $_POST['Threat
       ']);

    if ($Vulnerability == null && $Threat == null) {
        $Vulnerability = $_POST['Vulnerabilities'];
        $Threat = $_POST['Threats'];
    }

    $table = mysqli_real_escape_string($link, $_POST['Vul']);
    $nrec = mysqli_real_escape_string($link, $_POST['NRecords
       ']);

    for ($i=0;$i<$nrec;$i++) {
        $sql1 = "SELECT * FROM VulTab WHERE Vulnerability = '
           $Vulnerability'";
```

```
$result1 = mysqli_query($link, $sql1);
$count1 =  mysqli_num_rows($result1);
$count2 = $count1 - $nrec;

$sql4 = "SELECT * FROM VulTab WHERE Vulnerability = '
  $Vulnerability' AND Threat = '$Threat'";
$result = mysqli_query($link, $sql4);
$count =  mysqli_num_rows($result);
$count3 = $count - $nrec;

if ($count == 1) {
    $sql2 = "DELETE FROM VulTab WHERE Vulnerability='
      $Vulnerability' AND Threat='$Threat'";
    $result1 = mysqli_query($link, $sql2);

} else if ($count > 1) {

    $sql2 = "DELETE FROM VulTab WHERE Vulnerability='
      $Vulnerability' AND Threat='$Threat' LIMIT 1";
    $result1 = mysqli_query($link, $sql2);

    $sql12 = "UPDATE VulTab SET Vulnerability_Count=
      ".$count2." WHERE Vulnerability = '
      $Vulnerability'";
    $result12 = mysqli_query($link, $sql12);
```

```
            $sql11 = "UPDATE VulTab SET Threat_Count=".
               $count3." WHERE Vulnerability = '
               $Vulnerability' AND Threat = '$Threat'";
            $result11 = mysqli_query($link, $sql11);
        }
    }
    if ( mysqli_query($link, $sql, $query)) {
        echo "Records deleted successfully." ;

    } else {
        echo " " ;
    }
}
if (isset($_POST['submit1'])) {

    $sql6 = "SELECT DISTINCT Vulnerability,Threat,
       Vulnerability_Count,Threat_Count FROM VulTab ORDER BY
       Vulnerability ASC,Threat ASC";
    $result3 = mysqli_query($link, $sql6);
    $count3 =  mysqli_num_rows($result3);
    $index = 0;

    $sql101 = "SELECT * FROM VulTab";
    $result101 = mysqli_query($link, $sql101);
    $count101 =  mysqli_num_rows($result101);

    for ($x=0; $x<$count3;$x++) {
```

```
    $count4 =  mysqli_fetch_row($result3);
    $res = round(($count4[2]/$count101), 5);
    $sql12 = "UPDATE VulTab SET P_Vulnerability='$res'
      WHERE Vulnerability='$count4[0]'";
    $result12 = mysqli_query($link, $sql12);

    $sql7 = "SELECT * FROM VulTab WHERE Vulnerability='
      $count4[0]'";
    $result4 = mysqli_query($link, $sql7);
    $count6 =  mysqli_num_rows($result4);

    $res1 = round(($count4[3]/$count6), 5);
    $sql8 = "UPDATE VulTab SET P_Threat='$res1' WHERE
      Vulnerability='$count4[0]' and Threat= '$count4
      [1]'";
    $result8 = mysqli_query($link, $sql8);
}

$sql16 = "SELECT DISTINCT Vulnerability,Threat,
   P_Vulnerability,P_Threat FROM VulTab ORDER BY
   Vulnerability ASC,Threat ASC";
$result16 = mysqli_query($link, $sql16);
$count16 =  mysqli_num_rows($result16);

$res4 = 1;
```

```
$lpoint = mysqli_real_escape_string($link, $_POST['
   LCMpoint']);
if ($lpoint == null) {
    $lpoint = 0.4;
}
$lp = $lpoint*10000;
for ($z=0; $z<$count16;$z++) {
    $res7 = round((rand(100, $lp)/10000), 5);
    $res8 = round((1- $res7), 5);
    $count17 =  mysqli_fetch_row($result16);
    $res6 = round(($count17[2]*$count17[3]), 5);
    $res5 = round(($res6*$res7), 5);
    $sql13 = "UPDATE VulTab SET Non_Residual_Risk='$res5
       ', LCM='$res7', CM ='$res8',Prod='$res6' WHERE
       Vulnerability='$count17[0]' and Threat= '$count17
       [1]'";
    $result13 = mysqli_query($link, $sql13);
}

$critic = mysqli_real_escape_string($link, $_POST['
   Criticality']);
$cc = mysqli_real_escape_string($link, $_POST['
   CapitalCost']);

$sql16 = "SELECT DISTINCT Vulnerability,Threat,
   Non_Residual_Risk FROM VulTab";
$result16 = mysqli_query($link, $sql16);
```

```
    $count16 =  mysqli_num_rows($result16);

    $res6 = 0;
    $res50 = 0;
    for ($z=0; $z<$count16;$z++) {
        $count17 =  mysqli_fetch_row($result16);
        $res6 = $res6 + $count17[2];
        $res50 = $res50 + $count17[3];
    }
    $total = $res6*$critic;
    $loss = $total*$cc;
    $total1 = $res50*$critic;
    $loss1 = $total1*$cc;

    $del = abs($loss1 - $loss);
    echo "<b><font color='red'>Total Residual Risk: $res6</
       font></b>";
    echo "<br/><b><font color='red'>Final Risk: $total</font
       ></b>";
    echo "<br/><b><font color='red'>Expected Cost of Loss:
       $$loss</font></b>";
}

if (isset($_POST['submit2'])) {
    try{
        $return_val = 0;
```

```
$sql16 = "SELECT DISTINCT Vulnerability ,Threat ,CM,
   Non_Residual_Risk ,Prod FROM VulTab ORDER BY
   Vulnerability ASC,Threat ASC";
$result16 = mysqli_query($link , $sql16);
$count16 =  mysqli_num_rows($result16);
for ($z=0;$z<$count16;$z++) {
     $count17 =  mysqli_fetch_row($result16);
     $array1[$z] = $count17[2];
     $array2[$z] = $count17[4];
}
$numberOfThreats = $count16;
$mto = mysqli_real_escape_string($link , $_POST['
   MitigateTo']);
$target = (100-$mto)*0.01;

$cms = "";
$vts = "";
for ($i=0;$i<$count16;$i++) {
     if ($i > 0) {
          $cms = $cms.';';
          $vts = $vts.';';
     }
     $cms = $cms.$array1[$i];
     $vts = $vts.$array2[$i];
}
```

```
    exec('java -jar c:\\work\\optimize.jar '.
        $numberOfThreats.' '.$target.' '.$cms.' '.$vts,
        $output, $return_val);
    $max = sizeof($output);

    $sql25 = "SELECT DISTINCT Vulnerability,Threat,CM,
        Non_Residual_Risk,Prod,newCM FROM VulTab ORDER BY
        Vulnerability ASC,Threat ASC";
    $result25 = mysqli_query($link, $sql25);
    $count25 =  mysqli_num_rows($result25);

    $res = 1;
    $res1 = $max-1;
    for ($i=1; $i<$res1; $i++) {
        $count23 =  mysqli_fetch_row($result25);
        for ($y=0; $y<$count25;$y++) {
            $res60 = (1.0000-$output[$i])*$count23[4];
            $sql22 = "UPDATE VulTab SET newCM = '$output
                [$i]',NewResRisk = '$res60' WHERE
                Vulnerability='$count23[0]' and Threat= '
                $count23[1]'";
            $result22 = mysqli_query($link, $sql22);
        }
    }
} catch (Exception $ex) {
        echo "Errors: ".$ex->getMessage();
        echo '<br />';
```

```
}
$critic = mysqli_real_escape_string($link, $_POST['
    Criticality']);
$cc = mysqli_real_escape_string($link, $_POST['
    CapitalCost']);

$sql16 = "SELECT DISTINCT Vulnerability,Threat,
    Non_Residual_Risk,NewResRisk FROM VulTab";
$result16 = mysqli_query($link, $sql16);
$count16 =  mysqli_num_rows($result16);
$res6 = 0;
$res50 = 0;
for ($z=0; $z<$count16;$z++) {
     $count17 =  mysqli_fetch_row($result16);
     $res6 = $res6 + $count17[2];
     $res50 = $res50 + $count17[3];
}
$total = round(($res6*$critic), 4);
$loss = $total*$cc;

$total1= round(($res50*$critic), 2, PHP_ROUND_HALF_DOWN);
$loss1 = $total1*$cc;

$del = abs($loss1 - $loss);

$tar = (1-$target);
$riskPerc = $res6*100;
```

```
$res61 = round($res6 , 4);
$res51 = round($res50 , 2, PHP_ROUND_HALF_DOWN);


if ($tar > $res6) {
    echo "<b><font color='red'>Mitigate to percentage
        margin value should be lesser than $riskPerc</font
        ></b>";
} else {
    echo "<b><font color='red'>Total Residual Risk:
        $res61<span style='padding: 0 120px'> </span>
        Optimized Total Residual Risk: $res51</font></b>";

    echo "<br/><b><font color='red'>Final Risk: $total<
        span style='padding: 0 170px'> </span>
        Optimized Final Risk: $total1</font></b>";

    echo "<br/><b><font color='red'>Expected Cost of Loss
        : $$loss<span style='padding: 0 108px'> </
        span>Optimized Expected Cost of Loss: $$loss1</
        font></b>";

    echo "<br/>";
    echo "<br/><b><font color='red'>ECL Delta: $$del</
        font></b>";
}
```

```
}
// close connection
mysqli_close($link);

?>


</form>
</div>
<div>
  <table id="example" class="display" cellspacing="0" width="
     100%">
        <thead>
            <tr>
                <th>Vulnerability</th>
                <th>Threat</th>
                <th>Probability of Vulnerability</th>
                <th>Probability of Threat</th>
                <th>LCM</th>
                <th>CM</th>
                <th>Optimized CM</th>
                <th>Residual Risk</th>
                <th>Optimized Residual Risk</th>
            </tr>
        </thead>
</table>
</div>
</body>
```

```
</html>

<script src="http://code.jquery.com/jquery-1.11.1.min.js"></
    script>
<script src="http://cdn.datatables.net/1.10.6/js/jquery.
    dataTables.min.js"></script>
<script type="text/javascript">
    $(document).ready(function () {
        $('#example').dataTable({
            "processing": true,
            "serverSide": true,
            "ajax": "nextPage.php"
        });
    });

</script>
```

Code in PHP to help with the server side processing:

nextPage.php

```
<?php

// DB table to use
$table = 'vultab';

// Table's primary key
$primaryKey = 'MyKey';

```

```
// Array of database columns which should be read and sent
   back to DataTables.
// The 'db' parameter represents the column name in the
   database, while the 'dt'
// parameter represents the DataTables column identifier. In
   this case simple
// indexes
$columns = array(
    array( 'db' => 'Vulnerability', 'dt' => 0 ),
    array( 'db' => 'Threat',  'dt' => 1 ),
    array( 'db' => 'P_Vulnerability',   'dt' => 2 ),
    array( 'db' => 'P_Threat',      'dt' => 3 ),
    array( 'db' => 'LCM',       'dt' => 4),
    array( 'db' => 'CM', 'dt' => 5 ),
    array( 'db' => 'newCM', 'dt' => 6 ),
    array( 'db' => 'Non_Residual_Risk','dt' => 7 ),
    array( 'db' => 'NewResRisk','dt' => 8 )


);

// SQL server connection information
$sql_details = array(


    'user' => 'root',
    'pass' => 'sharmi@08',
    'db'   => 'MySQL56',
    'host' => 'localhost'
```

Appendix B

Linear Programming in Java

Java code for calculating risk optimization using Linear Programming:

optimize.java

```
package optimize;


import LP.Constraint;
import LP.LPException;
import LP.LPOptimizer;
import LP.LPProblem;


/**
 *
 * @author msahinog
 */
public class Optimize {


    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {


        int tcount = 10;
        float[] cms = new float[tcount];
```

```
float[] vt = new float[tcount];

int numConstraints = (tcount*3)+2;


//define the problem coefficeints
float[] problemArray = new float[tcount + 1];
for(int i=0; i<tcount; i++){
    problemArray[i] = 0.0f;
}

problemArray[tcount] = 1.0f;

//define the constraints
Constraint[] constraints = defineConstraints2(
  numConstraints, (tcount+1), cms,vt, target);

LPProblem problem = new LPProblem(problemArray,
   constraints );


//minimize the problem
problem.setMinimize(true);
LPOptimizer lpo = new LPOptimizer();
try{
    lpo.execute(problem);
```

```
            float[] fa = lpo.getResults();
            if (fa != null){
                for (int i=0; i<fa.length; i++){
                    System.out.println("x" + (i+1) + " = " +
                        fa[i]);
                }
            }else{
                System.out.println("No results found");
            }
        }catch (LPException lpe){
            System.out.println(lpe.getMessage());
        }

    }

    private static Constraint[] defineConstraints2(int
       numConstraints, int numVariables, float [] cms,float
       [] vt,  float target){

        Constraint[] cons = new Constraint[numConstraints];

        int conCnt = 0;
        //part 1
        for (int i=0; i<numVariables; i++){
            cons[conCnt++] = new Constraint(defineFloat(
                numVariables, i), 1.0f, 0);
        }
```

```
//part 2
for(int i=0; i<numVariables-1;i++){
    cons[conCnt++] = new Constraint(defineFloat(
       numVariables, i), cms[i], 1);
}



//part 3
for(int i=0; i<numVariables-1;i++){
    float[] vals = new float[numVariables];
    for(int j=0; j<numVariables -1;j++){
        vals[j]= 0.0f;
    }
    vals[i] = vt[i];

    //final vlaue of array is -1.0f
    vals[numVariables-1]= -1.0f;

    cons[conCnt++] = new Constraint(vals, 0.0f, 0);
}

//part 4
float[] vals = new float[numVariables];

for(int j=0; j<numVariables -1;j++){
```

```
            vals[j]= vt[j];
        }
        vals[numVariables-1] = 0.0f;

        cons[conCnt++] = new Constraint(vals, target, 1);

        return cons;


    }

    private static float[] defineFloat(int numVariables, int
        idx){
        float[] arr = new float[numVariables];
        for(int i=0; i<numVariables; i++){
            arr[i] = 0.0f;
        }
        arr[idx] = 1.0f;
        return arr;
    }

}
```