12-2011

Quantifying Computer Network Security

Ian Burchett
Western Kentucky University, ian.burchett@wku.edu

QUANTIFYING COMPUTER NETWORK SECURITY

A Thesis
Presented to
The Faculty of the Department of Mathematics and Computer Science
Western Kentucky University
Bowling Green, Kentucky

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science

By
Ian Burchett

December 2011

QUANTIFYING COMPUTER NETWORK SECURITY

Date Recommended __ December 8, 2011________

Dr. Rong Yang, Director of Thesis

Dr. Phillip Womble

Dr. James Gary

Kinchel C Doerner 9-JAN-2012

Dean, Graduate Studies and Research Date

ACKNOWLEDGEMENTS

I would like to acknowledge the immense assistance I have received from Dr. Rong Yang as director of my thesis, and all the time, experience, and consideration she has shared over the course of pursuing the winding path leading to completion of this thesis.

I would like to thank Dr. Phillip Womble for his insight and suggestions regarding the direction of this thesis, and the expertise and resources he made available for me to pursue this work.

I would like to thank Dr. James Gary for being accommodating and participating as a member of my thesis committee.

Lastly, I would like to thank my colleagues, friends, and family which have been supportive in this endeavor, and my entire educational career. I could not have completed this work without their support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

QUANTIFYING COMPUTER NETWORK SECURITY

Ian Burchett December 2011 62 Pages

Directed by: Dr. Rong Yang

Department of Mathematics and Computer Science Western Kentucky University

Simplifying network security data to the point that it is readily accessible and usable by a wider audience is increasingly becoming important, as networks become larger and security conditions and threats become more dynamic and complex, requiring a broader and more varied security staff makeup. With the need for a simple metric to quantify the security level on a network, this thesis proposes: simplify a network's security risk level into a simple metric. Methods for this simplification of an entire network's security level are conducted on several characteristic networks. Identification of computer network port vulnerabilities from NIST's Network Vulnerability Database (NVD) are conducted, and via utilization of NVD's Common Vulnerability Scoring System values, composite scores are created for each computer on the network, and then collectively a composite score is computed for the entire network, which accurately represents the health of the entire network. Special concerns about small numbers of highly vulnerable computers or especially critical members of the network are confronted.

CHAPTER 1: INTRODUCTION

Computer security becomes more important every day. With more and more critical assets being digital information stored on computers, transmitted across networks of computers, and available for access remotely, the number of ways that criminals can steal, modify, or destroy data are ever increasing [1].With this rising concern and focus on computer security, many administrators are looking to begin security auditing on their networks, and to maintain a security policy that helps them protect their assets. Identifying the condition of a network in order to remediate the security vulnerabilities it has, is a major task.

How to be informational about the security level of a network without describing the network in its entirety? Likewise, how can someone without extensive technical skill understand the report, and thereby have an understanding of the network security? Is there a way to simplify this security condition or report into an understandable form? This is the problem propose to solve. By simplifying network security into a quantified, simple value, effectively more information is gained from the reduction, in that the information is usable more readily, easily comparable, and reachable by a wider audience.

1.1 Motivation

Simplifying network security reports to a point where they reach a wider audience, and are easier to understand will ultimately increase the security level in computer networks. An increased understanding of the network security condition is necessary, in that currently network security reports are verbose and difficult to understand. Security experts agree that long, complicated network security reports are generally ignored by administration and clients, so reports must be concise and readable by the target audience (not the security experts conducting the security reviews) [9]. Network security reports are not much use if there is no one capable of digesting them and producing a response to the current security condition. Increasing computer security rests on better informing those in a position to make a difference, and enhance security on the system. Pursuant to this goal of making network security conditions on a computer network understandable by a wider audience, and in order to make network security quantifiable, thereby making the information applicable to more mathematical models, this research proposes to quantify the network security condition in to a simple metric. By simplifying this information into a usable state, the network security report will be more understandable, in a standard form, and thereby utilizable by administration, or network technicians, in order to remediate the problems on their network. More usable information in the hands of network administrators will enable networks to be more secure, in that a better understanding of the security “health” of the network will be imparted, enabling action to be taken if needed.

1.2 Goals

This research aims to simplify computer network security reports into a single scored metric, which gives a very accurate idea of just how secure a collection of machines on a computer network are. This metric will reflect the entire network’s security, as well as individual machines having an impact on the score. The score will properly reflect the level of security on the sample characteristic networks that the experiments will operate upon in this research. Generating a simple, quantified metric for network security could also enable integration of this method into future quantitative network security work.

This research's approach is quantifying network security by first getting the condition of the network, and then determining scores for its members, ultimately generating a composite score for the entire network. Obtaining a network condition is done by reading in a network condition, something similar to a network security scan, and using this as a representation of the network. The network condition is then matched against a list of known security vulnerabilities to scan for the signature of on the network's member machines. The presence of these vulnerabilities is determined by matching the network condition to the vulnerability signatures in a well-known security vulnerability database. A composite score for the entire network is then generated by first scoring each machine on the network, and then combining the scores for all the machines on the network into one composite score, via algorithms discussed later.

1.3 Overview

After the abstract, motivation, and goals of this research have been discussed in chapter 1, the remainder of the thesis is laid out as follows:

Chapter 2 provides the background for computer security through an overview of the common problems, the scale of risk by computer security vulnerabilities and solutions to computer and network security. Chapter 3 discusses the National Institute for Standards and Technology's National Vulnerability Database and its related components.

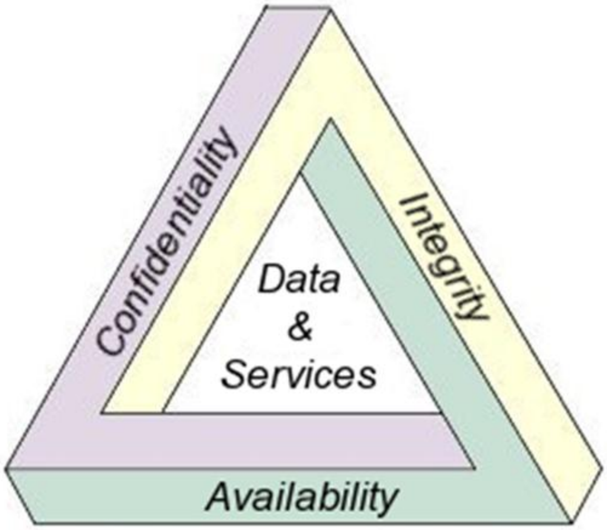
Discussion of its scoring system, CVSS, is also included. This database and the scoring system are utilized within the thesis work to provide a foundation of nationally accredited scores for network vulnerabilities, to which the thesis expands to the machine and then network level in order to get composite scores at these respective levels.

The main work of the thesis, the research towards simplification via quantification is presented in chapter 4. A few different approaches to the quantification of the network security report are demonstrated and compared. Chapter 5 discusses the implementation of the proposed algorithms and framework for quantifying the network. Chapter 6 explores the experimental paradigm to confirm the performance of the quantification algorithms. Comparison of the scores from the proposed compositing methods will be provided. Finally, chapter 7 discusses the efficacy of the proposed methods employed to quantify the network and suggests future work which goes beyond this thesis.

CHAPTER 2: BACKGROUND

Computer security is concerned with the protection and assurance of digital information. The protection of computers containing information via isolation from real and potential threats, and the confirmation that the data is not compromised through unknown threats is the goal. This security must be achieved while users and systems are undisturbed and business continues as normal, though; a security system has failed if the data to be protected is cut off from those whom it is intended for. The classic security mantra of maintaining confidentiality, integrity, and availability (the CIA Triad as shown in Figure 2.1 [36]) are what computer security marches on.

Figure 2.1: The CIA Triad



Confidentiality is maintaining the secrecy of data which must not be allowed to be seen by all clients. Data which is secret or sensitive falls within this category. Nearly all computer data falls under this scope when concerned with threats external to a computer system, in that there are many ways to compromise a computer, with most concerned with some information, which should be secret on a machine, such as passwords, system configuration, or system status. Major concerns with confidentiality are obvious things such as top secret documents, credit card information, and private records. Concerns over confidentiality are so strong today that many physicians will not even adopt modern digital medical records, due to concerns that the confidentiality of the data may be compromised, since this is a situation where near-absolute confidentiality needs to be assured [10]. Maintaining confidentiality is typically achieved via encryption of the data to be protected, in order to prevent access to the data, even if the data storage is breached by an attacker. Also, prevention of unauthorized access to computer systems is used in order to secure the data, in order to keep it from being copied or viewed.

Integrity of data must be maintained, meaning the data must be kept in its original form, unmodified, uncorrupted, and as stored. Data can not only be stolen, but it can be modified as well, and left in place. A classic example of this is to modify computer system passwords in order to create authorized access to systems by attackers, effectively creating their own logins, and free-reign on the system, particularly if the account is in an administrator role. Data stored on machines must also be protected from modification, to guarantee when it is accessed after being stored, it is true to its original form, and so malicious changes are not made to important information. Malicious code can also be inserted into seemingly legitimate software, modifying the software, potentially changing the functionality of the software, and perhaps disrupting the system and its data further. Some code modifications such as this have been known to cause complete data loss on systems through modification of operating systems [11]. Maintaining data integrity is necessary in order to ensure the data is as intended, and not changed by malicious parties.

Availability must be maintained in order to keep the data which is protected by security, available to those whom are authorized to view or change the data. Security has failed if everyone is kept out, including those who should pass security. Striking the balance between securing the network too tightly, and thereby preventing legitimate access, and securing the network too lightly, and thereby allowing too great a chance of malicious access, is a difficult task. This balance is a major concern with computer security, particularly when concerned with security which operates on a continuum of security, such as computer firewall configuration, or network filter administration. Data becoming unavailable due to over-zealous security procedures is one of the most common failures of computer security today, particularly from client and users' viewpoint [12]. Even if a computer system is never compromised in any way, clients are likely to have encountered cases where legitimate requests on their part for information were impeded or prevented due to such overbearing security procedures. The balance between data available to everyone and data available to no one is a continuum on which computer security administrators and experts must balance very carefully.

Network security is a growing concern with economies and global assets moving online, and becoming increasingly data-centric. In 2001, US-CERT (United States Computer Emergency Readiness Team) recognized 52,000 online computer security attacks; this increase in attacks was a 150% growth over the previous year [1]. Moving data online allows greater freedom of information, inherently increasing the ease of access to the information. These are desirable in general, but with intended access, there are created unintended avenues of access which allow those whom are not allowed to read/modify/delete the information to do just that. Preventing the unlawful and unintended viewing, recording, modification, or even removal of data is of paramount importance, particularly when these assets are becoming increasingly the backbone of global economies, and military assets.

The cost of security breaches is truly great, with the cost of identity theft from data intrusions being in the billions in 2003 according to the Federal Trade Commission [3]. In order to protect these increasingly exposed assets, administrators must work to remove avenues of unlawful access to this data, in order to preserve its content. Discovering these methods of accessing the data unlawfully can be hard to detect, and are often built into the very systems that they use, albeit mostly on accident. Something so minor as some computer software which was not written as securely as it should have been can be the culprit for exposing data. Alternatively, misconfigured computers or network hardware can be the issue. These machines can have network access ports which allow malicious code, or intruders to infiltrate and access the machine in question, remotely. Data is then compromised through a variety of means.

The security of computer systems is increasingly important, as data becomes more important in the global economy, and as more data is made available. The availability of the data very often correlates directly with the level of vulnerability the data is subjected to. Once data is out and available, keeping it in the hands of only those that are supposed to have it is a great challenge.

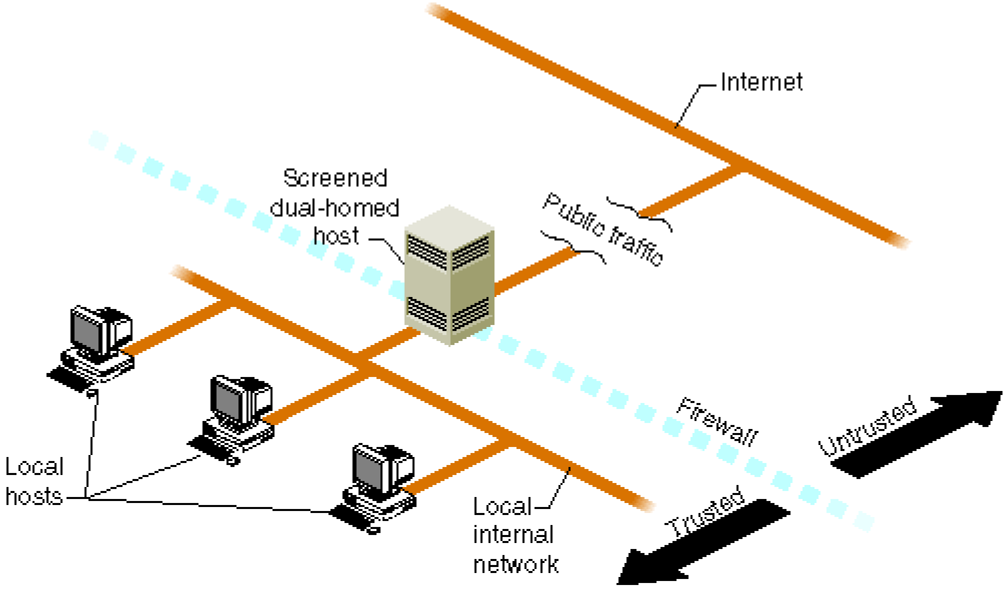
2.1 Computer Security

Computer security starts with understanding the types of vulnerabilities computers are subjected to. Computer security, for personal computers, is concerned more often than not with eliminating methods of exterior access to your machine by other computers which intend to access your machine. Things such as open communication ports, misconfigured security measures from the machine’s operating system, or even improperly secured sharing settings, can leave a computer wide open to attackers to get data or otherwise compromise your machine. Many factors play together to make a machine vulnerable to attackers, but with proper maintenance and vulnerability counter-measure, most security flaws can be counteracted, and sealed up to prevent any exploitation by attackers.

Securing a personal computer by keeping software updated with the latest version is one way to protect machines. Software vulnerabilities are constantly being discovered by security firms and software companies, which then enable the companies that produce the software to fix the problem. Fixing the problem removes the vulnerability, effectively eliminating the possibility of exploiting the computer via that avenue. Once the problem is fixed, a new patch or version of the software is created, which allows users to update the software on their computers, sealing this vulnerability. Without the patches, machines are left vulnerable to security vulnerabilities, waiting to be exploited. Also, with software constantly being updated with new functionality and features, the number of software vulnerabilities each machine possesses increases. A quantitative study of computer software security vulnerabilities shows that as operating systems and other software becomes more complex, the number of vulnerabilities will increase at a predictable rate [8]. With a significant rate of vulnerabilities being created all the time, continual patching of these vulnerabilities is necessary to close them off. The longer these vulnerabilities are left open, of course the greater chance of any individual machine being exploited via these vulnerabilities. Additionally, as new vulnerabilities are discovered, they become more and more commonplace as tools of the trade for criminal agents to exploit one's computers. Very common, powerful vulnerabilities which have been known for years can still be a threat, if one does not update their software after they install it. For example, a 20-year old vulnerability known colloquially as the "ping of death," was a Microsoft Windows vulnerability which allowed attackers to cause a buffer overflow, allowing the attacker to crash or gain access to the machine, depending on the content of the ping used in the attack. This was also exploited by some websites, which contained code which would automatically execute a ping of death attack on clients connecting to the site, which allowed websites to take control of personal computer machines very easily through gaining remote access to the machine, and saving the credentials for attackers who maintained the web server [4]. The vulnerability still affects all modern operating systems without applying the fix for it, and has affected them for years.

Personal computer software firewalls can protect vulnerable computers as well. This type of firewall differs from a network firewall device, in that it is a software firewall running on the machine itself, rather than filtering network traffic in-line. A firewall can be a dedicated network device, or a machine on the network, filtering traffic, or even just a piece of software on the individual machine, running in the background to protect it. A firewall is an application which scans incoming network traffic, and does not allow some of the traffic to pass. This in effect serves as a "fire wall" which keeps known bad traffic out, and allows only traffic which passes inspection by the firewall to pass. Firewalls help to stop exploitation of machines on the network, or on the machine running a software firewall, even before they begin. For example, the famous Slammer worm which is commonly regarded as the fastest spreading computer worm, could exploit a machine with just one packet sent to UDP port 1434, exploiting SQL Server's security vulnerability. Blocking traffic from all but trusted hosts that need to access SQL's management functions was critical in preventing any further spread of the worm [7]. A common application of a firewall would be blocking traffic on a particular computer network port. Traffic to a port which is known to be exploited commonly, can be blocked entirely from entering the network, effectively eliminating the security vulnerability as an access avenue. Firewalls on personal machines can also be fine-tuned to the user's needs. Not only can the network be filtered from the outside networks attempting to access it, but also the user can filter even the traffic which comes through to the user'sown specifications. Each port can be blocked in turn on your machine only, blocking traffic through a port entirely, or just for a specific application, etc., allowing your personal computer to be protected from internal network threats. An example for firewall is shown in Figure 2.2 [35].

Figure 2.2: Firewall Example



2.2 Network Security

Securing a computer network is somewhat similar to securing a personal computer, in that the desire is to eliminate access by unauthorized or otherwise undesirable agents. Access to a network is a two-part endeavor between two machines. A communication session typically is established between the two machines, which enable them to communicate, exchanging messages over some communication medium. Communication over most networks functions this way, and due to this, a major concern is removing the possibility of undesired communication channels being established between machines on the network, or accessing a machine on a network from the outside.

Preventing these communications from occurring is the grand challenge for enforcing network security. Besides security on personal computers and other machines on the network, network administrators must attempt to stop content before it reaches clients on the network. The idea of many network security measure or technique is that network security is the first line of defense, eliminating as many threats as possible before communications reach clients' machines, which are effectively the last line of defense against malicious agents communicating across the network.

One effective method of increasing the security in a network is to not allow malicious communication into the network at all, effectively stopping it at the "front door" of a network. Via network hardware which is in-line between the network being secured, and the wider network outside, such as the Internet, security can be enhanced. Network security devices, including network filters, spam boxes, firewalls, etc., are all intended to filter out content entering the network, which is not allowed based on the security policy of the network. Security administrators and network administrators can configure these devices to block traffic selectively based on a set of rules which detail the “fingerprint” of the malicious or suspicious content as it attempts to be relayed into the network. These rules are based on such things as the content of the data, format, whether it’s encrypted or not, the communication port used, transmission protocol, point of origin, point of destination, etc. These rules, which are complicated to create, and can be difficult to “fine-tune” to being just-right, are the basis for the performance of most network security hardware.

The rules on these machines must be tuned to a level which is “just right.” This means that the rule prevents malicious activity on the network by blocking malicious traffic trying to enter the network, but also the rules must not prevent legitimate traffic from entering the network. If this occurs, the security appliances are preventing proper utilization of the network, which is undesirable, to say the least. Intuitively, a network perfectly secured from the outside world would be one in which no traffic is allowed in. This type of network would not have any malicious traffic incoming from the outside, but obviously it has issues. A network must allow traffic, or else it is useless, and one might as well just unplug from the wider networks. On the other hand, a perfectly usable network would be one in which no traffic is prevented from entering the network. This allows the users the best freedom, since all traffic is allowed, and thereby no legitimate traffic is prevented from arriving at its destination. This type of network is a user’s utopia in concept, particularly if they are accustomed to strict network security rules, but as with the other paradigm, the security administrator’s utopia, this paradigm does not work in practice, since the entire network is exposed to all forms of attacks. Striking the balance between these worlds is the challenge. Quantitative analysis of firewall configurations has found that as the complexity and quantity of the rules increase, the number of configuration errors there are, so getting them right, but not too complex is difficult [6]. Ensuring access to the network, while preventing malicious access is a fine balance, and a challenge which occupies network administrators.

2.3 Impact of Security

Computer security is increasingly important with the move towards digital assets. As more and more companies are valued based on information they possess in digital form, or provision of digital services, the pressure to maintain security on these digital assets has increased. A security breach allowing access to digital information which is of a critical nature at a digital company can cost the company large amounts of money in lost market edge. Additionally, markets are very sensitive to security breaches more and more, as they receive widespread news coverage anymore. Even if a breach is found to have not revealed large amounts of information critical to the company, the damage may have already been done, in the form of market or trader panic, dropping the share price of publicly traded companies’ stocks [2].

Besides the damage done from revealing confidential information which might be of strategic or intellectual value to the companies in question, many companies must protect their customers’ information as well. With the surge in online sales, with large online dealers dealing in huge numbers of customers, customers’ personal information, and even financial information is at risk if the company is exploited. Personal information such as names, addresses, other demographical information, social security numbers, can be stolen, leading to the information being used to impersonate the person in financial transactions, leading to ruined credit, fraudulent purchases, etc., all based on the identity profiles which were stolen from a company which did not properly protect their customers' data.

For example, Sony Computer Entertainment, maker of the popular computer gaming console series, the PlayStation, had their online gaming and entertainment service hacked in 2011. The attackers had access to, and presumably saved, 77 million users' credit card data, used to subscribe to the service, or buy things through it. Despite Sony's claim that the data was encrypted on their servers, and protected, it was stored un-encrypted, and easily accessible through some of the system's vulnerabilities. The result was that the popular console's online service was offline for weeks, and much confidence was lost in Sony's console and their security when handling customers' sensitive data. Sony stated that the cost of the intrusion was approximately $171M [5].

Stolen information is one large source of damage to businesses and organizations with networked data servers, but also one must be concerned with interruption of service. For retailers, especially those online, the more hours which your storefront is open the more sales you receive. Keeping online businesses accessible at all hours, on all days, all year long is of critical importance. Even a few minutes of downtime can cost large amounts of money from lost sales, not only from direct missed sales when customers try to access your site, and it is down, but also through second-hand missed sales from customers' who lose confidence in your storefront since it has been "brought down" by attackers in the past. Keeping outside attackers from clogging your web servers with illegitimate communications requests, false users, or even clogging the network which feeds into your personal network, is of paramount importance. Being able to recognize the difference between a high traffic load and an attack is also an important skill for administrators, in that the behavior can often be very similar.

Overall, security administrators must be able to keep confidential information secret, and out of attackers' hands. Damage can not only be dealt to companies from lost technical or secret planning information, but also through loss of confidence, uncertainty about what information was lost, or even loss or revealing of customers' information to attackers whom will very possibly use the information in identity theft style crimes. The continued protection of this confidential information, while keeping it available to those who need to access it across the network, is of utmost importance for network and security administrators.

2.4 Security Auditing Software

Tools for scanning computers and computer networks for security risks exist fairly commonly today. Many network administrators, and especially computer security experts, must be able to get an idea of the condition of the network in order to remedy problems. Network scanning tools can provide detailed reports of the network conditions, vulnerabilities present, and sometimes what needs to be done for vulnerabilities based on certain databases which store the vulnerabilities and solutions. These reports list the vulnerabilities which appear, and which host they appear on. The tools described below are some of the most popular tools currently developed, and represent a cross-section of the vulnerability scanning software field.

2.4.1 Metasploit

Metasploit is an open-source security vulnerability scanner. Metasploit is the most popular scanning tool according to a recent SecTools survey [18]. Metasploit is designed to enhance network security penetration testing audits. Metasploit was developed to produce an open-source tool for determining network security condition via vulnerability detection and analysis. Metasploit maintains its own database of vulnerabilities detected and exploited successfully. Metasploit is also popular due to it being a widely-encompassing project, including a vulnerability scanner, and its sub-project, the Metasploit Framework, which is a popular framework for discovering new exploits, writing the code to exploit them, and then deploying the exploit. Newly discovered exploits are usually added to the Metasploit database, enhancing the utility of the database, since so many current professionals contribute to the database, and keep it up to date with the latest exploits being developed [17].

Metasploit detects vulnerabilities through simple signature matching of vulnerabilities to port scan results. Scanning results turns out ports which are vulnerable, and through the matching of these to the vulnerability list, signatures can be matched, and the vulnerability can be detected. Once vulnerabilities are detected, the resulting report details which vulnerabilities are present, in a large listing. The resulting scan can then be used to determine which machines to test with the exploiting framework, to see whether the vulnerability has been patched, even though the communication avenue is still open [19]. The resulting report delineates based upon host, and also contains detailed information about the host. Each host’s vulnerabilities are listed, with information about the specific identification of the vulnerability and when it was detected also listed [20].

2.4.2 Nessus

Nessus is a proprietary scanner, developed by Tenable Network Security, and is a cross-platform signature-based scanner utilizing plugins as the basis for determining which vulnerabilities are present [13]. Nessus was originally an open-source project, which has now become monetized and enterprise-level software in many respects. The classic Nessus 2 engine and its predecessors are open-source via an open-source license allowing reproduction but not sale. The release was forked with the creation of Nessus 3, whereupon all new content will be sold, and is privately licensed, though plugin updates for Nessus 2 are still being released, allowing it to still be up to date in terms of which vulnerabilities are known, albeit the older Nessus 2 engine [14].

Nessus scans the network first via a port-scan tool, designed to pick up which ports are open for communication to the device. Nessus has four different scanners available, and though some configuration, can alternatively use other existing scanners that are available on the internet [15] [16]. Nessus then utilizes exploits to attempt to exploit the vulnerabilities detected, in order to determine if the device really is vulnerable to the vulnerability. Nessus uses its proprietary NASL (Nessus Attack Script Language) to run these vulnerability checks. The definition of what attacks can be attempted, and what vulnerabilities are known, are updated on a weekly basis through what are known as "plugins." Once a scan is run, and vulnerabilities are detected, it provides a large report of what was detected in a choice of a number of formats including plain text, XML, HTML, and more. The vulnerabilities are listed which were detected, and can then be searched or filtered in order to determine the network condition more finely.

2.4.3 Nmap

Nmap is a security scanner which attempts to map out a network and discover the clients and applications running on the network, in order to understand the network's condition. In addition to this scanning, Nmap can do port scanning on clients, enabling a security analysis of the clients much like other scanners. Nmap can determine details about the hosts such as type of device, and presence of firewalls. Many other security tools use Nmap as their base in order to get the network condition [21]. Nmap is open source and cross-platform as well, enabling easy expansion and adaption of the tool to fit the needs of the user.

Nmap also has adaptive scanning, enabling what many tools do not; Nmap can scan a network much more successfully than other tools through careful scanning. Nmap has the ability to scan with attention paid to network latency, congestion, and even the target being resistant to the scans, in order to get a better scan on the network's condition [23]. This scanning also enables Nmap to not be detected and stopped by other network security automated devices, which enhances Nmap's effectiveness as compared to other simple scanning tools [22]. Nmap utilizes vulnerability databases as other tools do in order to match the open ports detected with known vulnerabilities, and stands out as the most directly useful network scanner for many projects. Nmap serves as the basis for many more complicated tools, in that its simple nature, and the structure of its implementation allows easy integration.

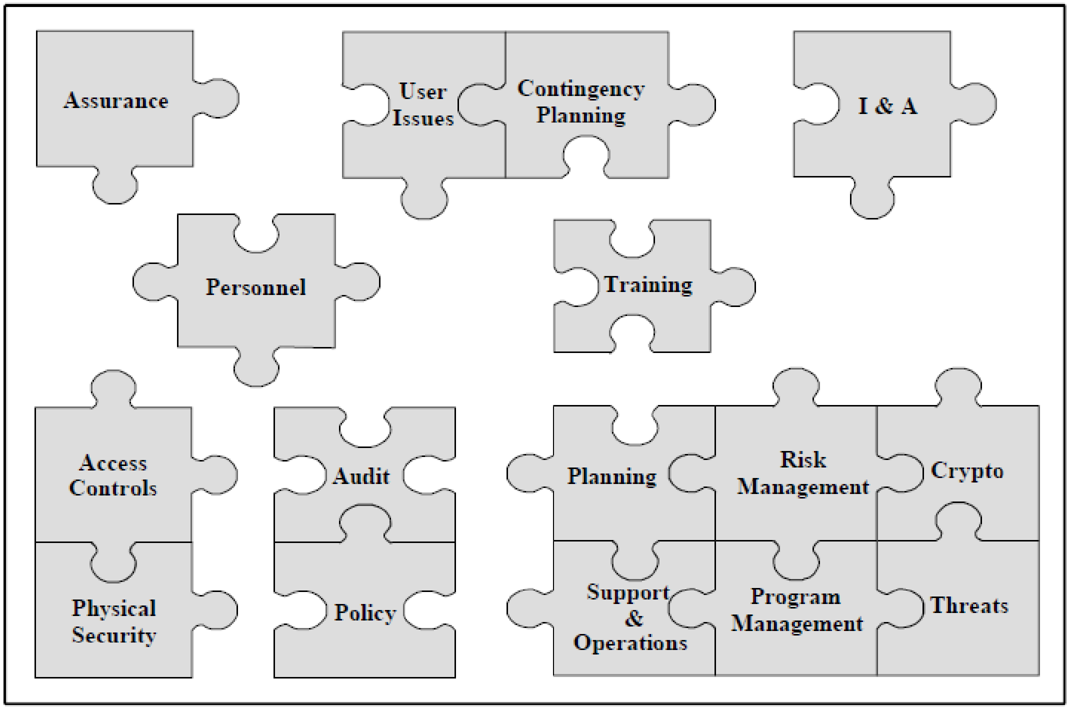
2.4.4 Audit Software Issues

All the audit software programs that have been described have common goals, and common issues, relating to the goals of this thesis's research. The reports generated are more complex and detailed than can be easily understood in many cases, since the volume of data created is very large. Filtering or otherwise data mining the report is required in order to understand just what is going on in the network. Additionally, just the reports are given, which does not necessarily give a score for how severe the issues are on the system(s) in question. The goals of this thesis are to confront this issue, and produce a score which simplifies these reports, providing a uniquely simple network health metric.

CHAPTER 3: NIST'S COMPUTER SECURITY ASSETS

With the ever increasing demand for computer data security, and the rising risk to national assets associated with national data of critical importance being more at risk, the United States Government funded initiatives to create a national security asset (as shown in Figure 3.1 [38]) capable of increasing the security on computer networks [24]. This security asset is concerned with increasing the level of security on government machines, overall hardening government computers and their integral data from outside attackers. Common security issues are catalogued, evaluated, and solutions generated for resolving the vulnerability and returning the machine to a more secure state. A strong degree of certification and professionalism in the identification and verification of the vulnerabilities and fixes within these databases was of paramount importance, in order to guarantee the usefulness of the database. By scanning a network, matching the network condition to the database and identifying network security issues this way, the research in this thesis intends to rate a computer network's security health via nationally accredited security metrics.

Figure 3.1: NIST Security Assets



3.1 Common Vulnerabilities and Exploits – CVE

The Common Vulnerabilities and Exploits Database (CVE) is a concerted effort on the part of the MITRE Corporation, a non-profit organization managing national defense and research facilities, foundations, and projects [26]. CVE works to combine publicly known common vulnerabilities into one database, uniting the many commercially maintained, and publically contributed security vulnerability databases [25]. This central database allows each of the vulnerabilities to have one unique identifier, a CVE id, such as “CVE-2001-1723.” The use of unique identifiers reduces the complexity of the international security threat identification effort, in that there are fewer duplicate vulnerabilities circulating, enabling a cleaner, simpler network report to be generated. CVE serves as more of a dictionary of vulnerabilities than a database, with each of the vulnerabilities listed and defined, but not explained to any great detail, nor solutions suggested, as in more advanced databases. CVE is a great and unique tool for centralizing vulnerability identification, and serves as a common-language for different security data sources and organizations.

3.2 Common Vulnerability Scoring System - CVSS

The Common Vulnerability Scoring System (CVSS) is an open framework for providing a repeatable quantitative score for computer security vulnerabilities. The degree to which a security risk is presented based upon how severe the security vulnerability is, is reflected in the CVSS score [27]. Each of the vulnerabilities can be given a CVSS score through a review process in order to evaluate it by security professionals, and the vulnerability's exploitations that have occurred in the past. In order to compare how severe each type of security vulnerability is to each other, we must evaluate them using a quantitative evaluation of them, such as CVSS provides. CVSS is unique as a scoring system, in terms of how reputable it is, and how wide-reaching its implementation has been, across many NIST supported security vulnerability assets.

CVSS utilizes a multi-faceted approach to scoring vulnerabilities. The final CVSS score is in fact a combination of the exploitability metrics and the impact metrics. The combination of these subscores yields the CVSS base score, which is utilized widely, and is independent of the situation, organization, network, and other variables in which the vulnerability may be present. The exploitability metrics are concerned with how the attacker will be able to access the machine with the vulnerability and how to exploit it. The complexity of the attack, the level of access needed to invoke the attack, and how deep in the network the system exists (how deep the attacker would have to delve from the network's entry point to exploit it), are all concerns. The impact metric is concerned with the CIA of the system. Is the vulnerability going to affect the confidentiality, integrity, or availability of the data, or a composite of this? In order to include considerations about the specific situation where the vulnerability has manifest itself, CVSS allows use of the environmental and temporal sub scores, which increases the accuracy of the score, provided the network situation is properly known, in order to provide this information. The environmental subscore considers how much of an impact this vulnerability would have upon the organization, and how many systems on the network are vulnerable on the network. Additionally, modifications to the impact sub-score are used, considering how much confidentiality, integrity, and accessibility is a concern. The temporal sub-score enables inclusion of situational data about the vulnerability in the wild per-se. Information about how long the vulnerability has been available openly (like released on the internet, and well known, for example), the types of fixes available to resolve the issue, and the level of validation that has been done to make sure the vulnerability exists, and is exploitable. Inclusion of this extra information makes the scores customized to your network, and they no longer apply to the worlds' situation any longer. Only the base-score is network and situational independent, allowing a wider level of utilization [27].

3.3 National Vulnerability Database – NVD

The National Vulnerability Database (NVD) is one of NIST’s important security assets for determining the severity of computer security risks. NVD is the sum of many other security databases, and utilizes the CVSS scoring system, allowing the fullest utilization of available public computer security risk analysis, and quantification methods via CVSS scores [28]. NVD is also linked with CVE, enabling comparison and expansion of NVD with CVE entries. Expanding CVE entries to include references for where the vulnerability was found, how it might be fixed, and much more, is the role NVD plays on expanding security databases, rather than just being a superset of other databases and dictionaries. NVD is also part of NIST’s Information Security Automation Program (ISAP), which is a move towards enabling computer controlled security appliances and software to increase computer security through automatic resolution of existing and newly discovered vulnerabilities [29]. The CVSS scores from NVD, and identified vulnerability signatures in NVD entries allows for this automated approach. NVD is used as the primary resource for finding vulnerabilities and determining their comparative severity and impact. Using NVD’s information about the vulnerabilities, vulnerability signatures can be derived, enabling matching of network conditions to the extracted signatures, then matching to CVE IDs, and getting the CVSS base score from the NVD entries, scores can be acquired for each of the vulnerabilities which has been identified from the matching process. NVD provides a reputable, widely used, constantly updated, and openly available resource for basing this research upon.

3.4 NIST Security Asset Conclusions

While NIST’s security assets that have been described are useful to many researchers, they are more of a foundation to further research than a security appliance in their own. The databases and dictionaries enable security professionals and developers of security tools to consolidate the many different definitions of vulnerabilities, and get an idea of how severe given vulnerabilities are, but that is as far as these resources takes you. Many security appliances, like those mentioned in sections 2.4.1-3 go so far as to tie discovered vulnerabilities to their CVE entry numbers or pull the vulnerability information from NVD, but they do not attempt to profile the entire network situation as a single security environment. These tools profile each of the vulnerabilities separately, providing information relevant to the vulnerability, the threats present that may exploit it, and its status as exploitable, but provide no groundwork for profiling the entire network. The other security auditing tools we have looked at may list the severity of each of the vulnerabilities, but this research expands upon this by compositing multiple vulnerabilities’ scores into a machine’s total score, and further along as an entire network’s complete score. Compositing these scores into combined scores builds upon NIST’s security foundation that they have laid with these tools, and expands the functionality of these resources though providing another application for them, considering the entire machine, or the entire network, providing a larger security condition.

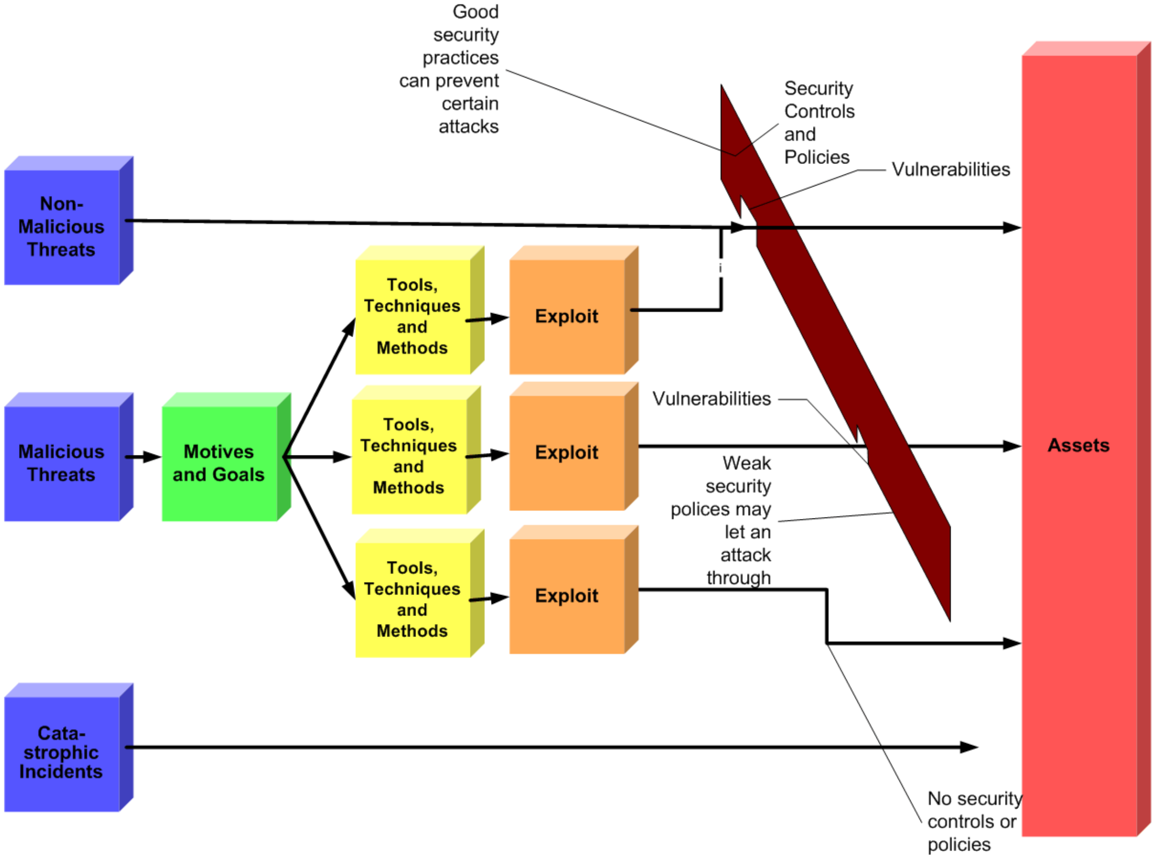
CHAPTER 4: QUANTIFYING NETWORK SECURITY

The new work of this thesis is concerned with defining the security “health” of a network, and generating a new metric for creating a composite, quantitative score which represents the health of a computer network’s security. The new system implemented in this thesis will provide a double value between 0 and 10 inclusive, in order to keep in line with NIST’s CVSS scoring system, to reflect the level of security or insecurity of a given network. A network of computers will be scanned with a port-scanning tool such as Nmap, determining which ports are open on machines in the network. Based upon the ports open, and matching to the NVD entries’ information about what situation describes a given single vulnerability, the certain security vulnerabilities which exist in a network can be determined. The security vulnerabilities detected are security exploits that may occur is a malicious agent exploits the vulnerability, meaning that the situation could be ripe for them to exploit the given machine, based on the condition detected on the network. Taking the list of vulnerabilities that the machines have, scores are retrieved from their respective NVD entries, and the scores are used as the basis for quantifying the network. The scores for each of the vulnerabilities are combined into a score for the given computer, and then the scores for all the computers are in turn combined into a single score for the entire network. This compositing is studied as the primary focus of the thesis, in that combining a network security report into a single quantified value, whilst still maintaining an accurate reflection of the network’s security situation is challenging. Three methods for this composition are proposed, demonstrated and compared, with the latter of the three being the result of experimentation and optimization of the compositing technique. The resulting score will be from 0-10 inclusive, in-line with CVSS scores, with the most vulnerable networks (the least healthy) holding a score of 10, and the most secure networks (the most healthy), with a score of 0 to 3.9 being high health, 4.0-6.9 being medium health, and 7.0-10.0 being low health.

4.1 Approach

The approach for this thesis work is to find a way to simplify network security reports to a point where they are more accessible, and more easily digestible by more users. The complexity and length of the current reports is too long to make them directly useful. It makes filtering, data mining, or some other method for extracting information from the reports necessary in order to utilize them effectively. The reasoning behind this is that if the data remains hidden, or is not understandable by those in a position to do something about increasing security, the data may well never exist. Increasing the viability of the data has the effect of increasing security in this situation, since the hidden data becomes available. Gathering a network situation via network scans, then working the data via this thesis's method garners a simpler representation of the network situation, enabling quicker, easier, and wider understanding of the network situation. A typical network vulnerability architecture is shown in Figure 4.1[37].

Figure 4.1: Network Vulnerability Architecture


Good security practices can prevent certain attacks
Security Controls and Policies
Vulnerabilities
Non-Malicious Threats
Tools, Techniques and Methods
Exploit
Vulnerabilities
Malicious Threats
Motives and Goals
Tools, Techniques and Methods
Exploit
Assets
Weak security polices may let an attack through
Tools, Techniques and Methods
Exploit
Cata-strophic Incidents
No security controls or policies

The thesis works as follows: first a network situation exists. This situation can be understood as a typical TCP switched network of many user machines connected to switches which are connected together behind some form of router which joins this local network to the wider network outside, usually the Internet. Typically a scan of security by a security administrator will be concerned with their internal network, the network behind this router. Utilizing a tool like Nmap, the research can scan the network to pick up which ports are open to communication on the machines within the network. Once the ports are detected, we know which communication ports may be used to exploit the machines on the network. In order to know just how the machines might be compromised, we consult with NVD in order to pull out known port vulnerabilities. The vulnerabilities which are identified as being port vulnerabilities are retrieved from NVD, and matched against the open ports on the machines within the target network. If a match is found, this means that the computer with the match may be exploitable via this vulnerability. The vulnerability is logged, and the rest of the network is checked over for vulnerabilities as well, until all possible matches are exhausted. Once the matches are found, the CVSS scores for each of the vulnerabilities are found, and are used in combining the scores into a composite score.

4.2 Compositing Methods

In order to simplify the scores obtained from NVD for the vulnerabilities which the system has detected on the network machines, the scores must be combined in some way to get a final value. A method has been devised to get the composite score for all the vulnerabilities on each machine, which works as follows:

Equation 1: Vulnerability Compositing Method

$$V(v) \rightarrow CVSS\ Base\ Score\ for\ Given\ Vulnerability$$

$$S(v) = 1 - {}^{V(v)}/_{10}$$

$$S(v_1, v_2, \ldots, v_n) = \prod_{i=1}^{n} S(v_i)$$

$$H(v_1, v_2, \ldots, v_n) = 10(1 - S(v_1, v_2, \ldots, v_n)) \tag{1}$$

The method for combining the vulnerabilities into one score for the machine is found by first taking each of the vulnerabilities in turn, and getting their CVSS base scores. The CVSS scores range from [0, 10] with the higher the worse security, we get this via the V(*v*) function. Next, the security function (*S(v)*) is applied to the score, rendering a score which is [0, 1], with the higher the better security. This number is utilized through generating the product of all these security scores from all the vulnerabilities, generating one composite number from [0,1], with the higher the better. This score is then converted back with the health function (*H(v)*) generating the final machine score, which matches the CVSS scoring method of [0-10] with 10 being least secure.

For compositing the machine scores into a final score for the entire network, many methods were tested, in order to get some comparative values, and determine the best way to composite these scores without losing information about the network security condition. The more accurate the final score is, the better, so having a final score that reflects very accurately the security situation in the network is the primary goal. To this end, the best of these compositing methods were experimented with, and provide some perspective on the efficacy of the last solution described, which appears to be the most accurate.

4.2.1 Linear Compositing

Linear compositing of the scores is the most straight forward approach, so this was tried first to get a baseline for the research. Obtaining the scores from NVD for each of the vulnerabilities, the scores were combined, giving a composite score for the entire network after each machine was composited via the vulnerability composting method from Equation 1. The compositing method for the linear compositing method is to take the sum of the client scores in the network we generated before, and divide by the number of clients, giving the mathematical mean. This straight forward approach treats all computers equally. The compositing is computed as follows: for clients C from $\{C_1$ to $C_n\}$ we have:

Equation 2: Linear Compositing Method

$$C(v) \rightarrow Client\ Vulnerability\ Composite\ Score\ from\ Equation\ 1$$

$$SUM(C_1, C_2, \dots, C_n) = \sum_{i=1}^{n} C_i$$

$$LC(C_1, C_2, \dots, C_n) = \frac{SUM(C_1, C_2, \dots, C_n)}{n} \qquad (2)$$

Issues with this approach are as follows. Firstly, the approach is perhaps too equal with its treatment of the clients. With many clients of the same composite scores, the network may get a score close to that of all the vulnerabilities. For example, if the machine had ten or more vulnerabilities which were all 2.5, the final score for the network would be 2.5. This does not truthfully reflect the security level on the network, in that the network, which has a large numberof vulnerable machines, is really more vulnerable than 2.5 , in that there are more ways to exploit the network than were the network to have just a single vulnerable client with a CVSS score of 2.5. This bit of information is lost when this linear compositing is done on these situations. Additionally, this same problem can perhaps hide severe issues. Were a network to have a number of weakly vulnerable clients, and a single critically severe security risk client, the severely vulnerable client may be obscured somewhat in the final score, since the score will be driven down from the level of the one vulnerability.With a network that contains a client score of 8.5 and other clients with lower scores, the net score will be less than 8.5 because the average will be brought down by the lower scores.The resulting score from the method can end up lower if a network has many low risk vulnerabilities than if a network does not, since the high score will not be reduced. This can cause issues with the accuracy of the score in situations where the numbers of vulnerable machines are not roughly equal. In light of these issues, linear compositing is not the best method to get an accurate picture of the network security condition.

4.2.2 Weighted Non-Role Based Compositing

The weighted non-role based compositing method is the second method which is proposed and tested. This method is similar to the vulnerability compositing method utilized on each machine. This method confronts the problem of less important vulnerabilities of lesser scoring severity than the more severe scores causing the score to be drawn down. This scoring method allows each machine to add the collective scoring, without reduction based on a score being lower. This method allows for the severity of the security vulnerability level on the network to increase as the quantity of the vulnerabilities in relation to the size of the network to increase. The product of the client scores is generated, resulting in a more accurate image of what the network situation is. The composite is generated as follows:

Equation 3: Weighted Non-Role Based Compositing Method

$$C(c) \rightarrow Composite\ Vulnerabability\ Score\ for\ Client\ by\ Equation\ 1$$

$$S(C) = 1 - {}^{C(c)}/_{10}$$

$$S(c_1, c_2, \dots, c_n) = \prod_{i=1}^{n} S(c_i)$$

$$H(c_1, c_2, \dots, c_n) = 10(1 - S(c_1, c_2, \dots, c_n)) \qquad (3)$$

This method addresses the issues encountered with the linear compositing method by taking into account the quantity the vulnerable clients on the network. This is confronted in the last compositing method. This method does take into account that more important vulnerabilities affect the composite score more, which is an important improvement over the linear compositing method, in that the scores are not reduced too much by less severely vulnerable clients. A score generated by this method represents the network situation more accurately, but it can be tweaked in order to generate even better results, which is what the final method, the role-based method demonstrates.

4.2.3 Weighted Role-Based Compositing Method

The weighted role-based combination is conducted via giving the members of the network different weightings based on how critical they are on the network. For example, if a machine has the access rights to the other machines on the network, or serve some sort of administrative role which allows the machine to have special privileges over the other machines, the machine will be given a higher weighting than other, less critical machines. This increased weight of the score on this machine allows the machine to be given a greater impact on the total score for the machine and the network at each level of the compositing process. The score can be adapted based on how critical the machine is, and allows for greater control of the network setup information. With this additional information about which machines are critical in the network, the situation is more accurately represented than when this information is absent. This weighting is a means to enhance the level of information in the network situation. This compositing is conducted as follows:

Equation 4: Weighted Role-Based Compositing Method

$C(c) \rightarrow Composite\ Vulnerabability\ Score\ for\ Client\ by\ Equation\ 1$

$P(c) \rightarrow$ *Importance Rating for Client (Exponent for Client's Score)*

$$S(C) = 1 - {}^{C(c)}/_{10}$$

$$S(c_1, c_2, \ldots, c_n) = \prod_{i=1}^{n} S(c_i)^{P(c_i)}$$

$$H(c_1, c_2, \ldots, c_n) = 10(1 - S(c_1, c_2, \ldots, c_n)) \qquad (4)$$

This compositing is similar to the weighted non-role based compositing method, save for the exponent applied to the client score. This exponent effectively increases the influence that the particular client with a higher importance score (higher exponent) has on the network's score overall. More important clients on the network will have a higher weight, as appropriate to the importance of the client, with scores from 1-5 being the tested paradigm. Ultra critical members are assigned an importance exponent of at most 5, with this being an extreme case, where even moderate vulnerabilities on this client may have extreme effects on the final score. Most clients in typical networks are assigned the standard importance of 1, which does not affect their score contribution. This weighting can be fine-tuned by administrators in order to get a better feel for the network situation, and can be updated to increase the accuracy of the network quantification. This method provides the best results in the experimental results shown in Chapter 6.

CHAPTER 5: SYSTEM IMPLEMENTATION DETAILS

Within this section of the thesis, discussion of how the algorithms and framework for quantifying the network were implemented will be provided. General approaches for the problems encountered, as well as specifics about the implementation details will be discussed, though code will not. Please refer to the code listings for code-level implementation details. The system was implemented in JAVA, utilizing JDK 1.7, and only Sun JAVA native libraries (no extraneous expansion libraries).

5.1 Framework Overview

The framework for the implementation of the algorithms is as follows: Several classes were developed to represent the network situation and to quantify the network's security situation. The problems of creating this system were overcome in several steps, with each being integrated into a central framework. The framework is a collection of classes which compartmentalize the functionality of the algorithm and enable the code to be very modular. The collection of classes is called together through static method calls to enable the full functionality for testing. Integration with NVD and the quantification algorithms themselves are called together through the framework of classes.

5.2 Inputs and Format

The input for the quantification is in several pieces. The first piece is a file which lists all the NVD database shards which are to be used by the system to find port vulnerabilities to detect. This file is a simple one entry per line text file which is picked up by the NVD Parser class, which extracts data though data mining techniques from these large NVD files. NVD makes the entries for NVD available through a set of large XML files, with their own special tag set, with one file being released each year, and a running-total file of the new vulnerabilities for the partial calendar year in progress (the *modified* entry in the set). The NVD shards can be downloaded directly from their website [30]. The entries, within their XML tags, allow easy extraction through data mining techniques. Each of these NVD shards is also to be made available to the software on the local machine.

The next input for the system is the network scan. The scan, such as Nmap generates, must be included in the system's format for the system to process. The scan itself contains the information about the network condition, as scanned by the network port scanner. The open communication ports on the client machines on the network are listed, as well as identified by the IP address of the client. The system also supports weighted combinations, so the respective importance factor of the clients in the network are also part of the input file, and can be edited to increase the accuracy of the report, if necessary. This scan is formatted in such a way that the software understands the entries. Scans from various software platforms for network scanning can be adapted to the required format through adaption classes in the platform, and more adaptors can be written to expand the tool's functionality.

5.3 Processing Inputs

Processing of the input files conducted as described here. The processing of the NVD source files, which are in XML format is done by first optimizing the file for our mining. Initially the files are formatted without whitespace, and have many tags to separate the sections. In order to easy data mining, the tags brackets (“<” and “>”) are removed and replaced with spaces, effectively space-delimiting the file for mining. The now space-delimited file is then mined for each entry, and each entry is identified whether it is port vulnerability or not. The vulnerabilities which are not are discarded, whilst the ones which are port vulnerabilities, are stored in the output file, which ultimately is the input for the NVD Matching class. The entries are mined through detection of the start tag for each entry. Each entry has a CVE identifier number, which is stored to uniquely identify each of the vulnerabilities in the database, and also is marked with a unique tag in the file. The combination is detected, and the ID is stored. Next the CVSS base score is extracted through detection of the CVSS scoring section, and extraction via regular formatting of the entry. The score is stored for later use in scoring the machines and networks based on these CVSS scores. The description of the vulnerability is extracted, and mined itself, with the section being found via its summary tag. The section is mined for mentions of specific ports being used in the vulnerability, in that NVD lists port vulnerabilities with their exploited ports in their summary descriptions. If a port is found in this section, the entry is kept, and is deemed port vulnerability, and stored in the output file. After all of the entries are processed, the input processing is finished for the NVD source files. The NVD source is currently just over

48000 CVE entries [31], and is processed in a few minutes on an average performance, modern consumer laptop.

5.4 Matching to NVD Entries

Determining which vulnerabilities are present on the network is done by matching up the condition of the machines on the network to the NVD entries describing the conditions required for a vulnerability to be present. As described in section 5.1.2, the system compares each machine’s condition to each vulnerability in the NVD entries used in this thesis, in order to determine which vulnerabilities are present.. The entries in this subset database have a number of communication ports associated with them, which must be open on the target machines in order to exploit the vulnerability on the machine. The system takes the network situation input file, which lists clients on the network via the unique identifier of their IP address, and the open ports on their machines, and matches these open ports to the ports required for the vulnerability to perhaps be present on the target machine. The ports which must be open for the vulnerability to be exploited are called the “vulnerability signature.” We search for a match between the signatures of each of the vulnerabilities on each machine in the network in turn, discovering which are present on the network. Once the vulnerabilities are detected, each client object in the system has a collection of unique vulnerabilities present on that machine, which in turn contains information about which CVE entry is detected, the CVSS base score for this vulnerability, and ancillary information such as the NVD description for the vulnerability. With this information discovered through this process of signature matching, the system is ready to score the machines and collective network.

5.5 Compositing and Scores Generation

Compositing of the CVSS scores for each of the vulnerabilities is conducted in several different ways in order to research the efficacy of different approaches, and ultimately was done to create the final role-based compositing method. Compositing via linear composition, weighted combination and the role-based compositing method were all implemented in order to develop an effective scoring quantification on a variety of systems in the experimental data later in this thesis. The details for the compositing methods are discussed below.

CHAPTER 6: EXPERIMENTAL PARADIGM

The experimental paradigm explored in this thesis is that of confirming the performance of the quantification algorithms via matching the scores to known vulnerable and so-called "patched" systems. In certain well-documented situations, vulnerabilities have been studied to a point where the setup required for the vulnerability to be exploited very successfully is known, and situations which fix this vulnerability are also known. Through exploration of a few such well documented examples, validation of the algorithms' scores can take place, and be discussed. Comparison of scores from all of the compositing methods will be conducted in order to give comparative results based on each approach.

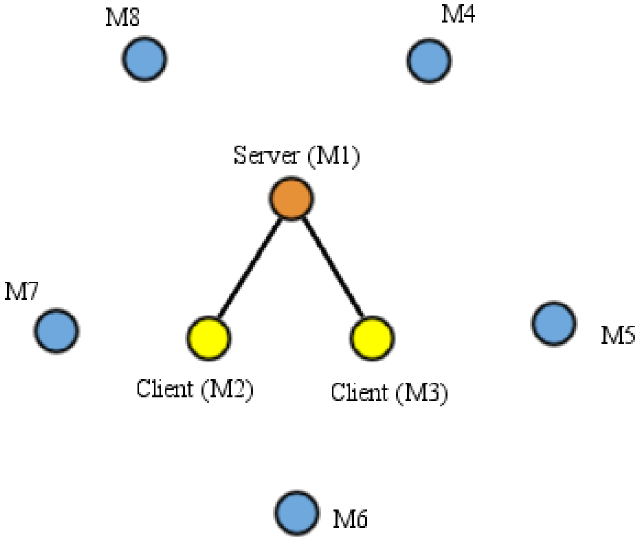
6.1 Scenario One - Windows Systems Vulnerability

The first scenario which is considered is in the case of Windows operating system vulnerability, allowing remote attackers to assault TCP port 135 with malformed packets. This entry is listed under CVE-2006-3880 [32]. The topology for this experimental scenario is shown in Figure 6.1. These malformed packets are sent constantly (as fast as possible), and have random integers inserted into TCP headers, enabling the vulnerability on the machines, as the machine attempts to process the headers. This vulnerability, exploited on a network of computers running the Windows operating system can be potentially isolated from the outside world through this exploit. The exploit causes a denial-of-service on the client via an IP stack hang, making communication with the affected machine not possible. All windows machines which have port 135 open, and have not been patched or upgraded to avoid this vulnerability are vulnerable to this attack. In effect, this vulnerability affects all vulnerable Windows machines equally, with no special relationship between the affected machines being created, though a critical service machine affected with this attack may stop serving its clients due to being unreachable, causing wide reaching issues, such as if the machine were a login-serving system. Were a serving machine exploited with this vulnerability, more machines would be affected, in that they would not be served their services by the machine since it is effectively cut off of the network. For these tables indicating the network present, a dash indicates no vulnerability present (such as the system only having one or zero vulnerabilities).

Table 6.1: Experimental Scenario 1 Details

Client Name	Client Role	Importance Level	Vulnerability 1	Vulnerability 2
M1	Server	3	5.0	1.2
M2	Client	2	5.0	2.1
M3	Client	2	5.0	-
M4	Non-Client	1	2.1	-
M5	Non-Client	1	1.2	-
M6	Non-Client	1	1.2	-
M7	Non-Client	1	-	-
M8	Non-Client	1	-	-

Figure 6.1: Experimental Scenario 1 Topology



The experimental setup for this experiment has several machines on the network with the vulnerability, and other machines which are not vulnerable. The scores are derived based upon applying the quantification framework that was written for this research on the model network, and comparing the security health scores resulting from the experiments. The situation involving the vulnerability, and then without the vulnerability are tested, for comparison as well, in that the network should be more vulnerable with the vulnerability than without it. The experimental results for this setup are shown in Table 6.1.

Linear compositing on the above situation yields a score of 1.962500 for the network, in that the scores are considered separately, with no correlation. The scores are reduced a bit due to a quarter of the network machines being completely secure. The average scores for the machines on the network is the resulting score, which reflects a generally secure network, which is the reality, in that the worst vulnerability is 5.0 and the quantity of vulnerabilities is relatively low.

Weighted compositing yields a score of 4.581250, since the information about which machines are especially critical due to their client server role, and being vulnerable to the attack through the server being reflected in the weighting. The client is ranked as a high importance member, with clients being medium importance, and the unrelated network members being low importance. The score reflects several clients having a 5.0 vulnerability, but still there being a few clients with very few to no vulnerabilities reduces the score a bit from the 5.0 of the experimental vulnerability.

The role-based compositing method yields a score of 7.056850, which more appropriately reflects the network condition. The quantity of higher severity vulnerabilities tips the score upwards. The larger vulnerabilities tend to help dominate the scoring, and as their quantity increases in relation to the total number of network members, they push the score upwards. The score reflects not only the importance of the machines on the network, but also increases the score above the base score of each machine, which reflects more realistically that the number of vulnerabilities on the network, the more vulnerable it is. The other compositing methods do not really reflect this. This method more accurately reflects the condition of the network with a score higher than the base of 5.0 from the experimental vulnerability.

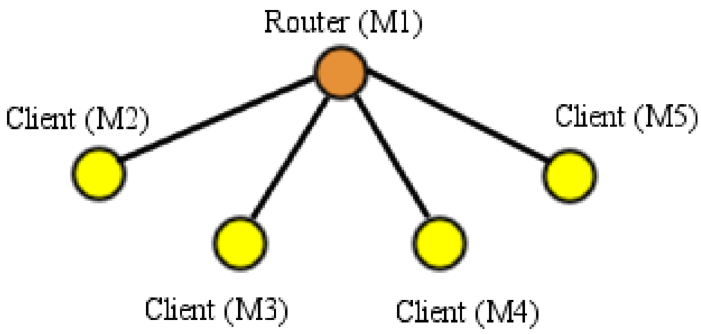
6.2 Scenario Two – Router Vulnerability

This experiment reflects CVE-2002-2159, which is a vulnerability on some Linksys (Cisco) routers, which allows remote backdoor access to administration and router control [33]. This vulnerability allows remote attackers to effectively control or shut down network traffic via any controls that the router has available. Perhaps most severe is that the attacking agent may move clients to a “DMZ” role, meaning they are no longer behind any sort of firewall, traffic control, etc., that the router might otherwise provide. This exposes the selected client(s) to remote access from the wider network on the outside of the router’s served network. Attackers might also gain critical information about the clients through the router configurations, and the DHCP table, listing all clients served on the network. This silent exploitation of the network allows backdoor access until it is removed, which means a silent backdoor for agents exploiting this vulnerability, which may not go noticed for a very long time. The router itself is of critical importance on the network, and the clients served by the router are of medium vulnerability, in that they stand to be potentially security probed by the outside network, or the attackers manipulating the network via the router’s control. These clients may have local protection such as application firewalls or antivirus applications which can protect them from this exposure, but clients depending on the router for security will be completely exposed by a savvy attacker exploiting this vulnerability. For this experiment we will consider a network behind such a firewall, which exposes its clients to this potential threat. The topology for this experimental scenario is shown in Figure 6.2. The experimental setup for this scenario is as follows:

Table 6.2: Experimental Scenario 2 Details

Client Name	Client Role	Importance Level	Vulnerability 1	Vulnerability 2
M1	Router	3	8.0	-
M2	Client	2	4.3	-
M3	Client	2	1.2	2.1
M4	Client	2	4.3	-
M5	Client	2	5.0	-

Figure 6.2: Experimental Scenario 2 Topology



This setup demonstrates a situation where all clients on the network are potentially at risk due to a member of critical importance, with power over the other clients being compromised. The importance of the router as a security asset, and the critical nature of not allowing the router to be compromised, lends the importance of the router to be much higher than that of the clients. Though the router is of high importance through the vulnerability itself, the clients served by the router are also potentially at risk, raising their vulnerability level as well. This situation is characteristic of a highly vulnerable client-server interaction where the exploited server may compromise the clients via information garnering or service control (such as traffic manipulation, or denial of service). This situation is also important, in that the vulnerability is a quiet backdoor which may be manipulated for long periods of time without being detected, in that the backdoor is built into the code for the router's software. The experimental results for this setup are shown in Table 6.2.

Linear compositing gleans a score of 5.419800, weighted non-role based compositing a score of 9.871254, and weighted role-based compositing a score of

9.994886. Compositing with the linear approach yields the average, which does not fully reflect the situation on the network, in that this score is less than the score of some of the vulnerabilities. The score loses its accuracy through averaging everything, and disregarding the frequency of appearance of the vulnerabilities so more vulnerabilities may actually yield a smaller score than a single vulnerability. This is the case here, where the total score across the machines is brought down from the 8.0 present on the router, due to the lower scoring machines which are less vulnerable. This reflects the problem with the linear compositing method. The weighted score is much more reasonable, in that it recognizes the importance of certain members of the network over the others. The router being so critical in this situation is recognized, and the severity of the problem, in that it affects all members of the network potentially is recognized as well, pushing scores up even more. The score is quite high via this realization, reflecting the severity of the situation, and fixing this vulnerability being such a driving concern in order to secure the network, which is at this point in the scenario, extremely vulnerable to this one vulnerability. The role-based compositing method provides a score a bit higher than the weighted compositing method, which reflects the quantity of scores present in the network, which the weighted score does not. The number of vulnerabilities on the machines raises the score a bit, since the more vulnerabilities, the more vulnerable the network is, since there are more points of entry, and more ways to exploit the machines on the network, via more vulnerabilities to exploit. These scores reflect a very vulnerable network, since very high severity vulnerability is located on the most critical machine in the network, and all clients on the network are served by the vulnerable machine, making them vulnerable in turn. This is a very insecure situation, and the score reflects this.

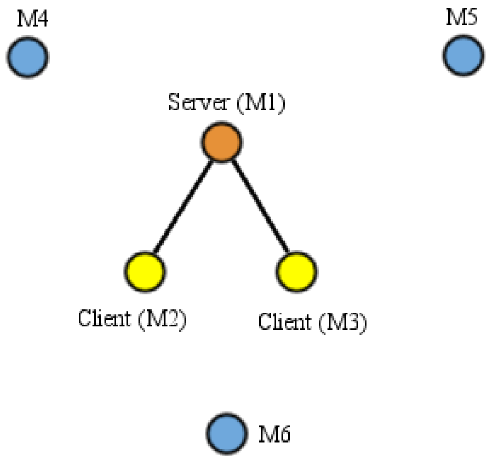
6.3 Scenario 3 – SQL Server Vulnerability

This vulnerability, which is CVE-2002-0649, is an exploitable vulnerability on SQL servers, allowing data manipulation on the databases being served [34]. This vulnerability can either cause denial of service on the server, or execution of inserted code on the server. The inserted code can do arbitrary things on the server, such as change rights, gain access to the machine, modify data stored in the databases, etc., making anyone served by these databases potentially at risk. Networks with SQL servers and clients are vulnerable to this. The machines which are clients are vulnerable, but quite so as the SQL server itself, and the non-clients present in the network, are for all intents not made more vulnerable via this vulnerability being exploited. Due to these roles, the SQL servers will be given a high importance, the clients a medium importance, and the non-clients a low importance to reflect this. The topology for this experimental scenario is shown in Figure 6.3. The experimental setup is as follows:

Table 6.3: Experimental Scenario 3 Details

Client Name	Client Role	Importance Level	Vulnerability 1	Vulnerability 2
M1	Server	3	7.5	-
M2	Client	2	4.3	-
M3	Client	2	1.2	2.1
M4	Non-Client	1	4.3	-
M5	Non-Client	1	5.0	-
M6	Non-Client	1	5.0	-

Figure 6.3: Experimental Scenario 3 Topology



The experimental setup shows a SQL server with the vulnerability discussed present on it. The machines which are listed as clients are the clients of this SQL server, which are somewhat vulnerable due to the exploit possible on the SQL server serving them. The machines which are not clients are not served, and are also not servers, making them independent members on the network, and though their vulnerabilities are a concern, they are not as vulnerable as the other machines, and do not contribute to the large-scale exploit which is possible over the nodes involved in the SQL group. The experimental results for this setup are shown in Table 6.3.

The linear compositing method provides a score of 5.2665, which is the average of the machines on the network. The score, as with the other scenarios, does not reflect the full situation on the network, in that it is blind to the importance of certain members over others, and cares not that higher level vulnerabilities on machines makes the network more vulnerable, but instead only shows that the average score on the network is less than 5 for the given setup. The weighted compositing method yields a score of

9.919534, which shows that the high importance of the critical member of the SQL server is shown as highly important, and brings the score very close to the maximum score, especially with the added scores of the medium importance clients to the SQL service. The role-based compositing method yields a score just higher than the weighted score, with a score of 9.963324. This score reflects that more vulnerable machines mean a worse score, which escapes the scoring systems of the other techniques. This score is more reflective of the severity of the situation, and again approaches the maximum score of 10 very closely, in that the situation on the network is very severely vulnerable.

6.4 Real World Data – WKU's Client Network

The techniques described were also applied to Western Kentucky University's (WKU's) user client network, in order to test the system on a real network. Nmap was utilized to scan the network and determine which ports were open on the client machines, and then the scan was processed with the processing framework code. The scan showed 11762 online clients, with a total of 262 vulnerabilities detected. WKU's network is configured as a small group of routers, all connected to one another through redundant links, and then from these routers, a structure of switches serves the clients. This switch structure is a tree-shape with a main distribution switch being the first step from the router, and the switches for each building or area being served by the distribution switch. This building-wide switch then serves many switches within the building, which in turn serve the clients. This structure can be called a small star-graph of routers with trees of switches 3 or 4 layers deep extending from the router network.The worst of these was an 8.0 vulnerability score. No machine was detected with more than one vulnerability, so the maximal client score was an 8.0. The composite network score garnered was a 5.061171 via linear compositing, an 8.02572 via weighted non-role based compositing, and an 8.57161 via weighted role-based compositing, noting a few machines which are known administrator machines on the network as medium priority. These scores seem to accurately reflect the health of the network, with a machine score of 8.0 present, and many clients with some vulnerability present on it. With more information about critical machines a more accurate score might be possible.

Table 6.4: Experimental Scores Table

Experiment	Linear Compositing	Weighted Non-Role Based Compositing	Weighted Role-Based Compositing
Scenario 1	1.9625	4.58125	7.05685
Scenario 2	5.4198	9.871254	9.994886
Scenario 3	5.2665	9.19534	9.963324
WKU Data	5.061171	8.02572	8.57161

CHAPTER 7: CONCLUSIONS AND FUTURE WORK

In this section of the thesis, discussion of the efficacy of the methods employed to quantify the network and achieve the goals the thesis set out to achieve are conducted. The various methods employed to achieve this goal of reducing the complexity of the network health report is used, and the approach is discussed as well. Future work which goes beyond this thesis is mentioned, in that more work can be done regarding this goal and extending its functionality.

7.1 Efficacy of Methods of Composition

The methods utilized to composite the scores for the individual machines on the network are discussed here in order to discuss what has been determined about shortcomings or strengths in the methods utilized. These methods each have their own instinctive reasoning behind creating them, and under analysis perform differently. The performance experienced under experimentation, and the observed behavior under different situations is discussed.

7.1.1 Linear Compositing

Linear compositing is the most straight-forward of the approaches, and the simplest form of composition experimented with in this thesis. The average gives a very intuitive value for the health of the network, in that the total health of the network perhaps appears to be the average health of all its members. This approach does not take into account any of the extra information which is utilized in the more role-based compositing methods. The end result of linear compositing is an average score, which intuitively seems to be a rational scoring method, but in practice breaks down quickly. The main issue with this method is that it simply does not reflect the security level on the network accurately, in that the more vulnerable clients present, the more severe the security threat to the network. With this method, the more vulnerable clients, potentially the more secure the network, if the averaging process is processing many low-severity threat clients, and one severe level threat. This would erode the high score garnered by the severe case, and result in a lower score for the entire network, which does not represent the situation accurately, since the more vulnerable clients, the more severe the network security threat. The average security level of each host is what the linear composite gives, which is not precisely what is wanted in this case, in that more information is known about the network situation, and this information will garner a better idea of the network health situation than this simple method.

7.1.2 Weighted Compositing

The weighted compositing method allows for a more accurate picture to be drawn of the network, via each client contributing to the final score. This is effective, in that it allows the clients to each contribute to the final score without drawing down the final score with lower composite scores from more secure clients. This method more reflects the reality that the more vulnerable machines the more insecure the network is, even if the machines outnumber the more insecure individual clients. This method greatly increases the accuracy of the scoring for the system, with only the role-based compositing providing a more accurate indication of network health.

7.1.3 Weighted Role-Based Compositing

Weighted role-based compositing allows the compositing method to take in account more information known about the network situation, such as if a router or server, or its clients are vulnerable due to a particularly vulnerable client, and such situations as special application on particular machines controlling administrator access for example, can be modeled. Through control of the weights of the members of the network, precise control can be applied to the members of the network, allowing fine-control of the network scenario on which the network quantification will be applied. Information about the network scenario known by network administrators can be used to adapt the model to more accurately reflect the network situation. This allows the method to be much more accurate in determining the severity of the network situation than the linear approach, raising the severity considerably whenever a situation which is made worse via certain members being of higher importance occurs. The method brings out much more information and a much more accurate score than the linear approach. This method can be made as accurate as the information about the importance of the individual members is understood and this information input into the system. Even with a mild understanding of the most critical network members, the compositing method works better than any other method tested.

7.2 Efficacy of Approach

The approach of trying to reduce the complexity of network reports into a single double value is an interesting question. The approach attempts to go nearly as far as one can go in terms of simplifying the value. A decimal value is much more efficient to understand, interpret, and otherwise handle than a large network report, but inevitably, some data will be lost about the specifics of the network situation. Through compositing down to such a level quantification is achieved though, which enables mathematical analysis of the network health, as opposed to reports which are merely a collection of data not so easily interpreted. This attempt at interpreting the information down to a single health score enables the ease of analysis and handling, but perhaps is not as detailed a reporting factor than other methods of network condition analysis, but the system cannot be asked to be. Ease of interpretation and simplification are the strong suits of this system, and were the goals that drove the research, so in that respect this system achieves its goals. Full analysis of the data can occur at the report generation level, but for large scale analysis where data complexity reduction, or quantification is desired, this system achieves what is not achieved via other systems.

7.3 Future Work

Additional work is possible with this system in order to enhance the accuracy of the system. Work relating to expanding the method for interpretation of the vulnerabilities to incorporate things such as expanded definitions for vulnerability signatures would enhance the detection of vulnerabilities. This work would enhance the network situation identification via expanding the ability of the program to detect NVD and CVE vulnerabilities through matching other things such as detection of machine operating system, probing vulnerabilities, and other advanced vulnerability detection techniques utilized by larger scale network scanners. Any way to increase the accuracy of the network scenario which is interpreted for the quantification would enhance the accuracy of the score generated by the system.

In addition to increasing the accuracy of the scan to bring in more of the vulnerabilities present in the network, the system may be improved via more analysis of the interaction of the elements in the network. The current system does not take into account the situation of the network fully as it might were things like attack graphs considered for determining exactly how vulnerable certain machines are in consideration with other machines which are not so vulnerable due to the attack graph being longer or more indirect. This type of analysis would add another layer of accuracy to the value, enhancing the result.

BIBLIOGRAPHY

[1] A. Garg, J. Curtis, and H. Halper, “Quantifying the financial impact of IT security breaches”. *Information Management and Computer Security*, 11(2):74–83, 2003.

[2] Kevin M. Gatzlaff, Kathleen A. McCullough, “The Effect of Data Breaches on Shareholder Wealth”, *Risk Management and Security Review*, 13(1): 61-83, 2010.

[3] Federal Trade Commission, “Identity theft survey report”, URL: *http: www.ftc.gov/os/2003/09/synovatereport.pdf*, 2003.

[4] Microsoft Tech Center, “Microsoft Tech Bulletin MS11-057-Critical”, URL: *http://technet.microsoft.com/en-us/security/bulletin/ms11-057*, 2011.

[5] PCMAG, “PlayStation Hack to Cost Sony $171M; Quake Costs Far Higher”, URL: *http://www.pcmag.com/article2/0,2817,2385790,00.asp#fbid=w2U0yFjxgos*, 2011.

[6] A.Wool, “A Quantitative Study of Firewall Configuration Errors”, *IEEE Computer*, 37(6): 62–67, 2004.

[7] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, and Nicholas Weaver. “Inside the slammer worm”, *IEEE Security and Privacy 1*, 4, 2003.

[8] O. H. Alhazmi and Y.K. Malaiya, “Quantitative Vulnerability Assessment of Systems Software,” *Proceedings of the Annual Reliability and Maintainability Symposium 2005* pp. 615 – 620, 2005.

[9] B. W. Lampson. “Computer Security in the Real World”, *Proceedings of the Annual Computer Security Applications Conference*, 2000.

[10] Loomis, G.A., J.S. Ries, et al., “If electronic medical records are so great, why aren’t family physicians using them?,” *J. Fam. Pract.* 51(7), 636-41, 2002.

[11] G. Sivathanu, C. P. Wright, and E. Zadok. “Ensuring data integrity in storage: Techniques and applications”, *In Proceedings of the ACM Workshop on Storage Security and Survivability (StorageSS '05)*, pages 26-36, 2005.

[12] Hoo, K.J.S. “How much is enough? A risk management approach to computer security”, *Ph.D. Dissertation*, Stanford University, 2000.

[13] Tenable Network Security, “Product Overview,” URL: *http://www.nessus.org/products/nessus/nessus-product-overview*, 2011.

[14] Lemay, Renai (CNET), “Nessus security tool closes its source,” URL: *http://news.cnet.com/Nessus-security-tool-closes-its-source/2100-7344_3-5890093.html*, 2005.

[15] Renaud Deraison (Tenable Network Security), “Nessus 4.4.1 Released,” URL: *https://discussions.nessus.org/message/9584,* 2011.

[16] SecTools, “Top 100 Network Security Tools,” URL: *http://sectools.org/,* 2006.

[17] Rapid7, “Metasploit Community,” URL: *http://www.rapid7.com/products/metasploit-community.jsp,* 2011.

[18] SecTools, “Top 3 Vulnerability Exploitation Tools,” URL: *http://sectools.org/sploits.html,* 2006.

[19] Kelly Jackson Higgins, “Metasploit gets new vulnerability scanning features,” URL: *http://www.darkreading.com/vulnerability-management/167901026/security/attacks-breaches/222000147/metasploit-gets-new-vulnerabilty-scanning-features.html,* 2009.

[20] Noobz Network, “Metasploit Pro Review,” URL: *http://www.n00bz.net/metasploit-pro/,* 2011.

[21] Ethical Hacking, “Integrate Nmap with Nessus,” URL: *http://www.ehacking.net/2011/06/integrate-nmap-with-nessus-tutorial.html,* 2011.

[22] Andrew Bennieston, “NMAP – A Stealth Port Scanner,” URL: *http://nmap.org/bennieston-tutorial/,* 2006.

[23] Fyodor, “The Art of Scanning,” URL: *http://www.phrack.org/issues.html?issue=51&id=11#article,* 1997.

[24] NIST, “NIST NVD FAQ,” URL: *http://nvd.nist.gov/faq.cfm,* 2011.

[25] CVE, “About CVE,” URL: *http://cve.mitre.org/about/,* 2011.

[26] MITRE, “About MITRE,” URL: *http://www.mitre.org/about/index.html,* 2011.

[27] NIST, “NVD CVSS Support v2,” URL: *http://nvd.nist.gov/cvss.cfm,* 2011.

[28] NVD, “NVD about Page,” URL: *http://nvd.nist.gov/about.cfm,* 2011.

[29] ISAP, “SCAP Homepage,” URL*: http://scap.nist.gov/,* 2011.

[30] NVD, “NVD Datafeed and Product Integration,” URL: *http://nvd.nist.gov/download.cfm,* 2011.

[31] NIST, “NVD Homepage,” URL: *http://nvd.nist.gov,* November 2, 2011.

[32] MITRE CVE, *“*CVE-2006-3880,” URL: *http://cve.mitre.org/cgibin/cvename.cgi?name=CVE-2006-3880,* 2006.

[33] MITRE CVE, “CVE-2002-2159,” URL: *http://cve.mitre.org/cgibin/cvename.cgi?name=CVE-2002-2159,*2002.

[34] MITRE CVE, “CVE-2002-0649,” URL: *http://cve.mitre.org/cgibin/cvename.cgi?name=CVE-2002-0649,* 2002.

[35] Tech Pubs, “Firewall Diagram,” URL: http://techpubs.sgi.com/library/dynaweb_docs/0630/SGI_Admin/books/IA_BakSecAcc/sgi_html/figures/firewall_env.gif, 2011.

[36] Technet, “CIA Triad Figure,” URL: http://blogs.technet.com/blogfiles/seanearp/WindowsLiveWriter/LayersDefenseinDepthPart1_B11E/CIA_triad.png, 2011.

[37] Microsoft Technet, “Security Diagram,” URL: http://technet.microsoft.com/en-us/library/Cc751212.sgfg0202_big(l=en-us).gif, 2011.

[38] NIST, “Security Pieces,” URL: http://www.pdffinder.com/get/an-introduction-to-computer-security-the-nist-handbook.pdf, 1995.