

EFFECTIVENESS OF NETWORK SECURITY  
FRAMEWORKS ACROSS SKILL LEVELS

by

MARY KARNES

B.S., University of Colorado, 2000

A thesis submitted to the  
Faculty of the Graduate School of the  
University of Colorado in partial fulfillment  
of the requirement for the degree of  
Masters of Science  
Interdisciplinary Telecommunications Department  
2003

This thesis entitled

Effectiveness of Network Security Frameworks Across Skill Levels

written by Mary Karnes

has been approved for the Interdisciplinary

Department of Telecommunications

---

Dr. Douglas Sicker

---

Dr. Tom Lookabaugh

---

Dr. Scott Savage

Date\_\_\_\_\_

The final copy of this thesis has been examined by the  
signators, and we find that both the content and form  
meet acceptable presentation standards of scholarly work in  
the above mentioned discipline.

Karnes, Mary (M.S., Telecommunications)

Effectiveness of Network Security Frameworks Across Skill Levels

Thesis directed by Assistant Professor Douglas Sicker

The objective of this study was to understand the effectiveness of network security frameworks across skill levels with the intention of understanding whether more extensive use of frameworks should be pursued and if so, for which skill levels are they most useful. Through the combination of a qualitative hands on experiment of network security professionals, a quantitative analysis of questionnaires and a study of actual pertinent threats to a corporate network, this study explored what skill levels are most helped by existing industry frameworks in order to understand the current and potential effectiveness of these frameworks.

Data collected during an applied experiment of security professionals showed the specific framework used to be unhelpful to network security analysts. This is surprising data that contradicts the original hypothesis that frameworks are helpful to network security analysts. The author cannot find an experimental flaw that strongly suggests the negative result is false. However, special cases need to be considered. Firstly, the experiment was limited in resources handicapping the ability of the data to fully answer the question. Secondly, it is possible that the framework tested was not a proper choice for the given experiment. Lastly there was a very interesting observation suggesting that the method for ranking skill level was flawed. These special cases give valuable information for subsequent research. Future research is recommended with the next round of testing incorporating a new skill level analysis a more abundant resource pool of analysts to test with.

## DEDICATION

I dedicate this to my mom. 😊

## ACKNOWLEDGEMENTS

There are so many people who contributed to my success. This would not have been possible without the steadfast encouragement and wisdom of an amazing advisor, Professor Doug Sicker. There is one other professor who was highly instrumental in providing academic advice, Professor Tom Lookabaugh.

Outside of academia, many co-workers contributed ideas, support and time into this thesis. These people include: Carey Bunn, Howard Bergstrom, Rhonda Hicks, Lacey Bostick, David Mackey, Michael Walter, Dr. Josh Lackey, Brooke Anderson, Lisa Molinelli and Lisa Freedman. David Mackey reminded me that I should write about something I know. Once I started writing about something I knew, Lacey Bostick reminded me that I need to solve a problem and helped show me how. Carey Bunn remained a faithful mentor, friend and supporter. Howard Bergstrom, Dr. Lackey and Brooke Anderson added thought, critique and advice. Michael Walter threw in enthusiasm and creativity into my design. Lisa Molinelli braved the challenge of editing my writing and amazingly, she succeeded. Lisa Freedman ran nearly one hundred miles brainstorming with me. There were also roughly forty network security analysts who spent their personal time completing my experiment to help me find a solution to the problem. And finally, Rhonda Hicks made it happen when it was about to fall apart. You are an amazing team of co-workers and I thank you.

I would like to thank my family and friends for enduring me during this time, you guys are awesome. I would also like to give praise to the Father. He is also awesome.

## CONTENTS

Preface .....	1
Chapter 1 – Introduction .....	2
Objective of the Study .....	4
Scope of the Study .....	7
Limitations and Assumptions .....	9
Definition of Terms.....	12
Chapter 2 – Literature Review .....	17
Network Security Frameworks.....	17
History.....	17
Trends .....	24
Experimental Psychology .....	26
Chapter 3 – Materials and Method .....	30
Quantitative .....	30
Qualitative .....	39
IDS Study .....	39
Chapter 4 – Results .....	41
Quantitative .....	41
Qualitative .....	53
IDS Study .....	56
Chapter 5 – Discussion of Results .....	60
Quantitative .....	60
Qualitative .....	64
IDS Analysis .....	66
Overall Trends .....	65
Chapter 6 – Conclusions .....	69
Future Research .....	70
Bibliography .....	71
Appendices	
A. Pre Experiment Survey (Skills Survey).....	73
B. Post Experiment Survey (Framework Utility) .....	75
C. Firewall Test Data: Firewall Rules .....	76
D. Firewall Test Data: Network Diagram .....	77
E. Firewall Test Data: Framework .....	78
F. Firewall Test Data: Directions and Answer Sheet .....	79
G. Firewall Test Data: Directions and Answer Sheet .....	80
H. Firewall Test Data: Comprehensive Finding Set.....	81
I. Vulnerability Test Data: Nessus Scan Output .....	83
J. Vulnerability Test Data: Other Scan Output .....	96
K. Vulnerability Test Data: Framework .....	101
L. Vulnerability Test Data: Directions and Answer Sheet.....	102
M. Vulnerability Test Data: Directions and Answer Sheet.....	103
N. Vulnerability Test Data: Comprehensive Finding Set.....	104
O. IDS Signature Cross Reference Comments .....	105

## TABLES

4.1 Descriptive Statistics of Skill Levels Within the Population.....	42
4.2 Actual Findings Across Skill Levels: Vulnerability.....	47
4.3 Actual Findings Across Skill Levels: Firewall.....	48
4.4 Average Findings: Vulnerability and Firewall.....	48
4.5 Top 50 IDS Attacks Signatures for September and October of 2003.....	56
4.6 IDS Signatures Cross Referenced with Open Source Framework.....	57
5.1: % of Findings Found By Skill Level.....	66

## FIGURES

3.1 Formula for Finding Skills Levels.....	38
4.1 Histogram of Security Experience.....	42
4.2 Histogram of Computer Experience.....	43
4.3 Graph of Skill Levels.....	44
4.4 Graph of Vulnerability Specific Skill Levels.....	44
4.5 Graph of Firewall Specific Skill Levels.....	45
4.6 Skill Level Graph of Participants in Firewall Experiment.....	46
4.7 Skill Level Graph of Participants in Vulnerability Experiment.....	46
4.8 A Graphed Comparison of Vulnerability Experiment Findings.....	48
4.9 A Graphed Comparison of Firewall Experiment Findings.....	49
4.10 Scatter Diagram of Vulnerability Findings; Across Skill Levels and Frameworks.....	50
4.11 Scatter Diagram of Firewall Findings; Across Skill Levels and Frameworks.....	51
4.12: Skill Level Versus Findings – Firewall Experiment .....	52
4.13: Skill Level Versus Findings – Vulnerability Experiment .....	52
4.14: How Many Participants Used the Framework Given.....	53
4.15: How Many Participants Understood the Framework Given.....	53
4.16: How Many Participants Had Used A Framework Before.....	54
4.17: How Valuable Framework Was to Participants During Experiment.....	54
4.18: How Many Participants Would Have Liked More Time.....	55
4.19: How Many Participants Believe A Framework Could Be Useful.....	55
4.20: Total Number of Signatures Reference in Framework.....	59
5.1 A Graphed Comparison of Vulnerability Experiment Findings.....	61
5.2 A Graphed Comparison of Firewall Experiment Findings.....	61

5.3: IDS Signatures Covered By Framework..... 68

## PREFACE

Before an introduction to the topic is presented, a brief summary of findings will be discussed to aid the reader in understanding the context and impact of the research. This study tested the effectiveness of network security frameworks across skill levels through quantitative and qualitative analysis. Data collected during an applied experiment of security professionals showed the specific framework used to be unhelpful to network security analysts. This is surprising data that contradicts the original hypothesis that frameworks are helpful to network security analysts. The author can find no experimental flaw that strongly suggests the negative result is false. However, special cases need to be considered. Firstly, the experiment was limited in resources handicapping the ability of the data to fully answer the question. Secondly, it is possible that the framework tested was not a proper choice for the given experiment. Lastly there was a very interesting observation showing that upper echelon analyst performance was extremely poor. These analysts generally did worse than the lower level analysts. This suggests that the method for ranking skill level was flawed. These special cases give valuable information for subsequent research. Future research is recommended with the next round of testing incorporating a new skill level analysis a more abundant resource pool of analysts to test with.

Hopefully this sets the stage for the reader to more quickly absorb the content of this paper. And without further ado, please continue to the introduction.

## CHAPTER 1

### INTRODUCTION

In 1965 Gordon Moore published a small four page paper on computing power. His observations, coined "Moore's law," indicated that computing power would exponentially increase over the years to come (Moore). Almost forty years later this law still holds true and in the past four decades, computing technology has flooded and affected almost every area of business, most notably in the form of the Internet. Companies that only ten years ago were wholly paper-based are now fully networked, paperless offices plugged into cyberspace with terms like email and World Wide Web floating ubiquitously through the office. The change comes complete and coupled with new corporate cultures and methods of doing business.

As company business models race towards creating products quickly, the infrastructure supporting design and quality is struggling to keep pace. The outcome of Moore's law has been smaller, faster and cheaper products at the expense of quality and reliability. Similar problems exist in technology infrastructures across many disciplines. Crimes are committed before laws are made, standards are not in place before technology is produced and technology unrelentingly burns forward. The security infrastructure has simply been outpaced. The law and standards bodies frantically try to build infrastructures to contain and manage the downpour. While these initiatives will eventually succeed, the technology sector is currently a volatile area in which to conduct business, especially in the security arena.

Lurking in the water is a cyber security game of cat and mouse, particularly in the software industry. Defective software is created, hackers find creative ways to exploit the software and then companies reactively hasten to fix the problem. This supports and reinforces a cycle of defective software with security flaws. Hackers have become free quality assurance testers for software manufacturers and, as security has yet to be integrated

as a main component of software products, customers have little choice but to continue to use the software.

As companies depend more and more on the Internet, the types of defects in software become more prolific and increasingly damaging. Security threats today can result in a complete halt of business, as was in the case in the distributed denial of service against Amazon in 2000. Even more recently were the MS Blaster and Sobig worms in August of 2003 which Mig2, the digital risk assessment company, states caused economic damages estimated at \$32.8 billion (Gordon). The Sobig virus, one of the most recent worms on record, is also considered the most damaging on record. These examples only encompass a limited set of attacks, they only address denial of service examples. A denial of service worm merely prohibits the company from using the Internet for a given period of time, it does not have a destructive payload meant to corrupt and damage end user data. In a corporate setting, this end user data can be critical intellectual capital. Companies increasingly rely on intellectual capital as a business differentiator and for competitive advantage and the threat of loss or damage to this data looms with increasingly destructive consequences,

Even still, companies simply cannot afford to ignore the seemingly infinite resources the Internet provides. The reliance on computing, computing power, and networking is growing and will continue to grow. Because of the increased threat of destructive consequences, demand for security remains a topic of discussion. Although the security industry does not noticeably appear to be gaining ground on the hackers, a security community is slowly rising. Security standards, methodologies, best practices, and education are beginning to emerge. These groups are not only building temporary life preservers but are developing fundamental controls to tame and hone the resources of the Internet.

One particular area in security that has been growing in the last ten years is in the methodology space. Specifically, security methodologies, otherwise known as frameworks, have been emerging to help combat the specific problem of securing networks so that the flood waters of the Internet can be limited and controlled. These frameworks have never undergone any sort of analysis for utility, usability, or comprehensiveness. This study

explores the issue of network security frameworks and their effectiveness across varying skill levels. Through the combination of a qualitative hands on experiment of network security professionals, a quantitative analysis of questionnaires and a study of actual pertinent threats to a corporate network, this study explores what skill levels are most helped by existing industry frameworks in order to understand the current and potential effectiveness of these frameworks.

Before this paper continues, it is important to define what a framework is. A network security framework is a written document guiding the testing process of analyzing the security posture of an environment or piece of an environment. This framework provides a repeatable, measurable process in which to provide network security.

### **Objective of Study**

The objective of this study is to understand the effectiveness of network security frameworks across skill levels with the intention of understanding whether more extensive use of frameworks should be pursued and if so, for which skill levels are they most useful. Specifically, organizations that rely on providing network security services would benefit tremendously from understanding if a framework could help network security analysts protect networked environments.

### *Caveats*

The experiment used to prove the findings was performed under constraints of time, money and human resources. These restraints limit the data to reflect only the current and potential effectiveness of network security frameworks within a given scope rather than to reflect effectiveness in its entirety. In order to truly understand the answer to the question, this experiment needs to be re-run in a more comprehensive way. Thus the objective of the study has to be narrowed to only answering a question of understanding if network security frameworks reveal an inclination or a potential to more effectively enable analysts to perform

security analysis in the setting of one Fortune 500 company with one specific framework in a time limited environment.

### **Significance of Study**

There are four reasons why the potential effectiveness of security frameworks is interesting to the network security industry:

- The rise of the security service industry
- The increased importance of securing networks
- The growth of the security industry and the lack of skill
- The need for comprehensive analysis in security

#### *The rise of the security service industry*

There is currently a trend of rapid growth in the network security service industry. Industries like managed security services are predicted by Frost and Sullivan to continually grow for the next 10 years to the tune of a compounded annual of growth of 27% (Frost and Sullivan). This indicates that there is and will continue to be a growing interest in security and thereby continued growth of service offerings. If a network security framework were shown to be effective for security analysts, security service providers would have good reason to spend time developing these frameworks as they could ultimately affect the quality of their service and their brand reputation.

#### *The importance of securing network steadily increases*

The number of attackers, threats, and vulnerabilities shows no sign of decreasing in the years to come. In fact, the “ceaseless revelation of vulnerabilities” is predicted to be a

high market driver in the security services industry (Frost and Sullivan.) With an increased reliance on intranets and the Internet to do business, threats to companies will continually increase causing an increased need for network security.

#### *The growth of the industry and the lack of skill*

The need for security appears to be growing faster than the skills necessary to protect networks. One reason for outsourcing security is a lack of qualified security specialists (Frost and Sullivan). An effective framework could help bridge this gap. On one hand, service companies could use such a framework to develop brand reputation and a measurable, quantifiable network security review. On the other hand, if a framework was found to be effective, middle sized and smaller companies would have more success securing their own networks because there would be relief in finding qualified analysts. An effective framework, while clearly unable to create qualified analysts in itself, could prove to increase the ability of lesser skilled analysts to perform security analysis. This would impact both service providers and service recipients alike.

#### *The need for comprehensive analysis in security*

One interesting dynamic in the security space is that an attacker need only find one hole into a network to compromise it, while a security analyst must comprehensively secure an entire environment. A well-designed, methodical, measurable and repeatable approach to network security could go a long way in helping analysts to comprehensively secure a network rather than partially securing it.

While analyst skill will always factor into a security assessment, frameworks enabling a less skilled analyst to perform superior analysis would prove to be valuable and worthy of study and development to both security service providers and recipients alike.

### *Framework measurement currently non-existent*

The reason this subject of study was pursued is that at first glance a network security framework appears to be a very helpful guide to an increasing need for a comprehensive security analysis tool. However, when trying to measure which framework is the best or even trying to distinguish usefulness of individual frameworks, any attempt at measuring is theoretical, biased and impractical. Even if a theoretical measurement tool were found to be adequate, it could not account for effectiveness across skill levels. The evaluation model would have to assume that every analyst was performing at pre-determined skill level which is very unrealistic. The frameworks that will be explored further on in this paper generally involve high level descriptions of tests to be applied to the networked environment, leaving specific practical implementations for the analysts to interpret and execute on their own. The interesting question is if the framework itself stimulates analysts of all skill levels to identify more weaknesses on average or if it is only useful for a limited set of skilled analysts. Many corporations have a vested interest in having mid-range analysts be able to perform security for a couple reasons, including the difficulty in finding skilled analysts and the time and expense in training analysts. None of the frameworks claim to educate the users. They do claim that if a skilled network security analyst uses it, that analyst will have completed a comprehensive review. They assume that a person with the right skills will be working with the framework. However, what happens when this framework is tested with regular corporate network security professionals? Do these frameworks show potential for a more effective analysis? This is one question the author hopes to answer.

### **Scope of the Study**

The specific experiment designed to answer the question of framework effectiveness in this study involved a hands on practical test of one specific, industry accepted framework. The actual framework used was Pete Herzog's *Open Source Testing Methodology Manual*. Specifically 40 network security analysts were asked to attempt to solve a problem (review

firewall rules or review vulnerabilities on a server). Half of the participants were given a framework to use as a tool to help solve the problem and half were instructed to solve the problem without using a framework. Prior to starting the experiment, the participants filled out a questionnaire to firstly understand if their skill level meets the requirements for the experiment (they must at least be familiar with networks and security to be useful to the study) and also to rank their skill level. This questionnaire can be found in Appendix A. Measurement of framework success was examined by the average number of findings per skill level group compared with average findings across skill levels. If a higher average prominently rose for a section of analysts using a framework, it was determined that the framework was useful to that specific group.

The participants in the experiment were network security analysts of some fashion; the skill level ranging from extremely novice to the very elite experts. This experiment was restricted to those within the security industry, meaning those with some, even if very minimal, security experience. The reason is that the experiment would not be helpful if it tested how a tractor operator or a professional chef analyzed security because the frameworks are all written for people who already know about security. The frameworks are not intended to be educational. Therefore this study would only be valid if performed by participants in the network security corporate industry, participants who would on any given day, be asked to perform security analysis on a corporate network. The participants in this thesis experiment included ethical hackers, security architects, firewall analysts, cryptographers, vulnerability scanning professionals, and various other security specialists. In order to participate in the experiment, these users had to pass a test to ensure that all participants were those that would have to, on any given day, perform network security analysis in a corporation. Understandably, the framework will be most effective if utilized by elite security specialists. However in the real world, this framework will be used by many less skilled, but still experienced, analysts. If this type of framework proves helpful to those of lesser skill, companies and analysts have a vested interest in seeing these frameworks pursued and further developed towards fulfilling their security needs.

The scope of this experiment is to test participants in the specific case of trying to find vulnerabilities, misconfigurations or weaknesses in a given server or firewall rule set. This study will not be able to show the effectiveness of the framework in its entirety. However, it will be assumed that if, on average, more vulnerabilities were found by an analyst using a framework, the framework is useful and further study is merited.

## **Limitations and Assumptions**

### *How the objective was obtained*

The study tested potential effectiveness by breaking apart an industry accepted network security framework into pieces and having network security professionals participate in an experiment using these pieces to solve a problem set. The results of the experiment were correlated across skill levels. This experiment design has inherent limitations and assumptions.

### *Breaking apart the framework – a bad idea?*

Some will assert that to break apart a framework to test it will invalidate the ability to judge the effectiveness of the framework. Frameworks by nature are not usually meant to be a step by step process and many components would be revisited during a given network security testing process. Frameworks in pieces are not as useful as frameworks in their entirety. So in one way, this experiment had the potential to jeopardize the data because each security analyst was only be asked to find specific vulnerabilities, misconfigurations and weaknesses in a specific piece of a network (either a server or a firewall rule set.) This means participants were only be asked to find symptoms, not diseases, of poor security in a confined setting. On the one hand, this can be considered strictly bug hunting and not helping to secure the fundamental problems of inadequate corporate policies and or implementation of corporate policies. On the other hand, this experiment was still very useful for two reasons:

1. In order to diagnose diseases, symptoms need to be recognized. This experiment tested how well participants were able to recognize symptoms of poor security. (Which is what each piece of the framework guides the user to do.) One example of why this is important is the situation of application testing which, as an aside, was not used in the experiment due to time restrictions. In application testing, it is important to understand whether an application has been coded securely but if the analyst cannot find an sql-injection bug, even if the framework specifically states, "check application for sql-injection vulnerabilities," then it cannot be relayed back through the application development department to implement a process to check for this. Does this mean the framework failed? Not exactly, it means that the framework was not effective for this class of tester. Perhaps a less skilled analyst should be using a different framework or the framework should be redesigned to accommodate additional skill levels.
  
2. The experiment was designed to be less reliant on the framework as a whole. Each problem given to participants was as wholly contained as possible. For example, vulnerability testing incorporated one Linux server with a vanilla install, that is to say the server had the operating system installed without any changes made to it. This left many vulnerabilities on the system. As such, the participants, acting as analysts, did not need to see or understand server functionality as they knew it is accessible by the open Internet and thus the analysts had full capability of speaking to vulnerabilities on this server without understanding other components of the environment this server sits on. In the same token the firewall test was created to solely focus on firewall rules. From the misconfigurations in the firewall it can be understood that the firewall administrator did not

understand basic security principles. Without seeing the entire network, the analyst can speak to the firewall issues.

In order for an information security specialist to understand the consequences of a poorly implemented network security policy, they must actually physically test the environment for weaknesses. This is the first step in diagnosing the real problem. Thus the experiment was scaled down in that it breaks apart the framework and tests it in pieces within boundaries of time.

#### *Why the experiment was scaled down*

A comprehensive network security framework involves analyzing a network security posture from numerous angles. In this experiment, 40 professional security analysts were asked to participate. Because of participation from professionals, a time limit MUST be strictly maintained. In order to comprehensively test a framework, many weeks would be involved per participant. This is impractical within the bounds of this thesis. It was decided that the experiment must be limited to one hour. Thus the framework must be broken apart. Specifically chosen were two pieces of the frameworks that could be bound by time; vulnerability research and verification and firewall testing. The data from this framework experiment will only show effectiveness of finding holes in these specific areas and will not identify how well the analysts were able to identify diseases, only symptoms. Again, this is still important to understand because in order to diagnose diseases, symptoms need to be recognized. It is also important to remember that this experiment was performed by actual professionals in the network security corporate industry, participants who would on any given day, be asked to perform security analysis on a corporate network.

## **Assumptions**

The more vulnerabilities an analyst can find, the better he/she can correlate these findings into recommendations against the corporate security policy.

The data used was from network security analysts of one company only. It is assumed that these results will show effectiveness of frameworks for this particular company and it cannot be assumed that this is the general outcome for network security analysts across the industry.

It is assumed that in the time limitation of one hour, a generalization can be made about the experiment and the analyst ability to gauge misconfigurations, weaknesses and vulnerabilities.

## **Definition of Terms and Variables**

The terms network and security have various meanings to various people and different interpretations can very much change the scope of the study. This next section will proceed to clarify what is meant by each term so as to understand the bounds of the study.

### *Network*

Merriam Webster defines a network as, "a system of computers, terminals, and databases connected by communications lines" (Webster). Webopedia defines a network as, "two or more computer systems linked together" (Webopedia). Essentially, a network is comprised of two pieces: computer systems and the medium of communication between the systems. The reason for having a network is for two or more systems to share data.

### *Network Security*

Network security involves protection of data while it is stored, processed and transmitted. Any component within the communications infrastructure that has a part in storing, processing or transmitting the data is a part of the security structure. This becomes more difficult to define when the human component is factored in. Human interaction is also a part of this data communications system. Humans, at every layer of network security, can compromise the safety of data through means including not following processes and procedures, giving out passwords or confidential data to unauthorized users, and simply by making mistakes. There is a physical component to network security as well, someone could not only steal company secrets via a web vulnerability but they could also physically walking onsite and steal a server. Network security in its entirety includes all aspects of data protection including the technology infrastructure, physical protection and human components. This study will only be analyzing one of these three aspects of security; this study will analyze the effectiveness that network security frameworks have on the electronic infrastructure component only. The definition of network security, for the sake of this thesis, is to protect data against electronic attack while it is stored, processed and transmitted.

### *Network Security Analyst*

A network security analyst is one who would be in a position to review and understand the security posture of an environment or piece of an environment and give recommendations for improvements. The goal of an analyst is not to just find holes, but to fix security in an environment. However, an analyst typically does not do anything to actually implement improvements, the initiative for change generally stems from executives while the actual fixes are implemented by system administrators.

### *Attacker*

An attacker is anybody posing a threat to the company, whether knowingly or unknowingly. To clarify, an attacker to network security can be anything knowingly or unknowingly causing damage to a company by compromising the electronic network infrastructure. In the case where security mechanisms are in place, this attacker takes advantage of a weak security implementation or a vulnerability in the security mechanism. Attackers include but are not limited to hackers actively attacking the network, curious script kiddies (newbies honing their skills), unauthorized eavesdroppers passively listening to the wire for information and employees obtaining unauthorized data. The scope of this study has limited the scope of attacker. Physical attackers and social engineers exist outside the scope of this test. These two parts of security are extremely important but for the feasibility of the experiments and tests involved, and for the factor of narrowing scope, these have been eliminated. This means the study will not be an all inclusive security study, but rather will be able to understand security against cyber attackers – those compromising the system using a computer terminal of some sort, whether on the environment network itself or remotely accessing it. It is important to note that if social engineering and physical security aspects are not addressed, danger still exists and the environment can still be compromised. Danger will always exist and no environment will ever be able to be entirely locked down. While this study only confronts the danger of the cyber attacker, this is still important and should be considered a major threat that needs to be addressed in every security environment.

### *Weaknesses, Misconfigurations and Vulnerabilities*

Throughout this study, analysts are asked to find misconfigurations, weaknesses and vulnerabilities in components causing some sort of threat to an environment. These are always grouped together because they embody the types of things that can be dangerous or harmful to a networked environment. This can range from a simple information leak

(weakness) to a threat that can be exploited to take control of the environment (vulnerability). Sometimes there is a simple misconfiguration in an application that leaves a door open for an attacker. As with the definition of an attacker, this definition will have to be scaled in scope for this study. From a broad perspective, this vulnerability, weakness, or misconfiguration includes any possible practical or theoretical way to retrieve unauthorized data, alter data, or cause harm while data is stored, processed and transmitted. Data can be stored on a computer hard drive or a person's brain. However, we are not looking at the human aspect in this study, just the cyber aspect. Some vulnerabilities are seemingly harmless and are actually implemented intentionally without realizing the danger that can be had by having them there. Having a contact name in a webpage html source can be an example of something meant to be useful but can have adverse affects when an attacker gets a hold of it. This would be considered a weakness as an attacker use this information to social engineer information. In this specific study, a misconfiguration, weakness or vulnerability will be restricted to mean a cyber hole – something that can be found electronically, via a computer workstation. These holes can stem from a security policy not being properly implemented but for this study, holes stemming from a user freely giving out their password to someone social engineering them will be excluded. This is not to say that the company should not have training on social engineering and how to prevent it, but simply that this study must be restricted. The value of the study is still intact because network security analysts will need to be able to identify cyber holes in the environment.

### *Effective*

The problem statement of the study is “The effectiveness of network security frameworks across skill levels.” It is important to understand how effectiveness is measured. One way to approach this is to measure how many additional weaknesses, vulnerabilities and misconfigurations a user finds in an environment using a framework (compared with not using a framework.) The flaw in this definition is that each weakness, misconfiguration and

vulnerability is a symptom of a disease, the disease being a poorly implemented or null security policy or process. An effective network security framework would not just identify these weaknesses but would also come paired with recommendations on mitigations. The goal of an analyst is not to just find holes, but to fix security in an environment. The effectiveness of a security framework really lies in how it can help cure the disease to create a more secure environment. However, in order to understand the disease, the network security analyst must identify the symptoms. On one hand, testing the number of holes suppresses the real goal of the framework. On the other hand, if the holes are not identified, it is difficult to diagnose the disease. In this specific study, effectiveness will be measured by number of weaknesses, misconfigurations and vulnerabilities found. The author understands this is not the real definition of “effective,” but this definition must be used during the study due to constraints of time. The author feels this definition will be adequate to solve the objective of the study.

## **Conclusion**

There has been a growing demand for security that matches the growing trend of cyber attacks. This demand for security cannot keep up with the infrastructure to protect networks and the supply of qualified security analysts. There has been a slowly emerging security community, one tool being developed in this community to combat the current problems is network security analysis frameworks. These frameworks would be useful to both technology users and providers, but these methodologies currently have not been evaluated for any type of effectiveness study. This study will proceed to understand the potential and current effectiveness of these frameworks as they have the potential to be an asset to the network security industry.

## CHAPTER 2

### LITERATURE REVIEW

This study involves analysis of network security frameworks as well as an experiment to explore these frameworks. Accordingly, the literature review is broken down into two parts; studies were done of both network security frameworks and experimental sciences. It is important to understand the background of frameworks to understand the future of frameworks. It is important to study the experimental science behind this study to understand how the specific experiment was designed and what criteria was used to make the study significant and credible.

### NETWORK SECURITY FRAMEWORKS

The first literature review section will encompass network security framework research. This review will cover the history of network security frameworks, the trends these frameworks show and predictions about future frameworks.

#### **History**

As the Internet only really started to take off in the 1990s, the majority of the security framework evolution was accomplished in the last 10 years.

#### *In the Beginning – Prior to Frameworks*

Before the existence of formal network security frameworks, there was still a fair amount of security literature that pertained indirectly to frameworks; that is to say, security literature that would influence the future frameworks written. The first big security publication of this kind was made in 1985 by the Department of Defense (DoD) and is entitled Department of Defense Trusted Computer Security Evaluation Criteria. This is the beginning of the release

of what is referred to as the Rainbow Series computer specifications. The rainbow books are a series of computer specification publications setting precedence for what is considered secure for various levels of government clearances. The rainbow series, currently with thirty different color books, continues to expand even today. However, of particular interest is the *Orange Book* released in 1985 (TSEC). This manual was a first to discuss and quantify different levels of computer security. In particular, the *Orange Book* discusses computer operations including architecture, storage, covert channels and integrity. This very granular guide gives quantifiable levels of security specifications (C1,C2,C3,B1, etc) enabling labels to be placed on systems to clearly identify the level of protection the system provides. While the *Orange Book* is still updated and used to this day, it is not a framework as much as an evaluation guide. However, many later security frameworks built off the concepts formed in this document. The *Red Book*, released in 1987, is also an evaluation guide but this book is focused on computer networks whereas the *Orange Book* is specific to computer systems. The *Red* and *Orange Books* are not network security frameworks, but they begin the story of how frameworks were born.

### *The First Frameworks*

The first formal documented security framework was published in 1988 by AT&T entitled, *A Security Methodology for Computer Networks* (Pierson and Witzke). Contrary to its title, this publication did not provide a methodology as much as it provided a small list of areas to address during network security analysis; it is better likened to a checklist. This becomes a consistent trend in the field of security frameworks: a publication named a methodology but consisting of a checklist. The next break into the framework space happened in 1990. The IEEE Magazine published an article describing a methodology for network security (Graft et al). Until this point, nothing had been done to address the problem of security and properly analyzing security in a consistent and measurable way. The goal of this article was to address the possibility of even creating a methodology for network security.

To try and answer this question, the authors created a framework and analyzed it against an application. This first framework was a framework for designing a secure network – not analyzing an already established network. The first step in this framework is the same as the first step in every framework that followed which is evaluation of need – in this case, coined “determining system requirements, security parameters and constraints” (Graft et al. 53). In their write-up, the IEEE found a couple of interesting conclusions with their research. One conclusion is a problem that still exists with methodologies today: the very strict requirements of frameworks handicap the ability to adapt to different environments. In order to build a framework, a framework author has to specifically guide the user through a repeatable, measurable and comprehensive method. In order to have such a clean analysis, steps must be carefully followed. The problem is that this starts drawing thick lines around the methodology, lines that cause the method to only be useful to a certain subset of variables. The clearer the lines are, the more measurable and repeatable the method but the smaller the subset of variables that the method is useful to. A big challenge in the network space, from this first study up until this current study is to build a framework that is flexible and comprehensive as well as concretely repeatable, measurable, and useful. A second and equally important conclusion from the IEEE article is that, even with the limitations of a framework, the authors all agreed that a methodology is “badly needed” in the industry (Graft et al. 57).

#### *The Government Sets a Precedent*

While this first methodology study took the first step in proving that a formal framework is not only feasible but possible, the subsequent methodologies attempted to perfect this idea. This first methodology by IEEE was written in 1990 but between 1990 and 1997 there is a dearth of information. In 1997, the Department of Defense (DoD) released the, “Common Methodology for Information Technology Security Evaluation.” One year after that, they released the, “Common Criteria for Information Technology Security Evaluation.” These common criteria methodologies are thousand page documents that cover, in detail,

how to evaluate security products and systems in accordance with international standards (ISO15408:1999). This type of evaluation permitted an end user to compare and assess various products according to one specific model. The trend of DoD publications is now starting to peek through. Their publications were usually based upon evaluations for comparison and classification of systems. So while they built a framework in that they underscored important areas to secure, the evaluation guide did not enable analysts to secure a system or network. Further, these guides were very complex and bulky to say the least. They did not provide a flexible approach usable on every day systems. Rather these DoD publications were designed for matters of national security and not for a corporation interesting in securing their network.

### *Standards Emerge*

The next big leap came in 1997. The IETF (Internet Engineering Task Force), a technical body of international volunteers concerned with the internet, created RFC 2196, formally titled *Site Security Handbook* (Fraser). The task force included network architects, vendors, researchers and operators. The document they created was to be used as a guide when creating corporate security policies. This publication gave factors to consider when creating security policies: identify threats, identify assets, identify who should be involved, and identify characteristics of a policy. Further it lists the areas of concern that the policy writer should consider in network security: services, firewalls, authentication, and passwords to name a few. This was and is still an excellent guide: exhaustive, applicable and time enduring.

### *The Deluge: Government, Standards Organizations, Vendors, Open Source*

After the DoD and IETF set the precedent for network security frameworks, the deluge of frameworks began. The year 2000 brought in a first manufacturer security framework. Cisco Systems published, *SAFE: A Security Blueprint for Enterprise Networks*

(Convery, et al). Cisco, a giant in the networking industry, wrote this framework for deploying a secure modular architecture design. While architecture oriented, this model also addresses applications, policies and specifics like server patch updates. The focus, however, is mainly architecture so the other issues, while mentioned, are not discussed in any detail. As expected, this framework is very specific to Cisco products. The year 2000 was also the start of the *Open-Source Security Testing Methodology Manual* created by Pete Herzog. This methodology is continually updated by experts in the industry, the latest update being August 2003. This manual is the first practical hands on reference guide to testing network security in its entirety. The Open-Source methodology gives a framework for analyzing security from all angles: from security policy to sever level vulnerabilities to social engineering to application layer reviews. The main goal of the document is to enable analysts to review all aspects of network security such that once all areas are tested, a comprehensive review of network security will have been completed.

#### *Frameworks Splinter into Disciplines*

2001 brought the *Octave Criteria* by CERT (Alberts and Dorofee). CERT is an internet security research center that provides security advisories, education and publications. This CERT Octave is a framework with a slightly different twist than the previously mentioned frameworks in that it is an evaluation model that centers on threat, asset and vulnerability evaluations. This is the start of a trend of niche frameworks: frameworks that do not address network security in its entirety, but do a comprehensive analysis of one perspective of security. The CERT risk evaluation process does not describe how to secure a network but rather how to analyze where potential vulnerabilities lie via three steps: building asset profiles, identifying infrastructure vulnerabilities and developing strategies and plans. Although this is not a detailed security model, it is generic enough to apply to any network. The Octave model is centered on protecting only the most critical assets and so it is not quite comprehensive enough to apply to an entire network.

At this point, frameworks started to split in various directions. It seems appropriate because also at this point, the security industry was growing rapidly and the body of knowledge necessary to understand security became so large that it was challenging to develop one framework covering all aspects of security.

Next in the chain of events, the National Institute of Standards and Technology (NIST) started releasing formal security publications. In 2001 came the *Security Self-Assessment Guide for Information Technology Systems*. In 2002 came *Guidelines on Firewalls and Firewall Policy* and *Security Guide for Interconnecting Information Systems Technology*. The latter is a general guideline for how to deploy an interconnection and as such is connection based. Connection based means it gives the reader a guide of things to think about when setting up a network connection encompassing technical, personnel, and policy issues involved in setting up a communication. So it not only encompasses security but basic practical day to day functions. This is one of the first all-inclusive checklists covering a wide range of topics in security including: firewalls, intrusion detection, auditing, authentication, logical access, virus, and encryption. This is a good example of a framework set up as a checklist. It provides the analyst with a checklist of what to think about and what to possibly include in the network to provide security. What it does not do is tell the analyst how to test this security, making the framework more theoretical than practical. This reveals another theme amongst frameworks; most frameworks fall into a theoretical space, listing important components but relying on the reader's expertise to interpret and be able to implement the advice.

#### *Niche Frameworks Grow – Commercial Publications Appear*

It was previously mentioned that the security frameworks are breaking into niche groups because of the growth of the industry and the vast knowledge necessary. The quick overview of what happens with these niche frameworks can be given by following one set of books, *Hacking Exposed* and its family: *Hacking Exposed Linux* (Hatch and Lee), *Hacking Exposed Web Applications* (Scambray and Shema), etc. The first book in the series, *Hacking*

*Exposed*, claims to be a security methodology for hacking a system (McClure et al). However this definition of methodology is different than the definition this thesis has been using. This book is essentially a guide for how to hack a system or network; it is a security methodology for hacking, rather than securing. This thesis purports that a security methodology is for evaluating security posture of an environment rather than simply hacking one piece of an environment. Admittedly, it is important to know how to hack something in order to understand how to fix it but the two are not always directly related. Because one knows how to hack a system does not necessarily mean one knows how to secure an environment. Nonetheless, this suite of books gives a great example of what is happening and what will continue to happen in the framework space. First came the all inclusive *Hacking Exposed* but the body of knowledge became so big that there had to be books for each discipline within security. Security methodologies of both definitions (the one used by *Hacking Exposed* and the one in this thesis) follow this trend.

#### *Certifications Appear*

In 2003, Pete Herzog's *Open-Source Security Testing Methodology Manual* was updated to version 3.0 and with that came an accompanying certification and training courses. Ideally, a trained, qualified tester using the methodology can be confident of competently analyzing network security in a comprehensive, repeatable, measurable way using this methodology. This certification does not mean the environment is hackproof, rather it means that the environment was evaluated to specified standards.

#### *Frameworks Not Mentioned*

Frameworks not mentioned in this literary review were frameworks extremely specific to one technology, for example, a framework for securing an ATM network (Schumacher and Ghosh). The security industry is a quickly moving target and the author predicts that as quickly as technology changes, frameworks will continue to branch out into many directions.

## *Conclusion*

The study of frameworks began with an AT&T research journal publication and ended up spanning various government, industry and commercial publications. They range from theoretical to practical approaches and span all types of technologies. Themes emerged in the history of frameworks that still hold true today:

- The first step in a framework is generally always the same: the evaluation of need: determining requirements and constraints.
- The very strict requirements of frameworks handicap the ability to adapt to different environments.
- Most frameworks fall into a theoretical space, listing important components but relying on the analyst's expertise to interpret and be able to implement the advice.

## **Trends in Network Security Frameworks**

The main trends in the security framework space are threefold: framework development is gaining momentum, frameworks are starting to expand into particular security disciplines, and frameworks are starting to be accompanied with certification-like status.

### *Gaining Momentum*

Security frameworks are slowly gaining momentum. In the 80's there was one framework. The 90's brought five additional. Between 2000 and 2003 there have already been at least five major publications establishing frameworks, with numerous commercial publications (such as the Hacking Exposed series) and various studies unaccounted for in this number. This trend shows no signs of decreasing.

### *Splintering Into Disciplines*

The security industry is growing quickly with new vulnerabilities found every day. As new technologies emerge, so do new security considerations and accompanying methodologies. This leads to a trend of specialty methodologies for specific technologies. The growth of knowledge necessary in the security industry makes it harder to create one framework that encompasses all of security especially for niche technologies, emerging technologies, and specialty areas. This author predicts that security will become more increasingly difficult to master which, due to the increasingly catastrophic nature of attacks, is obviously a major problem. One way this industry will need to keep up with the great need to analyze security and perform security on networks will be to have an increasing number of specialty frameworks.

### *Certifications and Stamps of Approval*

The author predicts a trend that industry accepted frameworks will follow the *Open-Source Testing Methodology Manual* example and start to include stamps of approval or certification status. Certifications have been a common way to show credibility in a vague area like security. Today, certifications are offered in many areas of computer technology: from the operating system hardware, applications, etc. Technology has been using certifications as a way to show utility and effectiveness in many industries and this trend will probably continue to spill into the security framework space. In security, it is difficult to prove that proper analysis is done because there is no final tangible product; often there is only a report. In the case of a report with no findings, it is difficult to understand the scope of work involved and the depth of testing performed.

### **Future of Frameworks**

After looking at the trends it seems clear that frameworks are moving quickly into the network security space. There is no one clear framework that is recognized as the ultimate standard of frameworks. It appears that frameworks are tending to move towards certifications: frameworks that help guarantee a specified standard of analysis. The author of this study predicts that network security service providers will begin using their methodology as a selling point. Specifically, the selling point will be their “seal of approval.” And while a service provider will never be able to guarantee a risk-free environment, it will be able to guarantee a specific level of quality. The author also predicts that qualified testers will emerge with correlating certifications. In sum, the future of frameworks is to become a selling point for service providers, possibly with different providers marketing different niche certifications.

## **EXPERIMENTAL PSYCHOLOGY**

The second part of the literature review is exploration of experimental studies in relation to the experiment performed in this study: what knowledge is available in the psychology space on the types of experimentation performed, the controversies surrounding experimental sciences, and the pertinent information necessary for designing the experiment.

This thesis involves an experiment with humans. Specifically it involves collecting data about the skill level of participants, creating a hands on experiment environment, and tracking participant outcomes to understand the usefulness of network security frameworks. Because of the human component in this experiment, the area of psychology was explored for best practices to protect the authenticity and credibility of the data collection. The literature review is mainly focused in the learning technology space. This is the area of psychology that addresses issues such as evaluating usefulness of learning tools and evaluating skills of participants. In a more general sense, the problem with experiments like the one proposed in this thesis is credibility. In order to combat any question of credibility, there are a few areas that need to be underscored during testing: qualitative versus

quantitative data collection, practitioner versus evaluator, and pre-made tools that should be used.

## **Controversies in Data Collection**

### *Qualitative versus Quantitative*

There is some debate on how research should be addressed with respect to collecting qualitative data versus collecting quantitative data and when each is most effective. This study uses a combination of both methods with the intention of allowing the methods to complement each other. The combination of both methods gives a broader understanding of the answer to the question and discovering any resulting convergences.

### *The Paradigm Debate*

There is something called the “Paradigm Debate” in which various researchers argue the validity of the two methods of data collection (Oliver, 5). This is a “philosophical debate about the relationship between the natural and social sciences” (Risjord et al. 40). Specifically, the debate is about benefits of collecting qualitative versus quantitative data versus collecting both. Qualitative data is interpretive and subjective, for example an interview, and is useful because it is very flexible to specific environments and problems. Qualitative data is also usually only focused on small groups of participants (Risjord et al. 47). The problem with qualitative data is it is hard to prove relevance and difficult to pull patterns and support conclusions (reductionism.) The second type of data collection is quantitative. This type of data fits nicely into statistical algorithms and charts and can support conclusions definitively, but is very limited in scope. The limited scope means the question one is able to answer is very narrow (Risjord et al. 47). There are arguments for why either paradigm is better in any given case and the ultimate question that needs to be addressed in any research study is whether or not the audience will find the type of paradigm used persuasive

and credible (Oliver, 5). A good compromise that is mentioned in various pieces of literature is to do a combination of both and triangulate the results.

### *The Triangulation Debate*

However even triangulation brings debates. The basic concept of triangulation is, according to researchers Risjord, Moloney and Dunbar, “the use of more than one method to investigate a phenomenon” (40) These different methods of data collection are intended to corroborate or complement each other so as to have a more confident answer to a problem (Risjord 44). It also is meant to help provide an answer that could not be given by one method (Risjord 44). However researchers agree that simply triangulating data does not in itself provide any guarantee that the method will provide unbiased and properly correlated answers to a question (Fielding and Fielding, 24). However, in the specific case of the study on network security frameworks, the duality helps to provide a broader understanding and is purposefully used because each method, on its own, is unable to answer the broader question posed about framework effectiveness. A qualitative study of interviews and questionnaires is not able to definitively show how effective the frameworks are and a quantitative study, due to the nature of network security, narrows the question down too much. The decision was made to proceed with a triangulation method with the intent that linking the data together will complement the findings and will give a broader perspective of effectiveness of security frameworks across skill levels. That is to say, the intent is that the qualitative data collected will help to increase confidence that the quantitative data was meaningful. The qualitative data will not solely be used “as a life-saving device to resuscitate” what was a failed experiment; it will be used to increase confidence and broaden the understanding (Green et al. 269). While the qualitative data collected in this study will give an increasingly broad and complementary view, it will not necessarily give a more objective view (Fielding and Fielding, 33).

### *Evaluator versus Practitioner*

The second issue to consider in experimental testing is evaluator versus practitioner. (Oliver , 5) Evaluators are professionals in evaluation, not in the subject being evaluated. Conversely, practitioners are professionals in the subject but have little or no evaluation skills. There are benefits to each, professionals being very sensitive and intuitive about the subject matter while evaluators can be more objective. This thesis will have to be performed by a practitioner and not an evaluator and this can jeopardize the results of the experiment if caution is not taken (Oliver, 5). The caution needs to take the form of a pre-made evaluation tool which will be explained in the next section.

### *Pre-Made Tools, Handbook Used*

Pre-made evaluation tools are experiment frameworks that guide a practitioner through an experiment. There are two that are highly respected: The LTDI Evaluation Toolkit (Havery) and the ELT Toolkit (Oliver). The author of this thesis used the LTDI Toolkit as a guide to designing and performing the experiment in order to help make careful, informed decisions and to preserve the integrity of the testing.

### **Conclusion**

The literature review encompassed a study on network security frameworks and the experimental psychology necessary in their effectiveness. Security frameworks are quickly developing and in order to test them, specific experimental psychology details should be followed to get the most reliable results.

## CHAPTER 3

### MATERIALS AND METHOD

The materials and method section is meant to be a recipe for anyone who would like to recreate each component of the data collection within this study. The components are broken down into three pieces; qualitative collection (experiment), quantitative collection (survey) and qualitative study (IDS research and correlation). The bulk of the research lies within the experiment and the survey; the Intrusion Detection Sensor (IDS) research is intended to complement and broaden the understanding of the data found in the experiment.

### QUANTITATIVE STUDY

Framework effectiveness has not been studied previously because there is no data in existence that can answer the question. Because no such data exists, it has to be created. It was decided that the only way to truly understand effectiveness is by a hands on road test of a framework. Using a standard, industry recognized, well developed framework and a specific security problem, the experiment will produce data to help understand if network security analysts do a more comprehensive job answering the problem using a specific framework as a guide. The experiment will test actual outcomes of how well analysts solve a security problem in two cases, with a framework and without a framework.

#### **Designing the Experiment**

In the literature review, there is a discussion regarding how experiments involving humans should be designed. The psychological literature cautions experiment practitioners of the pitfalls in evaluating the usefulness of various data collection tools. In order for this research to be credible and because the practitioner is not a skilled evaluator, a pre-defined

experiment design will be used. The toolkit chosen was the LDTI Toolkit. Following the LDTI Toolkit recipes, the following sections delineate how the experiment is designed.

*\*\* (All quotes from LDTI Toolkit)*

“What will your evaluation do?”

This evaluation will test the usefulness of a security framework in analyzing vulnerabilities in an environment. This evaluation will be the first time any such framework is analyzed so it is important to show firstly that frameworks can be properly tested and secondly, discover whether they indicate usefulness to users and thirdly, to what extent these frameworks are worth further study.

“Who is the evaluation for?”

This evaluation is for a master’s level academic research project with the specific intention that professionals and management alike will be able to understand the value of security frameworks so that, if valuable, effort can be spent to develop frameworks and distribute to security professionals.

“Can you deal with the practicalities?”

There are two main practicalities that need to be addressed: the number of professionals that will be able to participate and time constraints of the experiment. The biggest concern is the number of professionals involved in the testing. Some studies show that 15 is the minimum number of participants in each section (ALT Workshop) and as this study will have four sections, this demands 60 participants. The study designer estimated to get between 40 and 60 participants. The second practicality is time limitations; each participant in the experiment is a professional in the network security industry. This means the experiment needs to be extremely limited in time – maximum one hour to finish experiment from beginning to end. In order to limit time, the experiment needs to be scaled down as much as possible. For example, it is impractical to ask network security analysts to

find all vulnerabilities given a sample network, as this could take weeks of full time work. Instead, this study breaks apart the framework and tests two specific tests within it: vulnerability research and firewall testing. These two were specifically chosen because they could be properly set up to take one hour.

The byproduct of addressing these practicalities is research bias. How much can an experiment be scaled down while still maintaining real life attributes? The tests have been designed to simulate real world problems but they will only be able to include a microcosm of real world components. The data of the experiment, though it will not show the entire macrocosm of security usefulness of frameworks, will help show in one area, how useful the frameworks are. From here some trends can be analyzed and further studies can be recommended.

“What are your deliverables?”

The deliverable is a thesis research report prepared for a university panel of professors with the intention of also sending results to corporate management and technical journals.

### **Choosing a Methodology**

After researching the many types of methods in the toolkit, two pre-defined cookbook recipes are chosen: Designing experiments and pre/post testing questionnaires. The reason for these choices of methodologies is very specific. The study needs to be authentic but it is impractical to carry out this type of test in a corporate production environment. Instead, a simulated real-life lab will be set up. The pre questionnaires are given for two reasons, firstly to scale down the experiment and only allow participants who pass specific criteria. Secondly these questionnaires are a way to obtain skill level data on each participant in a relatively quick way. A benefit of questionnaires is the minimal time impact to the participant. The time that network security analysts are willing to give to a graduate study is limited so to protect their time and try to get the best possible data, questionnaires were the best choice of

determining skill. After the experiment there will be a post experiment questionnaire to help understand how valuable the framework is to the participant and to understand potential usefulness of these frameworks in the workplace. These post experiment questionnaires offer a qualitative view that the experiment itself cannot provide, affording an opportunity to explore any scenarios that are bound by the experiment. The post survey questionnaire consists of closed ended and open ended questions. Between the qualitative experiment data and the quantitative post surveys, data can be cross correlated to get the best understanding of the answer to the question. (The pre experiment questionnaire exists in Appendix A and the post experiment questionnaire exists in Appendix B.)

### **Gathering Data**

There is a pre-experiment questionnaire, a post-experiment questionnaire and performance test data. There is also the actual lab environment setup where data is gathered regarding how many vulnerabilities each participant found.

#### *Pre-experiment questionnaire*

The first step in gathering data is sending out an email asking participants to be involved in the study. If they choose to be in the study, they will fill out a skills survey (pre-experiment questionnaire) and a Human Research Committee approved consent form. The pre-experiment questionnaire is a skills survey that asks pertinent info about the participant's background in security, specific skills within security so that a proper skills ranking can be given to the participant. This questionnaire in its entirety exists in Appendix A.

#### *Experiment*

The second step in gathering data is inviting participants join the experiment. This section will replay the experiment details precisely so that this experiment can be repeated.

Experiment setup:

## Vulnerability Research: Server Setup

In order to do this piece of the experiment, a real-world simulated server is created. To make the experiment useful, this server needs to have vulnerabilities on it; enough to constitute a range of vulnerabilities. Further the server will have at least one false positive in it. The creative of this false positive should not be too contrived as the intent is not to bias the experiment or trick participants intentionally but rather to have a fair test. With these requirements in minds, the operating system chosen was a default install of Linux RedHat 7.1. (RedHat is chosen because there is no licensing fee necessary and 7.1 chosen because of innate vulnerabilities within it.) The options included adding a webserver, FTP server and SSH server. The SSH server was patched by following the typical RedHat download processes. The SSH upgrade through this process is what gives the false positive – when RedHat patches SSH does not change the banner so scanners will see this as being downlevel. Specifically banners will report that it is version openssh 3.1p1. The next step in building the experiment server is adding a line into the httpd.conf file instructing the webserver to not give out banners when nodes request it, this creates a condition where some scanners would not understand what level http server is installed so the scanner will not report a finding even though vulnerabilities exist. Everything else in the server is left default.

## Firewall Testing: Firewall Rule Setup

As with vulnerability testing, the goal with the firewall environment is to keep the rule set as unbiased as possible. If the practitioner of the experiment hand writes the firewalls rules, there is potential for bias towards making some rules fit in a specific framework or not fit in a specific framework. A default install of firewall rules is not an option because no functioning real-life environment could run properly on any type of generic rule set. Therefore it was decided to use firewall rules of an

existing environment. An environment is chosen that has many fundamental problems with the firewall rules. These firewall rules come paired with a network diagram. The only obstacle is that the firewall rule set was 30 rules long and for the experiment to last in one hour, the firewall rules need to be ideally 10-15 rules. With that in mind, a third party with little knowledge of the experiment is asked to parse the rules, protecting the functionality but removing and fat, so to speak. The rule base is scaled down to 11. The next order of business is to sanitize the rules and network diagram to insure the environment is fully anonymized. To accomplish this, all IPs are modified, and all hardware specifications of the network diagram were removed. Each piece of the puzzle is placed into a .gif file or .pdf file, stripping any company sensitive information off the document. The environment is now ready to be distributed to experiment participants. Note that in this experiment setup, there is no live firewall available to test, whereas in the vulnerability testing, there is an actual server up and running for people to poke at. The reason is the tests involving having the firewall up are much too time consuming – there is no chance of keeping the experiment under one hour. So, the firewall test is given with just firewall rule set and a network diagram. This scales down the firewall test but still protects the usefulness of understanding if the framework is effective in the portions relating to firewall rule setup. Actual firewall rules and network diagram exist in Appendices C and D.

#### Experiment Distribution:

Logistically, this experiment can be done from any remote location so distribution is all via email. An email is sent out paired with a skills survey and a consent form. Once these are completed, the experiment is mailed out. Each person is instructed to notify the practitioner when they start, and send the results one hour later. Logistically, in order to obtain a large number of people, participants range nationally and accordingly, all distribution and monitoring has to be completed remotely. Participants are given instructions and informed they can use any

resources such as books or the Internet but are not permitted to ask peers, friends or any other humans for help.

The reason for obtaining the skills survey first is so that the practitioner can quickly plug in questionnaire answers into a spreadsheet which has a formula programmed to rank skill level. This is to ensure experiments are distributed evenly across skill levels.

#### Post Experiment Survey

Participants are asked to fill out survey delineating all findings found during the exam. See Appendices F,G,M and N. Further, those using a framework during the experiment are additionally asked to fill out additional questions about framework usability and potential effectiveness. See Appendix B.

#### *Details: Vulnerability Assessment*

The experiment implementation details specific to vulnerability assessment are that each participant is given two vulnerability scans of the server when they are emailed the experiment package. (Appendices I and J) There is deliberation on whether or not this is appropriate. On the one hand, a security analyst could never do a proper security analysis without knowing how to set up, configure and run vulnerability scanners. On the other hand, this experiment required half the participants to follow a framework specifically stating a requirement of running two vulnerability scans on the host which brought up two problems: time and bias. Again the reader is reminded that there was a strict one hour time limit of participants. There is no way scans can be run (possibly through firewalls) and analysis given within a one hour timeframe. Secondly, the type of vulnerability scanner would have to remain open to the participants choosing. There are innate differences in scanners and each is unique and displays different information about a server. Some scanners are better against windows operating systems, some better for \*nix. In this case, the practitioner would

then be unable to determine if the framework was helpful to the participants or if the type of scanner used was helpful to the participant because too many unknown variables would be introduced into the data. The goal of the study was not to test the usefulness of scanners. Ultimately, in order to best answer the question of effectiveness, two vulnerability scans will be provided for every participant. This means the statement, “participants flagged as not using a framework,” is misleading because these participants are starting off with a mini-framework, a framework including two unique vulnerability scans.

#### *Details: Firewall Test*

The firewall test has little extra details to describe. Each participant is given a set of firewall rules (Appendix C) a network diagram (Appendix D) and half of the participants are additionally given a framework to utilize while analyzing firewall rules. (Appendix E) The instructions (Appendices F,G) are to find any weakness and/or misconfigurations with the rules and then give overall recommendations.

#### **Data Analysis Method**

Once all experiment data is received, the numbers are analyzed firstly to understand if a normal population is represented and secondly to understand which frameworks were helpful to whom. Data is analyzed via quantitative statistics for the questionnaires and performance data.

#### *Skills Survey*

The first analysis of the data is understanding analyst skills. Each participant filled out a survey of questions to help the practitioner rate their skill level. (Appendix A) The exact formula for finding skill levels per question is in Figure 3.1 below. Each participant’s answers were recorded in a spreadsheet and each letter answer correlates to a number. All numbers are added up and divided by the maximum score. The score is put into a percentage and this

number becomes the participants skill level. The skill levels range from one to one hundred.

Next, the skill levels are plotted to identify any natural breaks in skill.

**Figure 3.1 Formula for Finding Skills Levels**

Question 1: =IF(X="A",1,IF(X="B",4,IF(X="C",7,IF(X="D",10))))
Question 2: =IF(X="A",1,IF(X="B",4,IF(X="C",7,IF(X="D",10))))
Question 3: =IF(X="A",1,IF(X="B",4,IF(X="C",7,IF(X="D",10))))
Question 4: =IF(X="A",1,IF(X="B",4,IF(X="C",7,IF(X="D",10))))
Question 5=IF(X="A",3,IF(X="B",3,IF(X="C",3,IF(X="D",0))))
Question 6=IF(X="A",0,IF(X="B",2,IF(X="C",5,IF(X="D",7))))
Question 7=IF(X="A",0,IF(X="B",2,IF(X="C",5,IF(X="D",7))))
Question 8=IF(X="A",0,IF(X="B",2,IF(X="C",5,IF(X="D",7))))
Question 9=IF(X="A",0,IF(X="B",2,IF(X="C",5,IF(X="D",7))))
Question 10=IF(X="A",0,IF(X="B",2,IF(X="C",5,IF(X="D",7))))
Question 11 =IF(X="A",1,IF(X="B",4,IF(X="C",7,IF(X="D",10))))
Question 12 =IF(X="A",1,IF(X="B",4,IF(X="C",7,IF(X="D",10))))
Question 13 =IF(X="A",0,IF(X="B",2,IF(X="C",5,IF(X="D",7))))
Question 14 =IF(X="A",0,IF(X="B",2,IF(X="C",5,IF(X="D",7))))
Question 15 =IF(X="A",0,IF(X="B",2,IF(X="C",5,IF(X="D",7))))
Question 16 =IF(X="A",0,IF(X="B",2,IF(X="C",5,IF(X="D",7))))
Question 17 =IF(X="A",0,IF(X="B",2,IF(X="C",5,IF(X="D",7))))
Question 18 =IF(X="A",0,IF(X="B",0))
Question 19 =IF(X="A",1,IF(X="B",4,IF(X="C",7,IF(X="D",10))))
Question 20 =IF(X="A",1,IF(X="B",4,IF(X="C",7,IF(X="D",10))))

*Findings Ranking*

As mentioned earlier, during the experiment each participant fills out an answer sheet listing every finding and recommendation they have for the specific problem they are given to work on. For each finding, the participant is allotted one point. Findings are averaged across users and skill levels to understand on average how many findings each participant finds with the framework and without the framework. Later, these numbers are compared to understand if there is a tendency for the frameworks to help analysts identify findings.

### **QUALITATIVE ANALYSIS**

All participants using a framework during their experiment are asked to fill out a short qualitative survey to understand subjectively, how the framework was perceived and also answer various questions posed about the study concerning the potential effectiveness of frameworks. The analysis of these questions is straightforward, each answer is merely counted and averaged. For example if four out of eight people say the framework was useful, the conclusion is that 50%  $[4/8]*100$  of the participants feel the framework was useful.

### **IDS STUDY**

To complement the experiment data, a study of Intrusion Detection Sensor (IDS) data is completed. Intrusion detection is a means of identifying attacks to a network. Sensors are set up at various critical segments on a network in order to alert appropriate technicians that a potential attack is occurring. The nature of IDS systems is reactionary as IDS systems used in this data collection are signature based. When a new attack is discovered, signatures of that attack are placed in a file and if the sensor detects traffic matching this pattern, it flags an alert to a monitoring station. The study scope will be to cross reference each attack with the same security framework used in the experiment to understand if the framework, in theory and if executed expertly, would find this weakness in the network so as to fix it. This study is to examine the theoretical and potential effectiveness of the same framework used in the experiment.

### *Data Attributes*

The study of IDS data involves collecting and sorting two months worth of attack signatures to understand the top 50 attacks in the last two months to a large scale operation of monitoring intrusions. The data is taken from a monitoring center that tracks thousands of sensors placed nationally on numerous networks. The method for studying the data is relatively straightforward, counts are made of each signature and the top 50 counts are pulled into a file. These 50 are considered the top and research is done on each of these to understand the nature of the attack.

### *Caveats*

There are a few caveats with this IDS study that need to be mentioned. IDS data, by nature is only going to catch unencrypted attacks going across the wire. IDS data would never catch encrypted attacks, social engineering attacks, new (zero day) attacks or fundamental flaws in network security architecture. IDS attack data is very limited compared to the number of actual threats and vulnerabilities in a given network. However, many threats and vulnerabilities are unknowns and are theoretical, this IDS data gives a good solid understanding of actual pertinent attacks, which is why it was chosen for the study.

This IDS study will help identify the potential effectiveness of network security frameworks from the perspective of active attacks across the wire. It will not take into consideration skill levels or practical effectiveness because the experiment data will help understand the latter.

### **Conclusion**

The intention of this section is to give specific details enabling a complete recreation of all aspects of the research performed in this study. The details provided show how the initial design is constructed following the LTDI toolkit, the experiment details for the qualitative and quantitative sections, the number crunching methods and finally, the IDS study.

## CHAPTER 4

### RESULTS

This section contains the results of the data collected throughout the study; data from the experiment, surveys and IDS study. The analysis of this data is given in the next chapter.

#### QUANTITATIVE RESULTS

##### Skills Survey

This first piece of data received from participants is a survey asking specific questions about skill levels. The analysis of this data tells quite a bit about the group as a whole. Here are some simple descriptive statistics about the group:

Total Number of Participants: 38

Average Skill Level 44

Highest Skill Level 75

Lowest Skill Level 9

Median Skill Level 42

*\*\*All skill level statistics were derived from the skills survey; all numbers are a number represented out of 100.*

Skills are also broken down by expertise. Specifically, between five and seven questions are asked specific to the participant's knowledge of firewalls and vulnerabilities. These are pulled out and averaged with the more generic questions to give these genre specific skill levels. Descriptive statistics are given of both in Table 4.1.

**Table 4.1 Descriptive Statistics of Skill Levels In the Population**

<b>Skill Level</b>	<b>Vulnerability Specific Skill</b>	<b>Firewall Specific Skill</b>
Mean	52	37
Highest	96	89
Lowest	12	6
Median	53	37

The skills surveys also give other interesting information about the characteristics about the population:

Figure 4.1 shows how many years experience the experiment participants have while figure 4.2 shows how many years experience these participants have in the computer industry in general.

**Figure 4.1 Histogram of Security Experience**

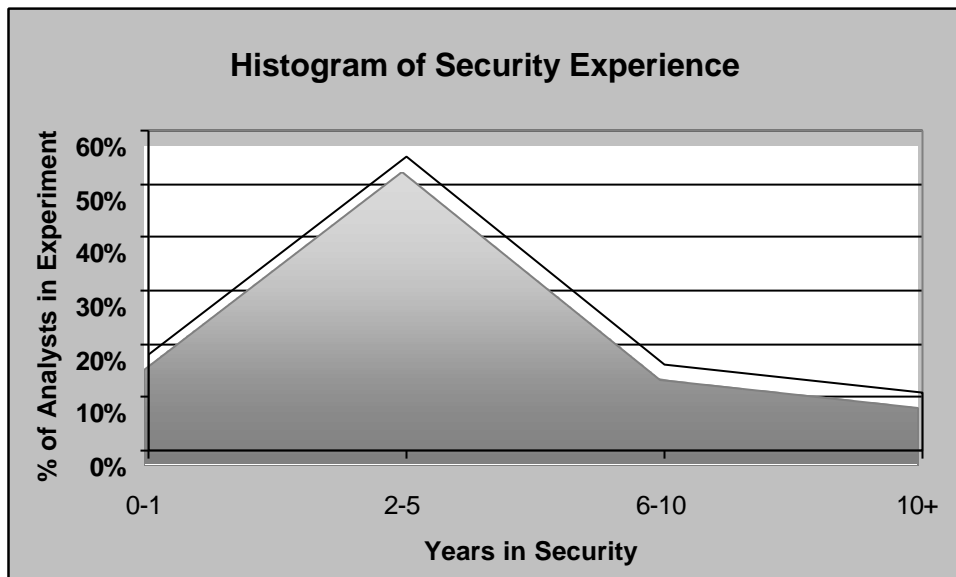
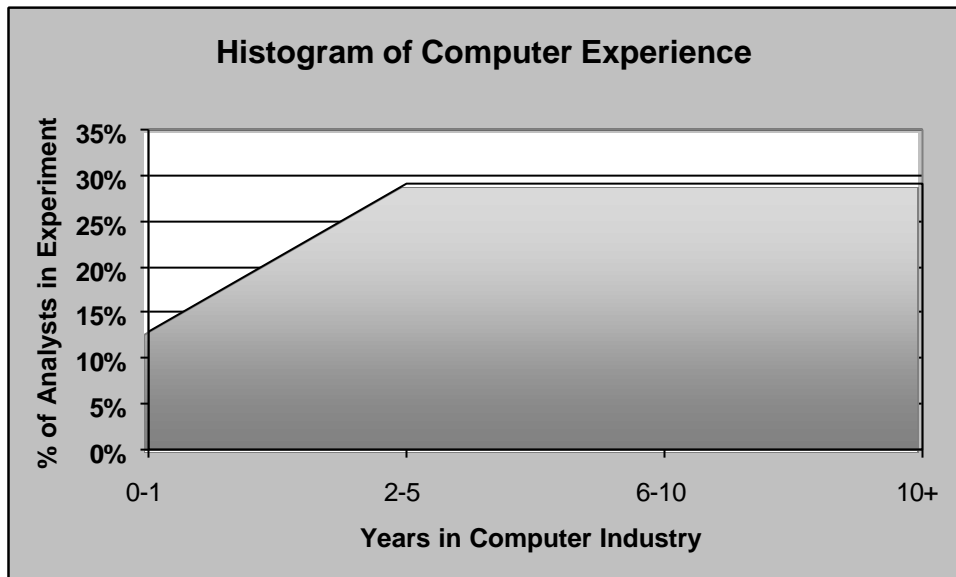


Figure 4.2 Histogram of Computer Experience



Additional statistics about the experiment population:

89% have a professional certification in security (ex: CISSP, GIAC)

92% have NOT used a formal methodology or framework to perform network security.

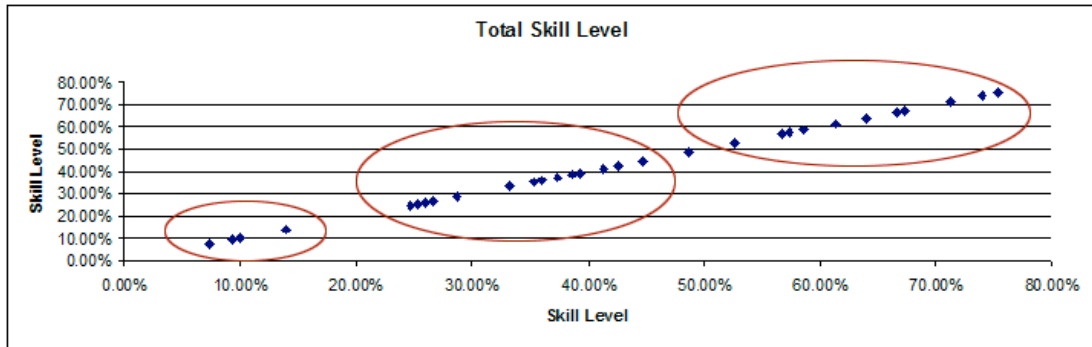
The goal of the experiment is to understand the effectiveness of the framework across skill levels and accordingly the skill need to be broken into groups. The individual skill levels are plotted and natural groups fall out of the graph. These are used for comparisons.

The graph of skill levels is shown in Figure 4.3. The natural breaks turned out to be:

Total Skill Breaks

- Level 1: 0-20
- Level 2: 21-45
- Level 3: 46-100

**Figure 4.3 Graph of Skill Levels**



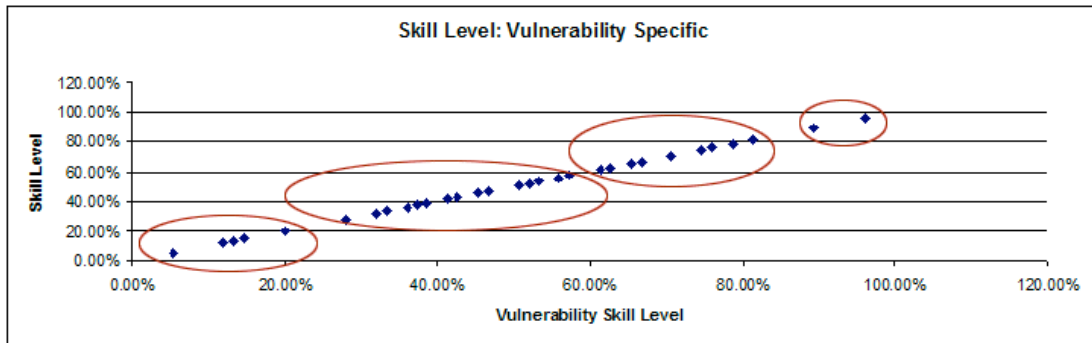
Note that figures 4.3 to 4.7 are purely linear graphs mapping skill level on both axes. The reason for such a simple one to one mapping was so that the author had the ability to play with the numbers to find natural breaks in skill levels.

Figure 4.3 shows the total skill breaks but because the experiment addresses two specific security disciplines, skills were further broken down into vulnerability skill level and firewall skill level. The results are below.

**Vulnerability Specific Skill Breaks**

- Level 1: 0-20
- Level 2: 21-60
- Level 3: 61-100 (note the last two breaks were put in the same group.)

**Figure 4.4 Graph of Vulnerability Specific Skill Levels**



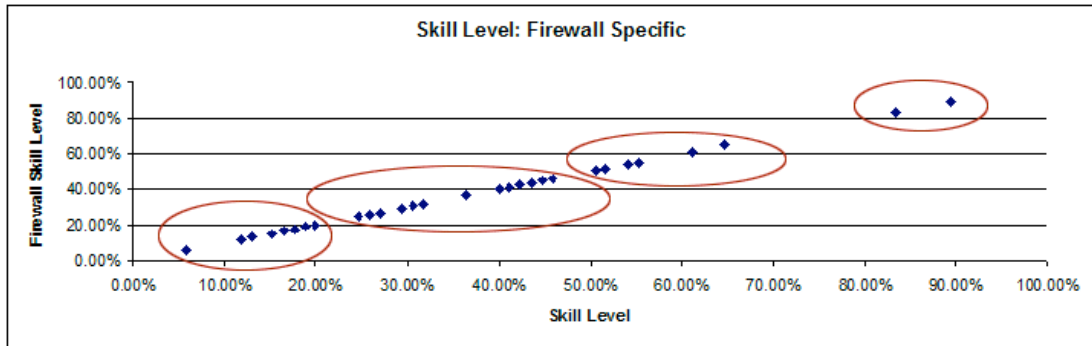
### Firewall Specific Skill Breaks

Level 1: 0-20

Level 2: 21-50

Level 3: 51-100 (note the last two breaks were put in the same group.)

**Figure 4.5 Graph of Firewall Specific Skill Levels**



The interesting discovery when breaking out skill level groups is that skill level groups broke into almost the same exact ranges whether the analysis be specifically for firewall, vulnerability or total skill.

When analyzing effectiveness against skill levels, even though the breaks of skill levels turned out equivalent, it is more sensible use the firewall skill level for the firewall experiment and the vulnerability levels for the vulnerability experiment. This is because the nature of findings and types of findings is very different for each test and also because many of the analysts tended to be much stronger at one or the other. This is normal as network security is such a large discipline that analysts tend to have niches and expertise in certain areas. It would be more confusing to put them all in one bundle to analyze effectiveness, thus the analysts will be gauged according to their specific level of expertise in the domain of their experiment. The breakdown of skill levels of people that actually participated in the two experiments are in figures 4.6 and 4.7 below. Note that these are the skill level breakdowns that will be used in the comparisons.

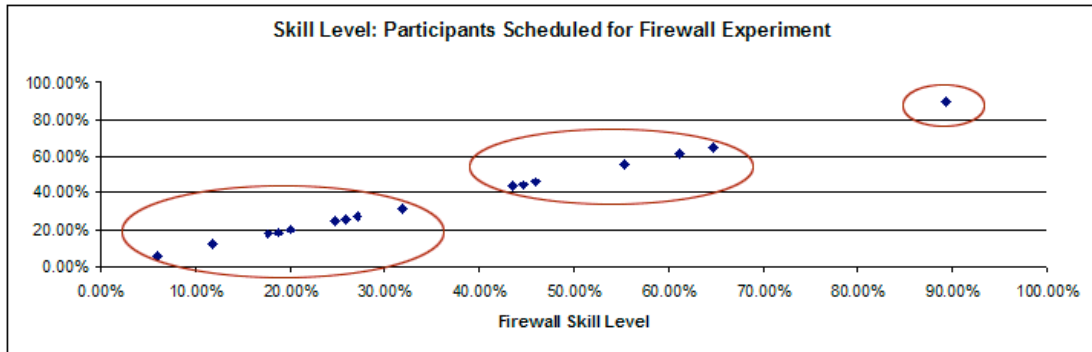
Firewall Skill Level Breakdowns:

Level 1: 0-40

Level 2: 41-70

Level 3: 90

**Figure 4.6 Skill Level Graph of Participants in Firewall Experiment**



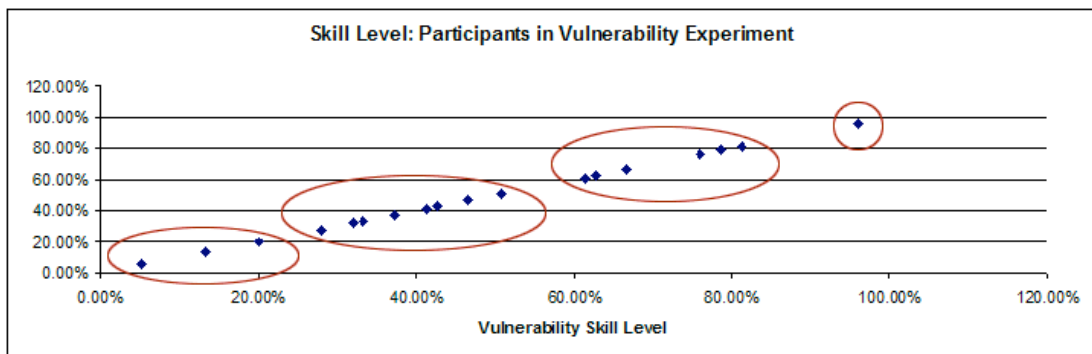
Vulnerability Skill Level Breakdowns:

Level 1: 0-20

Level 2: 21-60

Level 3: 61-100 (note the last two levels were combined)

**Figure 4.7 Skill Level Graph of Participants in Vulnerability Experiment**



## Experiment Results

The quantitative data results explore how many findings participants found on average. First is descriptive statistics of the population as a whole:

Framework: Vulnerability Research

Total possible vulnerabilities, misconfigurations or weaknesses: 20

Average vulnerabilities, misconfigurations or weaknesses found: 8

Framework: Firewall Test

Total possible vulnerabilities, misconfigurations or weaknesses: 15

Average vulnerabilities, misconfigurations or weaknesses found: 5

The following is descriptive statistics of the data broken down by skill level:

Framework: Vulnerability Research

Average vulnerabilities, misconfigurations or weaknesses found: 8

**Table 4.2 Actual Findings Across Skill Levels: Vulnerability**

Vulnerability Experiment			
	Skill Range	Findings With Framework	Findings Without Framework
Level 1	0-20	na	8
Level 2	21-60	6.7	9.8
Level 3	61-100	8.5	4.5

Framework: Firewall Test

Average vulnerabilities, misconfigurations or weaknesses found: 5

**Table 4.3 Actual Findings Across Skill Levels: Firewall**

Firewall Experiment			
	Skill Range	Findings With Framework	Findings Without Framework
Level 1	0-40	4.6	3
Level 2	41-70	5.7	11
Level 3	71-100	na	6.5

**Table 4.4 Average Findings: Vulnerability and Firewall**

	Firewall Experiment	
	With Framework	Without Framework
Avg Skill	35.6	46.3
Avg Findings	5.15	6.8
Vulnerability Experiment		
	With Framework	Without Framework
Avg Skill	59.2	51.7
Avg Findings	7.6	7.6

**Figure 4.8 A Graphed Comparison of Vulnerability Experiment Findings**

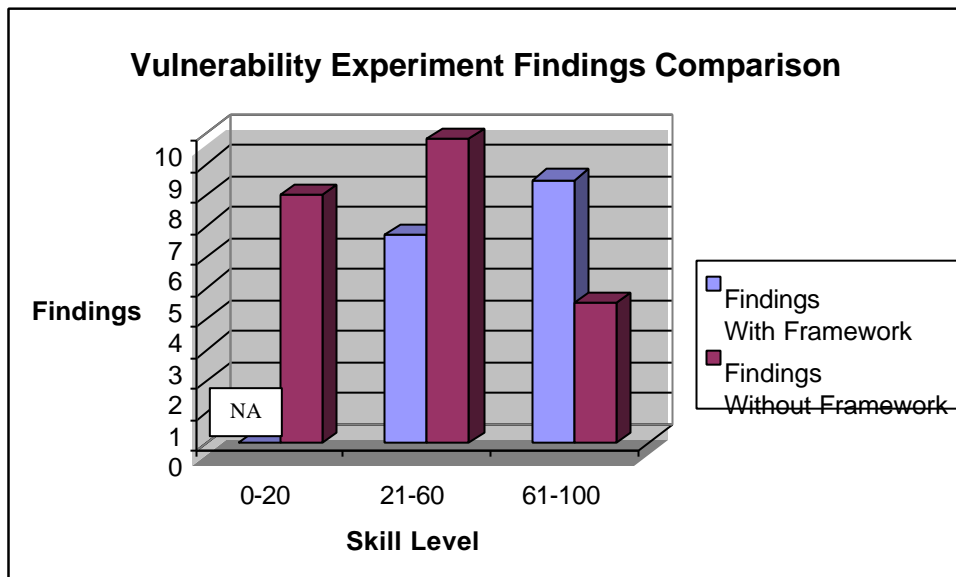
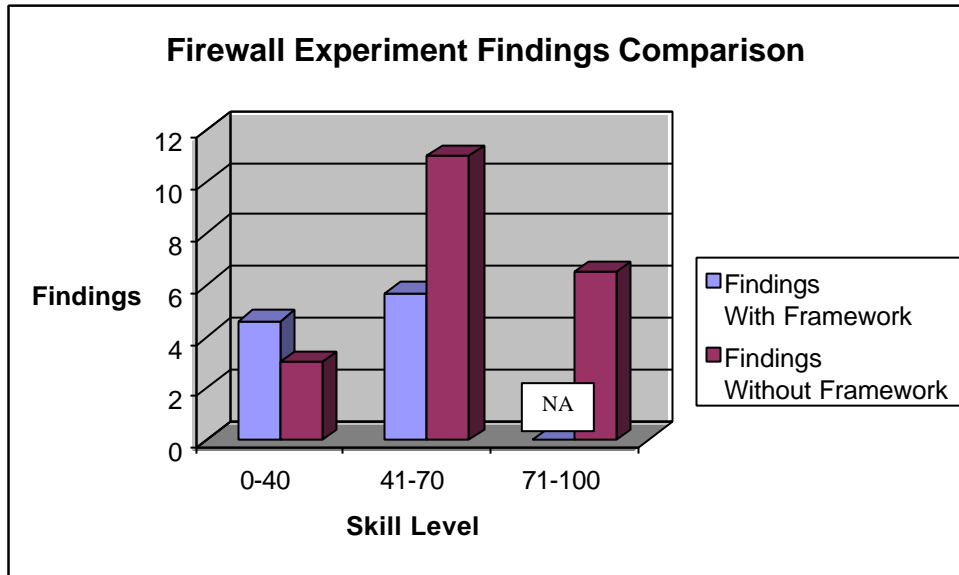


Figure 4.9 A Graphed Comparison of Firewall Experiment Findings



#### Scatter Diagrams

The next visual format of the data was to put all the findings into a large scatter plot to see if there were any tendencies and/or trends across skill levels. (Note that these scatter plots are not using linear regression as a means of finding trend lines, they are a simple plot of skill level against findings with the trend line approximated.) The scatter plots and trends lines are as follows:

**Figure 4.10 Scatter Diagram of Vulnerability Findings; Across Skill Levels and Frameworks**

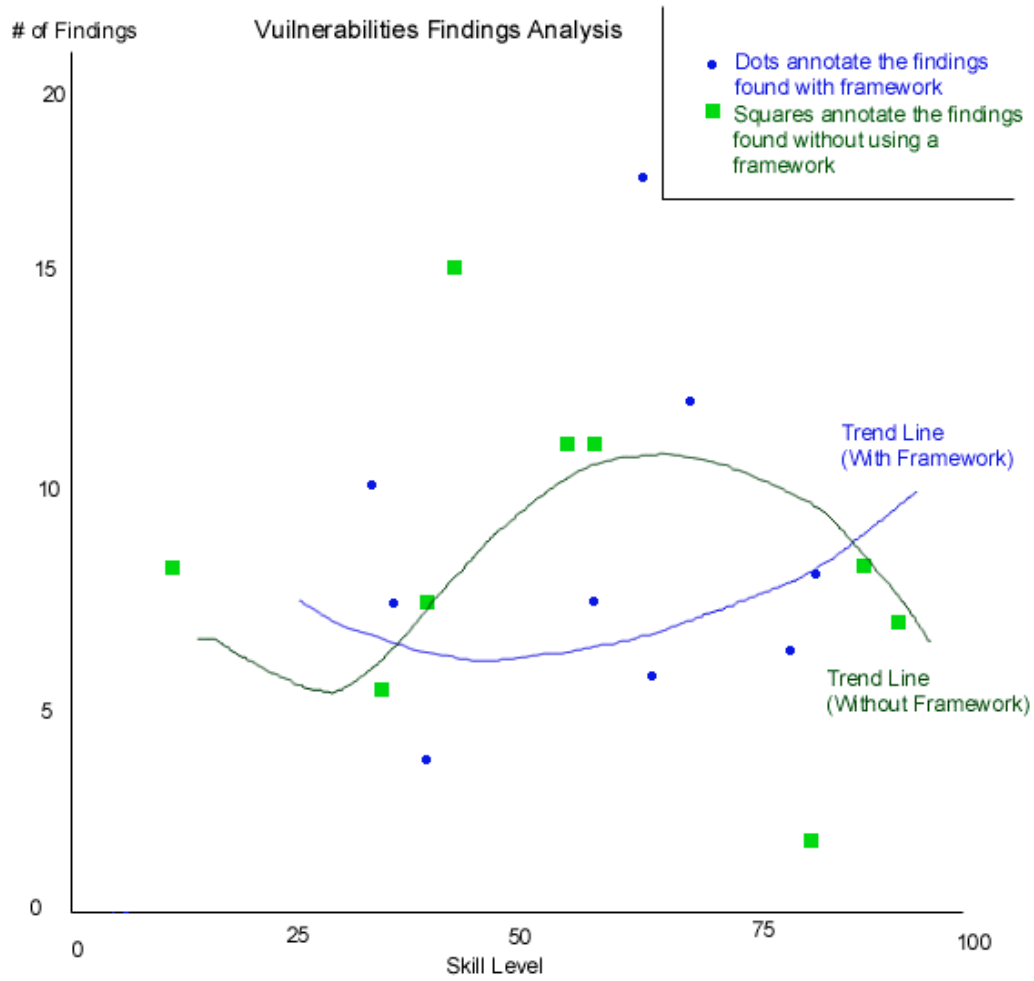
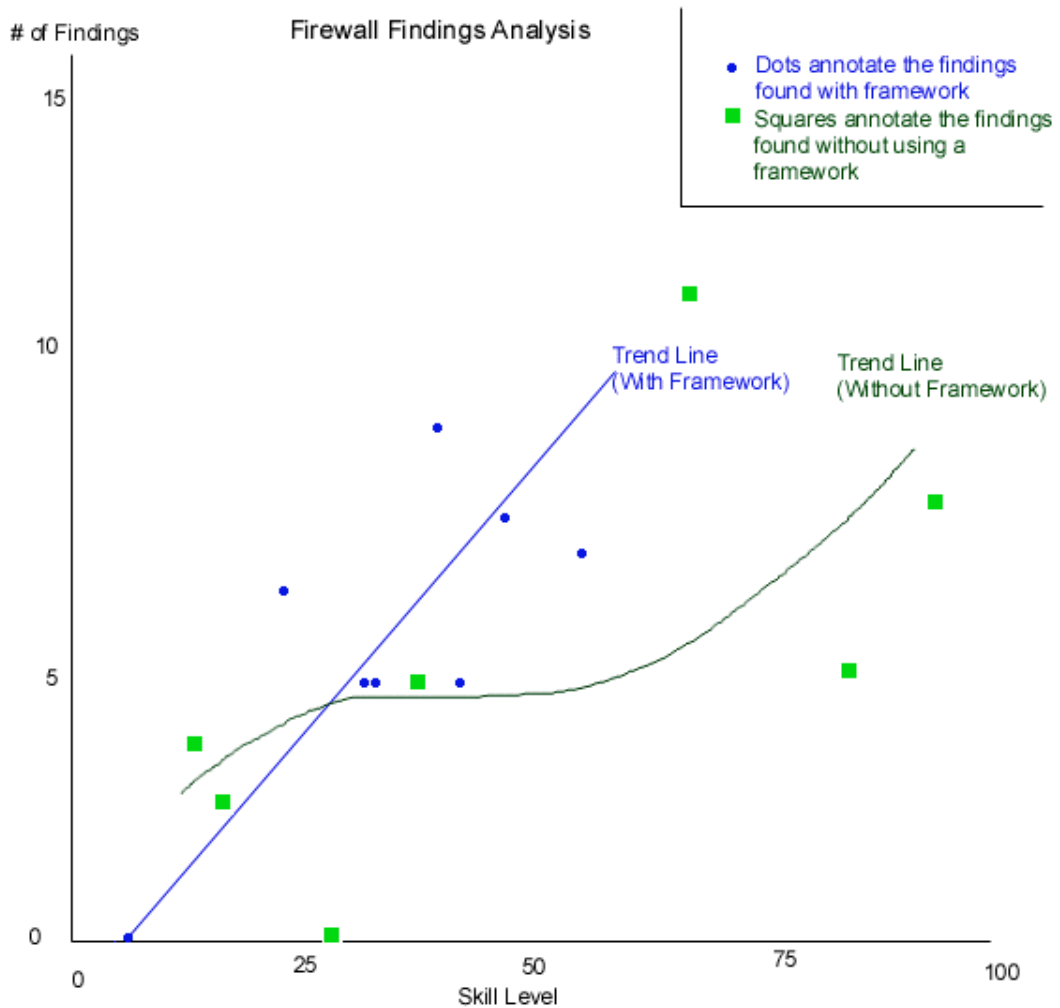
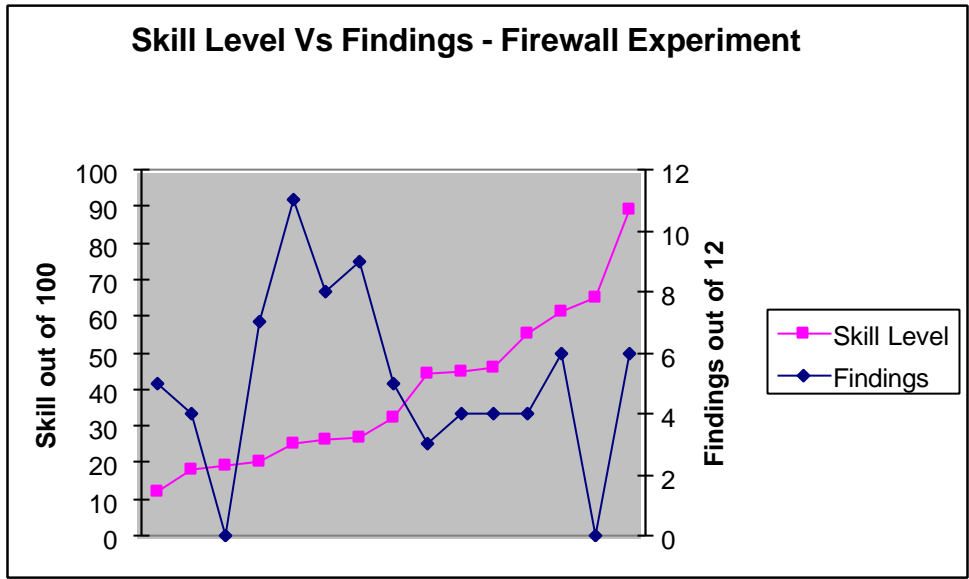


Figure 4.11 Scatter Diagram of Firewall Findings; Across Skill Levels and Frameworks



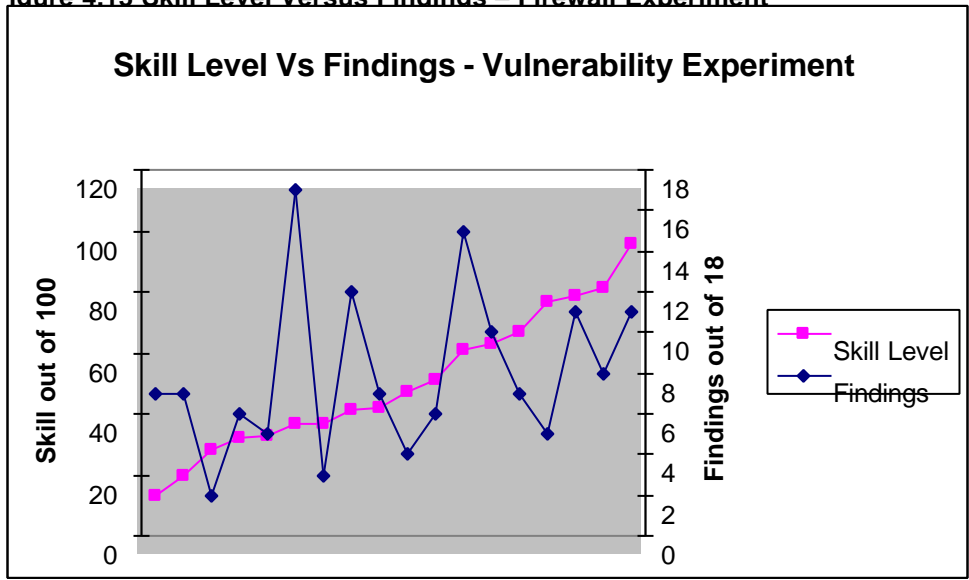
In Chapter 5 there will be more discussion about the surprising observation that the upper level analysts typically do worse than any other level of analyst. Accordingly, the data will be viewed from a findings versus skill level view, taking out the framework variable altogether. This type of graph begins to expose that as skill levels increase, findings tend to go down. This possibly stems from an inadequate skills rating and again, this matter will be discussed in Chapter 5.

**Figure 4.12 Skill Level Versus Findings – Firewall Experiment**



As you can see from the above figure 4.12, the general trend is not for the most skilled analysts to have the most findings. This next figure 4.13 shows the same data but for the vulnerability experiment reinforcing the fact that the upper level analysts were not the best performers. Further when this vulnerability data is correlated simply by findings and skill level, the numbers seem chaotic and random. This is theorized to be a flaw in the skill analysis portion of the experiment. More discussion follows in chapter 5.

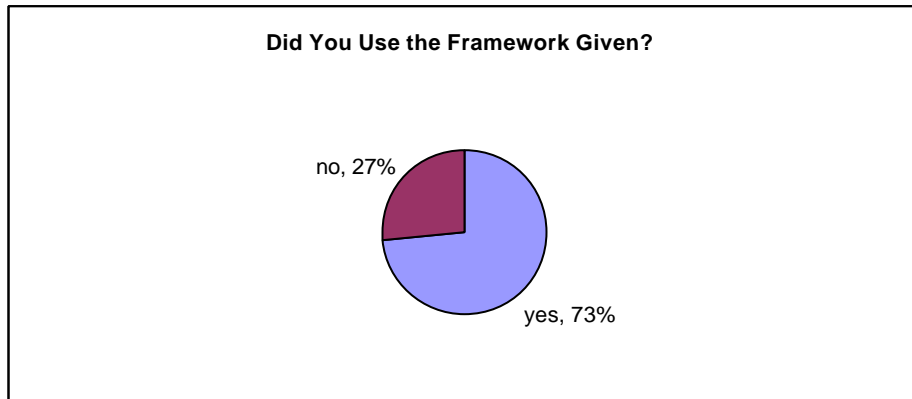
**Figure 4.13 Skill Level Versus Findings – Firewall Experiment**



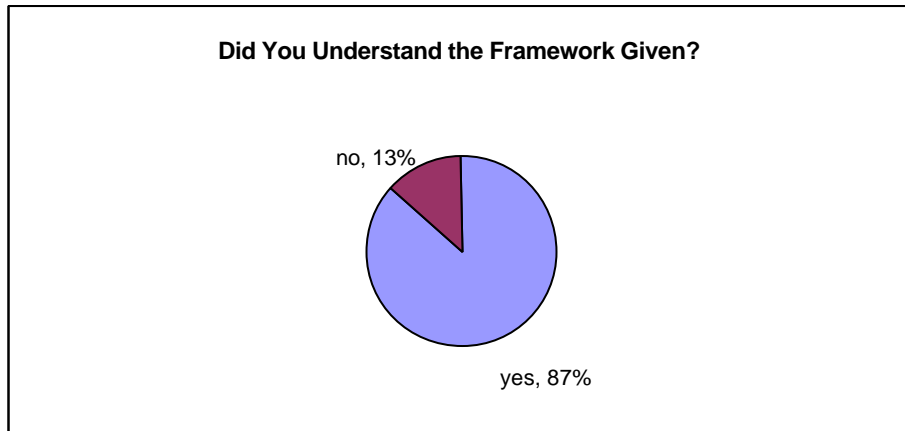
## QUALITATIVE RESULTS

After the experiment is completed, participants who used the framework are asked many questions about the usefulness of the framework and the potential usefulness of the framework. Here are the summaries of these questions:

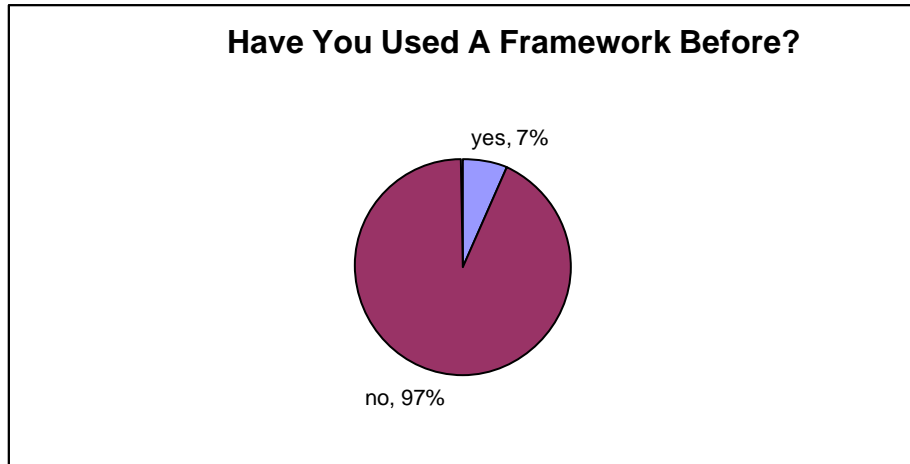
**Figure 4.14 How Many Participants Used the Framework Given**



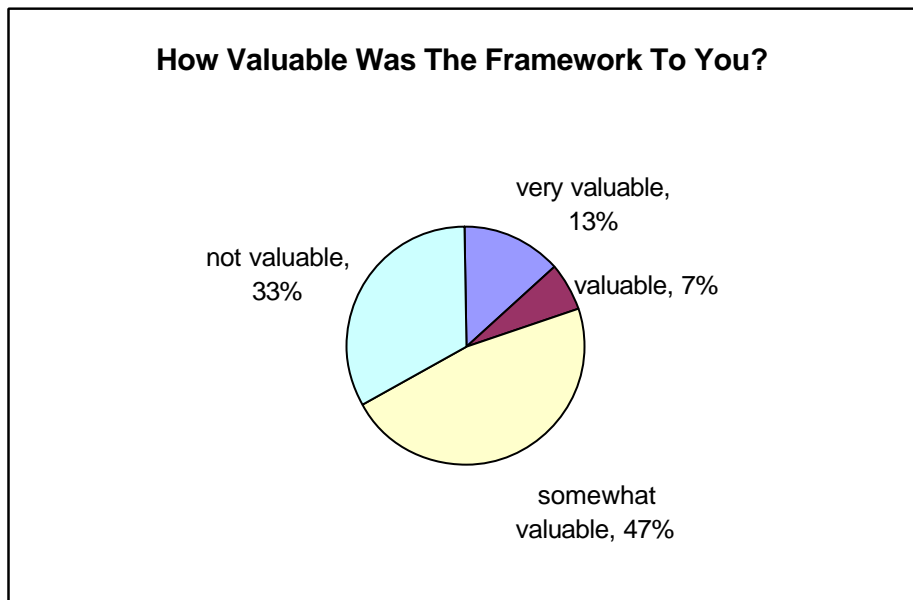
**Figure 4.15 How Many Participants Understood the Framework Given**



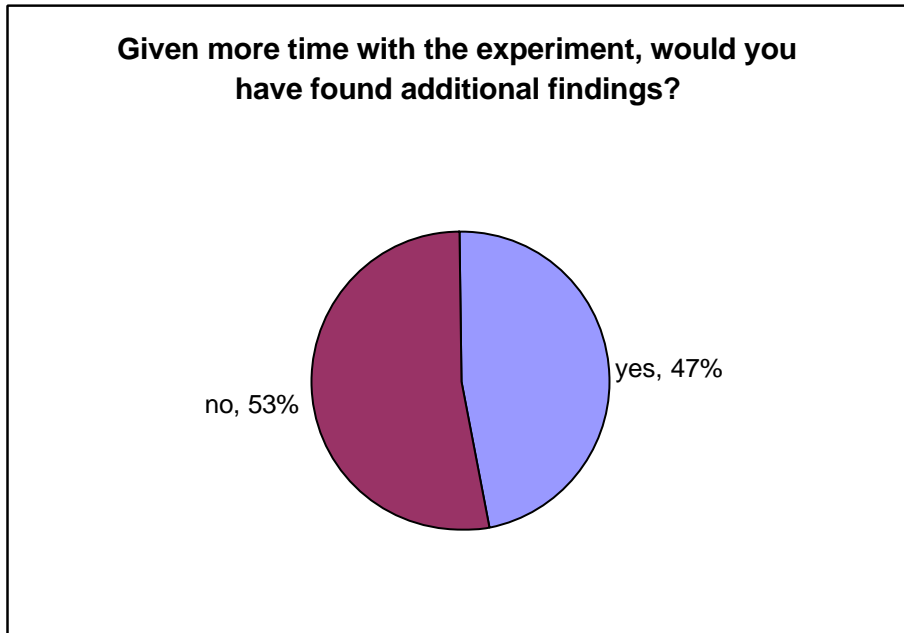
**Figure 4.16 How Many Participants Had Used a Framework Before**



**Figure 4.17 How Valuable Framework Was to Participants During Experiment**

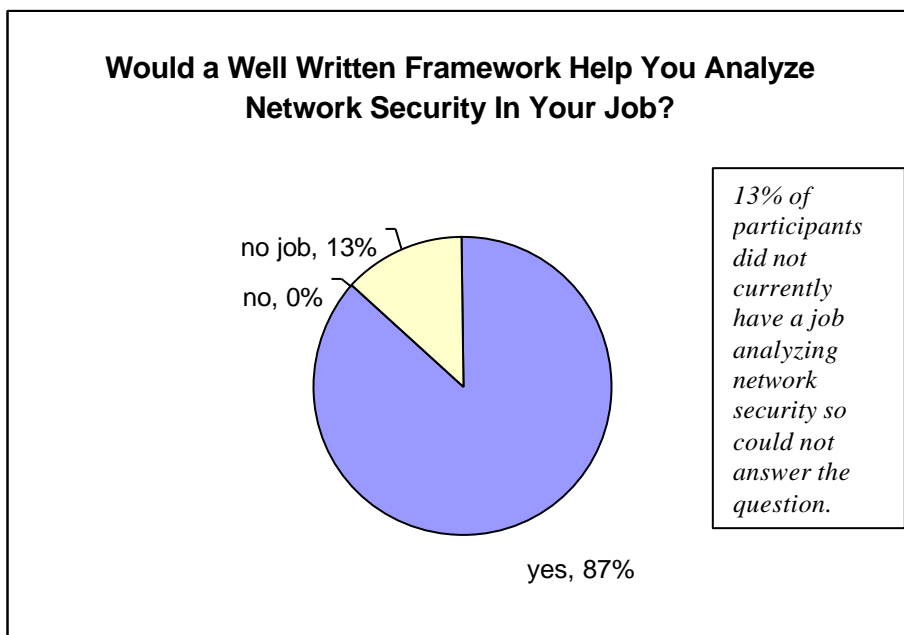


**Figure 4.18 How Many Participants Would Have Liked More Time**



\*interestingly, this question was answered significantly different for the different experiments, those who took the vulnerability experiment stated 63% would have found more given extra time more time whereas with the firewall experiment, only 29% said they would have found more.

**Figure 4.19 How Many Participants Believe a Framework Could Be Useful**



## IDS STUDY RESULTS

The top fifty attacks from a corporate IDS monitoring center were used to complement the previously stated experiment findings. IDS data was collected and sorted and the top fifty attack signatures are shown in Table 4.5.

**Table 4.5 Top 50 IDS Attacks Signatures for September and October of 2003**

SIGNATURE	Count	Rank
=====j		
Nachi_Worm_ICMP_Echo_Request	37987441	1
IP_Localhost_Source_Spoof	10729724	2
ICMP_Sweep_Echo	7777031	3
MS_SQL_Control_Overflow	6579249	4
ICMP_Flood	4039391	5
ICMP_Smurf	1514686	6
Windows_SMB_RPC_NoOp_Sled	1170358	7
Windows_RPC_DCOM_Overflow	1093055	8
Windows_Registry_Access	654192	9
SNMP_Protocol_Violation	529498	10
Limewire_File_Request	436989	11
IIS_CGI_Double_Decode	436318	12
Q_Mail_Length_Crash	426050	13
General_Loki	394245	14
WWW_WinNT_cmd_DOT_exe_Access	375583	15
IIS_DotDot_Crash_Bug	290860	16
Windows_RPC_DCOM_Overflow	255897	17
IP_Fragment_Overwrite_Data_is_Overwritten	246801	18
Jolt2_Fragment_Reassembly_DoS_attack	218285	19
Gnutella_Server_Reply	169128	20
Gnutella_Client_Request	166746	21
IIS_DotDot_EXECUTE_Bug	163812	22
Windows_RPCSS_Overflow_2	153825	23
TCP_SYN_Port_Sweep	144830	24
IP_Fragments_overlap	132754	25
URL_with_XSS	117839	26
Lotus_Domino_database_DoS	103220	27
DNS_Zone_Transfer_High_Port	83379	28
WWW_General_cgi_bin_Attack	77694	29
WWW_IIS_Internet_Printing_Overflow	70664	30
WWW_IIS_Unicode_Attack	68259	31
FetchMail_Arbitrary_Code_Execution	62101	32
Route_Up	54390	33
Orphaned_Fin_Packet	50796	34
Ident_Improper_Request	47311	35

Sendmail_Invalid_Recipient	47286	36
IIS_CGI_Double_Decode	46896	37
Root_exe_access	38332	38
IP_Fragment_Attack	36474	39
Missed_Packet_Count	34848	40
KaZaA_v2_UDP_Client_Probe	31949	41
BadBlue_File_Disclosure	28984	42
Route_Down	27625	43
UDP_Bomb	26911	44
Cisco_Securce_ACS_Directory_Traversal	26408	45
UDP_Flood	25898	46
SMTP_AUTH_Brute_Force_Attempt	24998	47
IIS_CGI_Double_Decode	22957	48
IOS_HTTP_Unauth_Command_Execution	21618	49
Back_Orifice_Ping	21391	50

The results of whether each attack is covered by the framework are shown in Table 4.6.

**Table 4.6 IDS Signatures Cross Referenced with Open Source Framework**

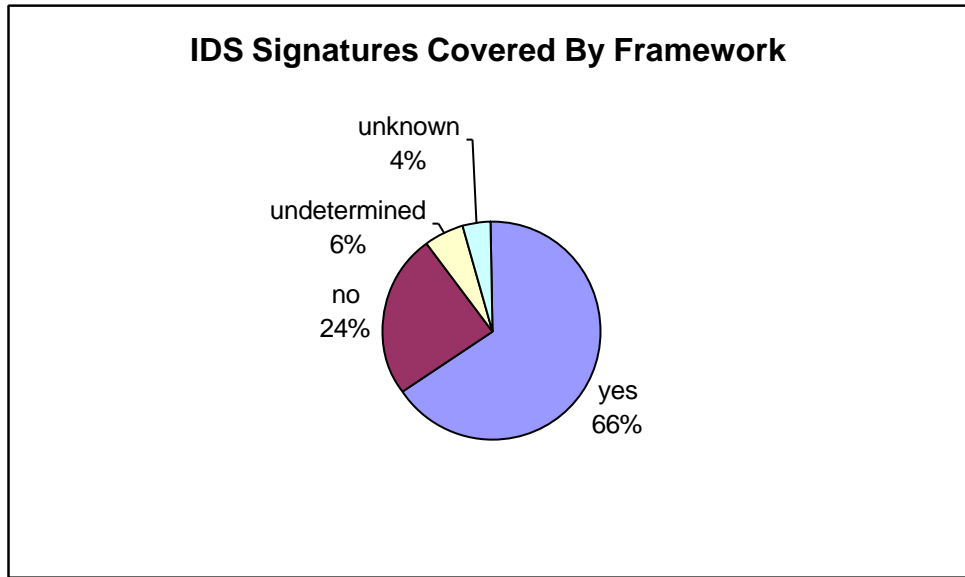
SIGNATURE	Covered In Framework?
=====j	
Nachi_Worm_ICMP_Echo_Request	yes
IP_Localhost_Source_Spoof	yes
ICMP_Sweep_Echo	maybe
MS_SQL_Control_Overflow	yes
ICMP_Flood	maybe
ICMP_Smurf	maybe
Windows_SMB_RPC_NoOp_Sled	yes
Windows_RPC_DCOM_Overflow	yes
Windows_Registry_Access	unknown
SNMP_Protocol_Violation	no
Limewire_File_Request	no
IIS_CGI_Double_Decode	yes
Q_Mail_Length_Crash	no
General_Loki	yes
WWW_WinNT_cmd_DOT_exe_Access	yes
IIS_DotDot_Crash_Bug	yes
Windows_RPC_DCOM_Overflow	yes
IP_Fragment_Overwrite_Data_is_Overwritten	yes
Jolt2_Fragment_Reassembly_DoS_attack	no
Gnutella_Server_Reply	no
Gnutella_Client_Request	no
IIS_DotDot_EXECUTE_Bug	yes
Windows_RPCSS_Overflow_2	yes
TCP_SYN_Port_Sweep	yes
IP_Fragments_overlap	yes

URL_with_XSS	yes
Lotus_Domino_database_DoS	yes
DNS_Zone_Transfer_High_Port	yes
WWW_General_cgi_bin_Attack	yes
WWW_IIS_Internet_Printing_Overflow	yes
WWW_IIS_Unicode_Attack	yes
FetchMail_Arbitrary_Code_Execution	no
Route_Up	unknown
Orphaned_Fin_Packet	yes
Ident_Improper_Request	no
Sendmail_Invalid_Recipient	no
IIS_CGI_Double_Decode	yes
Root_exe_access	yes
IP_Fragment_Attack	yes
Missed_Packet_Count	no
KaZaA_v2_UDP_Client_Probe	no
BadBlue_File_Disclosure	yes
Route_Down	unknown
UDP_Bomb	yes
Cisco_Securce_ACS_Directory_Traversal	yes
UDP_Flood	yes
SMTP_AUTH_Brute_Force_Attempt	no
IIS_CGI_Double_Decode	yes
IOS_HTTP_Unauth_Command_Execution	yes
Back_Orifice_Ping	yes

The comments for each of these cross referenced sections resides in Appendix O. This appendix talks in details about why each signature was identified as being covered or not covered in the framework and which part of the framework each signature is covered under.

Total number of signatures covered by frameworks is shown in figure 4.20.

**Figure 4.20 Total Number of Signatures Covered By Framework**



All signatures are analyzed to see if they are covered in the framework and if so, which part of the framework covers them. Of the total signatures covered, most fell into one of three categories; Vulnerability Assessment and Research, Access Control Testing and Internet Application Testing. This is interesting and useful because the two tests used in the experiment were Vulnerability Assessment and Research and Access Control Testing. On average, the framework is identified as theoretically able to cover between 66% and 76% of the findings and on average, the analysts who used the framework are finding about 33% of the findings.

### **Conclusion**

Chapter 4 material presents the results of all data collected in an uninterpreted form. Chapter 5 will proceed to translate these finding to understand if they help answer the question of framework effectiveness across skill levels.

## **CHAPTER 5**

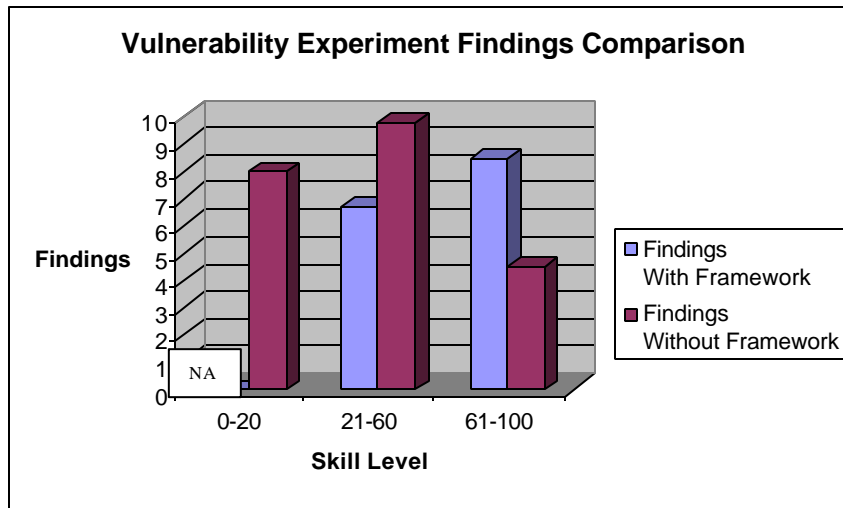
### **DISCUSSION OF RESULTS**

Chapter 4 gave the results of data collection without attempting any interpretation. Chapter 5 will make judgments and give analysis according to the data received during the study. This section will discuss framework effectiveness from both a quantitative point of view and qualitative point of view, while offering a criticism of methods and citing overall trends.

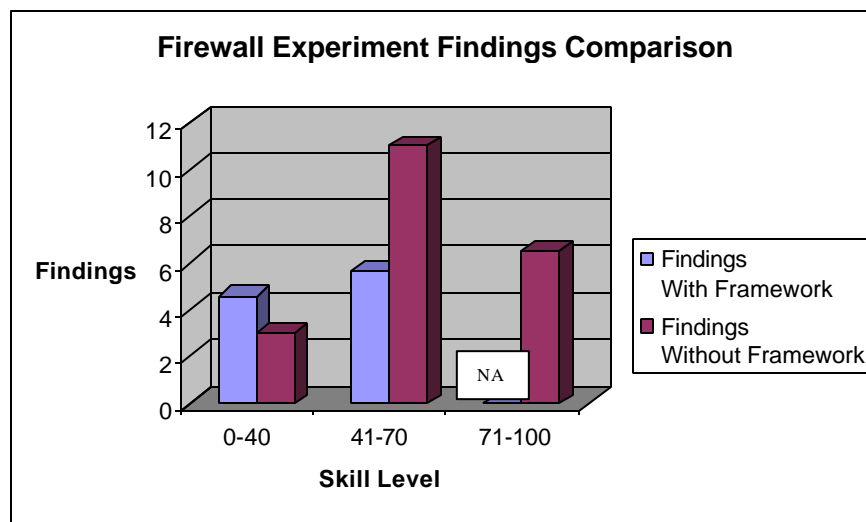
### **QUANTITATIVE ANALYSIS**

The qualitative results expose a few interesting trends and tendencies though none of them distinguish the frameworks as significantly increasing the ability of analysts to perform security. In fact, the general result is that the framework was hurtful to analysts. The first analysis of the data looks at simple averages of findings across skills. From this perspective, it appears that frameworks tend to be hurtful to all skill levels, both for firewall and vulnerability analysts. The only group of analysts that measures to be helped by the framework is the expert level vulnerability analysts. See Figures 5.1 and 5.2 below.

**Figure 5.1 A Graphed Comparison of Vulnerability Experiment Findings**



**Figure 5.2 A Graphed Comparison of Firewall Experiment Findings**



In fact, the idea that frameworks cause detriment to security analysis is reinforced when looking at the average findings for all participants without separating them by skill levels. Collapsing the data and removing the skill level variable shows that the group as a whole performed better without using the framework. However, it should be noted that the group skill averages as a whole were slightly higher for the framework participants. This means that the slight decrease in findings closely correlates with the slight decrease in skill level on average. Regardless of small intricacies, this quantitative data, viewed as simple

statistic averages, shows that the framework as utilized during this experiment was not effective but was actually slightly hurtful to participants.

These simple averages describe the data trends clearly and mathematically but they do not capture the heart of the data because of the limited number of participants and the range of skill levels involved. The averages wash out unique trends in the data by normalizing the group as a whole. While it is useful to see this normalized data, it will also be beneficial to have a picture of the data in a more raw form. The second analysis of the data, a scatter plot, is from a visual perspective so as not to lose the individual perspective. This analysis is completed by plotting findings and skill levels on a diagram to see if any tendencies or trends can be identified. This data is located in Figures 4.10 and 4.11. This way of looking at the data sheds additional light on the answer to the question.

The graphs reinforce that the framework is again hurtful to analysts during vulnerability testing. The only exception to this statement is the upper level analysts. They appear to be helped by the framework. This observation is interesting; one would assume that an expert level security professional with tremendous technical skills would have little difficulty analyzing vulnerabilities. However the results from this experiment reveal that these analysts are most helped by a framework.

In the case of the firewall test, the scatter plot results disagreed with the mathematical average results. The firewall plot indicates the framework is useful for all skill levels; it shows a trend line of increased findings if the participant uses a framework to solve the firewall security problem. It appears that the framework, when viewed with actual skill levels, and not averages, is slightly helpful (helpful by approximately 2 findings per analyst). 2 findings out of 15 total (in the firewall experiment) equates to a 13% increase in findings. This is a significant number but due to the variance in findings data, this trend line does not supply enough confidence to guarantee this number. Another notable trend is that the framework, in the firewall test, shows indications of helping the most experienced analysts more than other skill levels. This trait is consistent with the vulnerability findings.

When the firewall analysis is considered from different angles, it shows conflicting results. Looking at mathematical averages, it appears hurtful but looking at trend lines on a scatter plot it appears helpful. Part of the problem in the discrepancy amongst analysis is due to only having forty participants. A more clear conclusion would be drawn with 100 analysts. However, with the data produced from the forty participants, it can confidently be concluded that the framework has potential to be helpful, but not to an extent that would merit a sense of urgency in responding to this conclusion of the study.

One interesting observation is that the upper level analysts show tendencies to be discover the least amount of findings in a given security problem. The author's prediction was this skill level would perform the best. What is the possible explanation? One answer was that these analysts are not familiar with analyzing security in such a high level way. These are the folks that are usually coding exploits rather than analyzing vulnerabilities on a server. This leads to the conclusion that perhaps the way skill level was derived in the analyst survey was not an appropriate way to gauge skill. The author attempted to gauge skill in a way that would put very elite hackers at one end of the spectrum and newbies at the other end of the spectrum. From the data output, this was probably not the best way to categorize skill as the elite hackers were certainly not the most qualified for analyzing security from a vulnerability analysis point of view. It would be very interesting to do retro analysis on the data to understand what traits enabled analyst to perform well in the study so that a new skill level questionnaire could be formed.

The next observation is that the mid level analysts were the best analysts, far better than the upper level analysts. However, these analysts were hurt tremendously by the framework. As these analysts appear to be the ones who are most effective, they are the target audience for the framework. The framework is currently failing this target audience.

The overall trend of quantitative analysis indicates the framework used during the experiment was hurtful to most and only effective for participants with the top skill levels. This is interesting but it does not help solve the problem of the gap between demand for and

supply of quality network security analysts. In order to bridge the gap, the framework needs to be effective for the mid and low range analysts.

## **QUALITATIVE ANALYSIS**

After participants completed the experiment, they were asked some questions about the framework used and about frameworks in general. A few important findings emerge out of the analysis of this questionnaire. Firstly, 100% of participants who regularly perform network security analysis daily said that a framework would be helpful to them in their job. Furthermore, 97% of participants said that they have never used a framework before. Putting the two together it appears that, currently, frameworks are not widely used but the perception (even after a test in which majority opinion was that the framework was not very helpful) is that a framework could be of value. These two findings will drive many of the conclusions of the study because they underscore the importance of and demand for frameworks.

### *Criticisms*

In the post-experiment questionnaire, there is a chance for the participants to voice opinions, concerns, and comments about the experiment. The majority of the feedback was negative feedback relating to the framework given. Participants as a whole felt that this framework was not useful to their skill level. Experienced participants felt the framework would be good for beginners and beginners felt the framework would be good for more experienced testers. Nobody felt that they were of adequate skill level to properly utilize the framework. There was also frustration at the experiment. About 30% of the participants felt that they could have used more time. Some who felt they could have used more time expressed concern at being able to perform any type of security analysis in such a short period of time. There was much disgust for the framework given. Two participants indicated that there was no way this type of framework could help so they did not even try to use it.

### *Commendations*

Although mostly negative, there was some positive feedback about the framework. These comments generally fell in the line of, “the framework did not necessarily help me find anything, but it reminded me of a few areas I may have forgotten to look at.” This statement sums up the current state of frameworks in general; the existing industry’s best practice frameworks tend to be a checklist that would serve to remind an analyst to inspect certain areas of a network but are not tools providing any further insight as to what is important when checking these areas. For example, in the firewall test, many participants noted they would have liked further detail on what specifically to look for in a firewall access control list. Overall there was positive feedback that a framework has the potential to be very helpful in the workplace.

### *Surprises*

The practitioner of the experiment predicted a few things that did not come to fruition. In the firewall experiment, the framework specifically mentions a few checks to be made in regards to looking at firewall rules. Three specific checks are: a deny all rule at the end of the rule set, antispoof rules, and egress filtering. The practitioner predicted that participants using the framework would automatically find these findings in the rule set because they were specifically pointed out in the framework, were straightforward, and easy to understand. However, there was statistically no difference between those using the framework and those not using the framework when finding these, even though the practitioner thought of these as easy and obvious findings. So even though a checklist is useful, there is no guarantee that, even if understood, it would be effective in helping an analyst catch security weaknesses.

The overall feel from the participants in the post survey comments is that this experiment is not be able to judge with any certainty the effectiveness of frameworks because the framework given was not useful. This is interpreted to mean participants feel that the industry best practice framework is not useful as written. There could be two reasons

for this: the framework is simply not useful, or the experiment was set up inadequately and did not enable to analysts to utilize the framework.

The more reasonable answer lies in the middle: the specific framework used and security frameworks in general are not intended to help analysts find bugs, they are intended to provide a comprehensive, measurable, and repeatable way of analyzing security. However, because of the way the framework is written, it is perceived to be ineffectual to the users within this experiment, even those with great experience in the field.

Another surprise is that, on average, analysts found less than half of the vulnerabilities on any given system. Also interesting is that the level 3 participants were significantly worse at finding vulnerabilities than their level 2 counterparts. (See Table 5.1) While this does not answer the question of network effectiveness, it is interesting to discover that skill level is not necessarily the major factor in a comprehensive security analysis.

**Table 5.1 Percent of Findings Found By Skill Level**

% of Total Findings Found				
	Firewall Experiment		Vulnerability Experiment	
	With Framework	Without Framework	With Framework	Without Framework
Level 1	na	40.0%	30.6%	20.0%
Level 2	33.5%	49.0%	38.0%	73.3%
Level 3	42.5%	22.5%	Na	43.3%

The qualitative analysis broadens the understanding of the quantitative data. With just quantitative data, it appears that frameworks utilized in this study were ineffective but when combined with qualitative data, it is more clear that though the frameworks were ineffective in this study, it does not mean frameworks are and will be ineffective in their entirety.

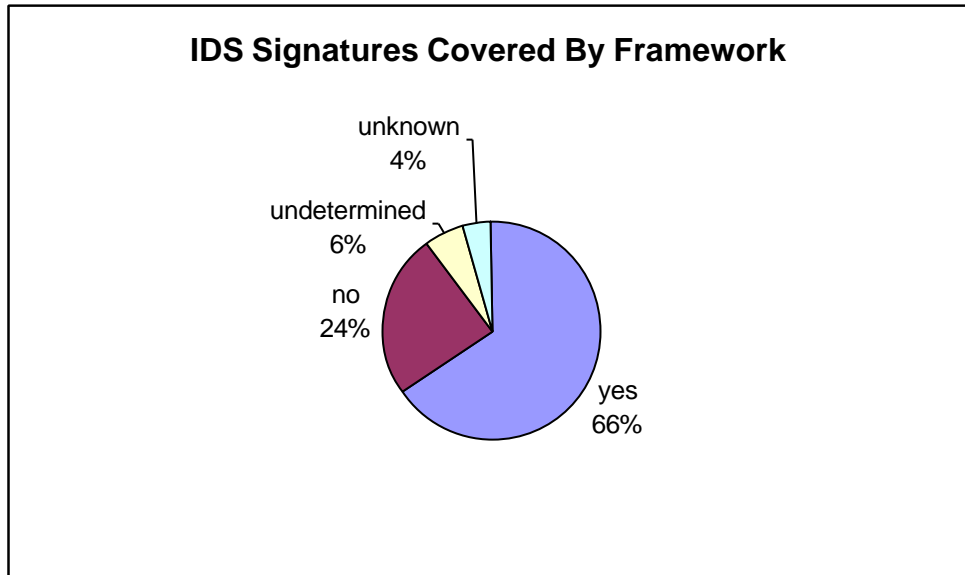
## IDS ANALYSIS

The quantitative data shows practically how many findings an analyst discovered using the given framework. IDS analysis will show theoretically how many findings the

framework can cover. The average amount of findings actually uncovered by participants in the experiment is less than half of the findings in the environment, but the results of the IDS data show that the framework theoretically covers 66-76% of all IDS attacks. However, this number is misleading because there are certain attacks that are essentially unavoidable. For example certain denial of service attacks are impossible to mitigate; it is simply a matter of bandwidth. Previously, the author mentioned that IDS data only shows one piece of network security attacks: the ones coming across the wire and visible by sensors. However, these are pertinent, active, measurable, and quantifiable attacks which is why they are used in this study. The framework covers 66-76% of these signatures and those defined as not being covered by the framework all fall under the umbrella of, "if an IDS was in place, it could be configured to be notified of and possibly block such attacks." Because the framework extensively covers IDS functionality and testing, for all intents and purposes the framework does exhaustively address each of these findings in one way or another. The conclusion is that the framework is theoretically effective in catching all attacks that an IDS sensor would flag. To play devil's advocate, the reader is reminded that in the IDS study, the author made assumptions when identifying if the framework covers certain signatures. Some of these assumptions were quite broad. The author had to repeatedly assume that an expertly qualified analyst with a good understanding of attacks and threats was analyzing security without flaw.

This IDS data complements the experiment data because it reinforces the potential of the framework while also exposing the usability and analyst skill level issue. If a framework can theoretically cover every type of exposure but practically analysts miss exposures when using it, the framework has room for improvement and the analysts have the need for education.

**Figure 5.3: IDS Signatures Covered By Framework**



#### **OVERALL TRENDS**

What does this analysis of quantitative, qualitative and IDS data mean for the framework? It means the hands on road test showed that the low and mid skilled users are not currently able to use this framework as a useful tool for analyzing security.

Does this mean frameworks are ineffectual as a whole? Absolutely not. This study also showed that the frameworks are desired and regarded as potentially very useful by analysts. It also showed that, quantitatively, the framework showed tendencies to help by as much as 13%. The intrusion detection analysis reinforced that, in theory, the framework can cover the majority of intrusions.

## CHAPTER 6

### CONCLUSIONS

The goal of this study has been to examine a possible method of understanding effectiveness of network security frameworks across skill levels by attempting an experiment whereby analysts use a network security framework to solve a given problem. This study was unable to prove with any level of distinction that frameworks are effective in helping analysts solve network security problems. Further, the majority of data collected indicates the framework was hurtful to analysts. This is surprising data that contradicts the original hypothesis that frameworks are helpful to network security analysts. The experiment was critiqued to understand if the negative result could be false. This critique exposed three special cases that need to be considered. Firstly, the experiment was limited in resources handicapping the ability of the data to fully answer the question. Secondly, it is possible that the framework tested was not a proper choice for the given experiment. Lastly there was a very interesting observation suggesting that the method for ranking skill level was flawed. These special cases indicate there is room for further exploration in this study.

The case for further exploration is underscored by the results of the literature review and qualitative research findings. Notable information compiled suggests very strongly that there is potential for frameworks to become very useful and that analysts desire a helpful framework. Currently the effectiveness is limited and is perceived by users to be inadequate. In sum, there is a demand for this type of tool and, while it is not perfected yet, further development towards an effective framework will help bridge the gap of the demand for security and lack of analysts who can support security infrastructure.

## **Future Research**

There were several extremely interesting observations during the study that merit further research. The first idea that rolls of the list is to take this same experimental approach from a new top down paradigm. That is to say, during this thesis, the analysis centered on bug hunting which is a bottom level approach to security. As security is only effectual when pushed from the top down, it would be interesting to take the approach of using a networked organization (mock or real), having dedicated security analysts spend a large amount of time analyzing security from every aspect, and composing findings from a policy point of view. It would be interesting to understand if the analysts could translate bugs into upper level policy recommendations that would be pushed down to fix the environment.

Another future research idea involves a new skill level analysis. The most interesting observation to the author during this study was that the upper echelon of analysts were the most ineffective analysts. This is a good indication that the skill level analysis performed in the study was not appropriate. The author had a presumption that the elite analysts are those that can code zero day exploits and ingeniously develop new ways to attack systems and while these analysts are immensely talented and priceless in the security arena, they might not be best choice for analyzing network security from a holistic point of view. It would be very useful to do retro analysis of the experiment data to understand which skill traits correlate with the most number of findings. This would then create a new profile of upper echelon and help understand which types of skills are best suited to vulnerability analysis, firewall analysis, or whatever discipline the experiment encompassed.

The last area of research involves specifically understanding which tools are the best enablers for analysts. A framework is one of many tools an analyst can use as a guide for analyzing security. A useful study would be to compare results using different tools to identify if one tool emerges as more effective for a particular group.

Many interesting observations emerged from the study. This was the first attempt at analyzing security and there is still much work to be done in understanding the effectiveness of frameworks across skill levels.

## BIBLIOGRAPHY

- Alberts, Christopher J. and Audrey J. Dorofee, OCTAVE Criteria, Version 2.0. December 2001
- Anderson David, Dennis J. Sweeney and Thomas A. Williams. Essentials of Statistics for Business and Economics. West Publishing Company, 1997
- Convery, Sean and Bernie Trudell, SAFE: A Security Blueprint for Enterprise Networks. 2000
- Frost and Sullivan, Tapping the Potential of Managed Security Services: Opportunities for European Operators and System Integrators. 2002
- Fielding, N. and J. Fielding, Linking Data. Sage Publications, 1986
- Fraser, B. ed. RFC 2196: Site Security Handbook. September 1997
- Gordon, Sharon. "Virus Damage Worst on Record for August." Datamation 2 September 2003. [itmanagement.earthweb.com/secu/print.php/3071051](http://itmanagement.earthweb.com/secu/print.php/3071051)
- Graft, Donald, Mohnish Pabrai and Uday Pabrai, "Methodology for Network Security Design." IEEE Communications Magazine, November 1990
- Green, J, V. Caracelli, and W. F. Graham, "Toward a Conceptual Framework for Mixed-Method Evaluation Designs," Education Evaluation and Policy Analysis. 11(i.3). pp. 255-274
- Hatch, Brian, James Lee, and George Kurtz. Hacking Linux Exposed: Linux Security Secrets and Solutions, Osborne/McGraw-Hill, 2001.
- Hatch, Brian, and James Lee. Hacking Exposed Linux (2<sup>nd</sup> Edition). Osborne/McGraw-Hill 2002.
- Havery, J. "The LTDI Evaluation Cookbook," Glasgow: Learning Technology Dissemination Initiative. 1998
- Herzog, Pete, Open-Source Security Testing Methodology Manual. 2002
- McClure, Stuart, et al. Hacking Exposed: Network Security Secrets and Solutions (4th Edition), Osborne/McGraw-Hill, 2003
- Moore, Gordon E. Cramming More Components Onto Integrated Circuits. 1965
- "Network." Merriam Webster Dictionary: <http://www.m-w.com/cgi-bin/dictionary>. 2003
- "Network." Webopedia Dictionary: <http://www.webopedia.com>. 2003
- Oliver, Martin, "An Introduction to the Evaluation of Learning Technology." Educational Technical & Society. 3(4) 2000 [http://ifets.ieee.org/periodical/vol\\_4\\_2000/intro.html](http://ifets.ieee.org/periodical/vol_4_2000/intro.html)
- Oliver, M. "The ELT Toolkit,"1999 <http://www.unl.ac.uk/tltc/elt/toolkit.pdf>
- Pierson, Lyndon G., and Edward Witzke. "A Security Methodology for Computer Networks." AT&T Technical Journal 67 1998

Risjord, Mark, Margret Moloney, and Sandra Dunbar, "Methodological Triangulation in Nursing Research," Philosophy of the Social Sciences, 31(1), March 2001, pp. 40-59.

Scambray, Joel, and Mike Shema. Hacking Exposed Web Applications (1<sup>st</sup> Edition). Osborne/McGraw-Hill 2002.

Schumacher ,H.J. (Jerry) and Sumit Ghosh . "A Fundamental Framework for Network Security Towards Enabling Security on Demand in an ATM Network." Computers & Security 17(6) 1998, pp 527-542.

### **Government Publications**

5200.28-STD. Trusted Computer System Evaluation Criteria (TSEC), U.S. Department of Defense, December 1985 (Orange Book).

CCIMB-99-031. Common Criteria for Information Technology Security Evaluation, Version 2.1 August 1999.

NCSC-TG-005. Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria (TNI), NCSC, July 1987 (Red Book).

National Institute of Standards and Technology, Guidelines on Firewalls. January 2002

National Institute of Standards and Technology, Security Guide for Interconnecting Information Systems Technology. January 2002

National Institute of Standards and Technology, Security Self-Assessment Guide for Information Technology Systems. November 2001

## Appendix A: Pre Experiment Survey (Skills Survey)

### Entrance Survey

---

You are invited to join in an experiment for a University of Colorado graduate student. The nature of the experiment is to understand more about network security frameworks and their usefulness to security analyst across many skill levels.

This information will be kept confidential. You will be identified with a 5 digit number and individual user information will never be published with any identifying personal information.

This is not a test! This information will be used to understand participant skill levels. The study needs participants at all levels of skill, from novice to expert, so please do not over or under value your skills.

### Directions: Circle one answer for each question or fill in the blank

---

1. Do you understand TCP/IP? (Yes or No or "I understand TCP/IP a little bit.")
  - a. yes
  - b. no
  - c. I understand TCP/IP a little bit
2. What protocol uses Port 22?
3. Name a port scanning tool
4. How many addresses exist in a /25 network?
5. Name the three steps in a TCP handshake?
6. Explain, very briefly (10 words or less), the difference between a firewall and a filtering router  

---

---

---
7. How many years have you been involved in computer networks (job or recreational)?  
a. 0-1   b. 2-5   c. 6-10   d. more than 10
8. How many years have you been involved in computer security (job or recreational)?  
a. 0-1   b. 2-5   c. 6-10   d. more than 10
9. How many boxes have you hacked (any type of unauthorized privileges on a box)?  
a. 0-1   b. 2-10   c. 11-20   d. more than 20
10. How many times have you helped architect a secure, networked environment?  
a. 0-1   b. 2-5   c. 6-10   d. more than 10
11. Circle any professional security certifications you currently maintain
  - a. GIAC (which one \_\_\_\_\_)
  - b. CISSP

c. other (please list the name \_\_\_\_\_)

12.

How often do you use the following?	a. Never	b. Yearly	c. Monthly	d. Weekly
Nessus				
nmap				
netcat				
linux				
*BSD				

13. How familiar are you with firewalls?

a. beginner    b. intermediate    c. advanced    d. expert

14. How many times have you reviewed firewall rules?

a. 0    b. 1-5    c. 6-20    d. more than 20

15.

How often do you have exposure to the following? (Working with, building, or reviewing rules for.)	a. Never	b. Yearly	c. Monthly	d. Weekly
PIX				
Firewall-1				
Netfilter (Iptables)				
Proxies				
VPN devices				

16. Do you currently use a formal methodology for analyzing security on a network, server or application?

Yes    No    (If yes, please give name of \_\_\_\_\_)

17. How would you rate your security skill level as relating to reviewing server level vulnerabilities?

a. beginner    b. intermediate    c. advanced    d. expert

18. How would you rate your skill level as relating to reviewing firewall security?

a. beginner    b. intermediate    c. advanced    d. expert

**Appendix B: Post Experiment Survey**

1. Did you use the framework given?
  - a. Yes
  - b. No
  
2. Did you understand the framework given?
  - a. Yes
  - b. No
  
3. Have you used a framework before?
  - a. Yes (If so, which one: \_\_\_\_\_)
  - b. No
  
4. If you have used a framework before, how does this one compare?
  - a. better
  - b. worse
  - c. the same
  - d. I have never used a framework before
  
5. How valuable was framework to you?
  - a. Very valuable (Helped me find 2 or more findings)
  - b. Valuable (Helped me find 1 or more findings)
  - c. Somewhat useful (Probably helped me)
  - d. Not useful (Did not help me find any vulnerabilities.)
  
6. Given more time with the framework, would you have found more findings? (I.e., was time too limited?)
  - a. Yes
  - b. No
  
7. How could this framework be better?  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
  
8. Would a well written framework help you perform better network security analysis in your job?
  - a. Yes
  - b. No
  - c. I do not perform network security analysis in my job

9. Comments about test:

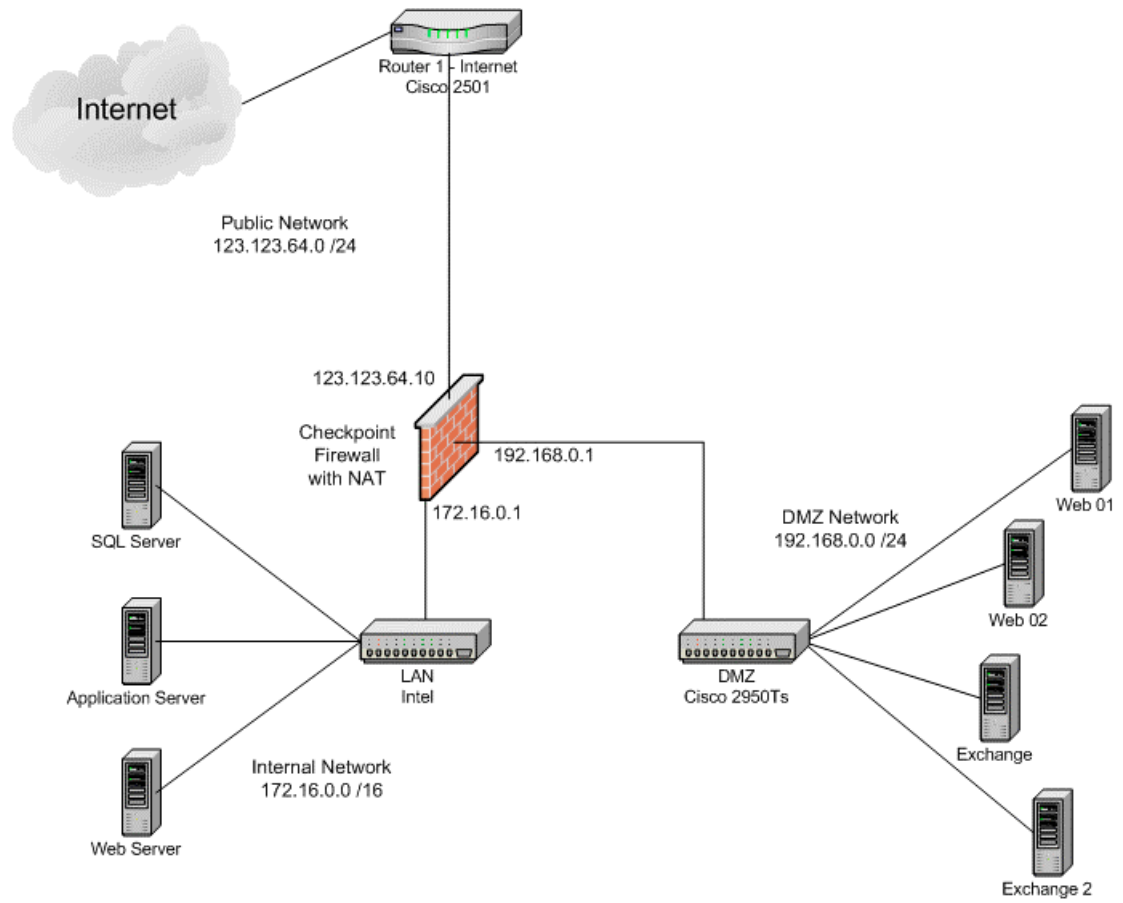
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

## Appendix C: Firewall Test Data: Firewall Rules

Firewall Policy: CompanyA\_Ruleset.W

RULE	SOURCE	DESTINATION	SERVICES	ACTION	TRACK	TIME	INSTALL ON	COMMENTS
1	Internal_LAN net 172.16.0.0 / 16	External_Network net 200.200.200.0 / 24	Any	accept	Long	Any	Gateways	
2	Spam_Domain net 123.123.123.0 / 24	Any	Any	drop	Long	Any	Gateways	This is the IP address for prodigy.net.mx, from which we have been receiving thousands of pieces of Virused spam daily
3	Any	Internal_WebServer host 172.16.1.42	http https	accept		Any	Gateways	
4	Internal_WebServer host 172.16.1.40	Any	http https	accept		Any	Gateways	
5	Any	Any	dns domain-tcp domain-udp	accept	Long	Any	Gateways	
6	Internal_host host 172.16.0.25	Internal_host host 172.16.1.230	Any	drop		Any	Gateways	
7	Internal_host host 172.16.0.230	Internal_host host 172.16.2.25	Any	accept		Any	Gateways	
8	DMZ_Exchange_Server host 192.168.0.2 DMZ_Exchange2_Server host 192.168.0.12	Any	Any	accept	Long	Any	Gateways	
9	Internal_Web_Server host 192.168.0.7	External_FTP_Host host 123.123.123.12	ftp	accept	Long	Any	Gateways	
10	Any	DMZ_Exchange_Server host 192.168.0.2 DMZ_Exchange2_Server host 192.168.0.12	smtp http https ftp	accept	Long	Any	Gateways	To allow OWA on the exchange servers
11	Any	Internal_Host host 172.16.0.204	tcp-high-ports pcANYWHERE pcAny2UDP pcANYWHERE-data pcANYWHERE-logs	accept	Long	Any	Gateways	

## Appendix D: Firewall Test Data: Network Diagram



## Appendix E: Firewall Test Data: Framework

23 August 2003

### 8. Access Control Testing

The firewall controls the flow of traffic between the enterprise network, the DMZ, and the Internet. It operates on a security policy and uses ACL's (Access Control Lists). This module is designed to assure that only that which should be expressly permitted be allowed into the network, all else should be denied. Additionally, the tester is to understand the configuration of the firewall and the mapping it provides through to the servers and services behind it.

Reviewing the server logs is needed to verify the tests performed on the Internet presence especially in cases where results of the tests are not immediately visible to the tester. Many unknowns are left to the analyst who has not reviewed the logs.

<b>Expected Results:</b>	Information on the firewall as a service and a system Information on the features implemented on the firewall Outline of the network security policy by the ACL List of the types of packets which may enter the network List of the types of protocols with access inside the network
	List of live systems found List of packets which entered the network by port number List of protocols which entered the network List of unmonitored paths into the network

#### Firewall and features identification

1. Verify the router type with information collected from intelligence gathering.
2. Verify if the router is providing network address translation (NAT)
3. Verify the penetrations from strategically determined packet TTL settings (Firewalking) completed in the Port Scanning module.

#### Verifying firewall ACL configuration

4. Test the ACL against the written security policy or against the "Deny All" rule.
5. Verify that the firewall is egress filtering local network traffic
6. Verify that the firewall is performing address spoof detection
7. Verify the penetrations from inverse scanning completed in the Port Scanning module.
8. Test the firewall outbound capabilities from the inside.
9. Determine the success of various packet response fingerprinting methods through the firewall
10. Verify the viability of SYN stealth scanning through the firewall for enumeration
11. Measure the use of scanning with specific source ports through the firewall for enumeration
12. Measure the ability of the firewall to handle overlapped fragments such as that used in the TEARDROP attack
13. Measure the ability of the firewall to handle tiny fragmented packets
14. Test the firewall's ability to manage an ongoing series of SYN packets coming in (flooding).
15. Test the firewall's response to packets with the RST flag set.
16. Test the firewall's management of standard UDP packets.
17. Verify the firewall's ability to screen enumeration techniques using ACK packets.
18. Verify the firewall's ability to screen enumeration techniques using FIN packets.
19. Verify the firewall's ability to screen enumeration techniques using NULL packets.
20. Verify the firewall's ability to screen enumeration techniques measuring the packet window size (WIN).
21. Verify the firewall's ability to screen enumeration techniques using all flags set (XMAS).
22. Verify the firewall's ability to screen enumeration techniques using IPIDs.
23. Verify the firewall's ability to screen enumeration techniques using encapsulated protocols.
24. Measure the robustness of firewall and it's susceptibility to denial of service attacks with sustained TCP connections.
25. Measure the robustness of firewall and it's susceptibility to denial of service attacks with temporal TCP connections.
26. Measure the robustness of firewall and it's susceptibility to denial of service attacks with streaming UDP.
27. Measure the firewall's response to all types of ICMP packets.

#### Reviewing firewall logs

28. Test the firewall logging process.
29. Verify TCP and UDP scanning to server logs.
30. Verify automated vulnerability scans.
31. Verify services' logging deficiencies.

Copyright 2000-2003 Peter V. Herzog, ISECOM – The Institute for Security and Open Methodologies – [www.isecom.org](http://www.isecom.org) – [www.osstmm.org](http://www.osstmm.org)

## **Appendix F: Firewall Test Directions and Answer Sheet for Framework Experiment**

Directions: You are a security consultant for Llama corporation. You are given a list of firewall rules and a network diagram. Using only the information given, please list any weaknesses, misconfigurations and/or recommendations regarding the firewall. (You should not have to make too many assumptions, but feel free to write any.) Please also use the Open Source Security Testing Methodology Manual as a guide in finding these weaknesses and/or misconfigurations. You have one hour to complete this exercise.

*\*\*Please note that the methodology incorporates physical verification of certain attacks. Unfortunately you will not have network access to physically test this firewall so you will not be able to incorporate these into your findings. This is purposeful as to restrict the exercise in time duration.*

**Weaknesses and/or Misconfigurations:**  
*(Please list one at a time and number them)*

**Overall Recommendations:**  
*(Please list one at a time and number them)*

## **Appendix G: Firewall Test Directions and Answer Sheet for Non Framework Experiment**

Directions: You are a security consultant for Llama corporation. You are given a list of firewall rules and a network diagram. Unfortunately you will not have network access to physically test this firewall. Using only the information given, please list any weaknesses, misconfigurations or recommendations regarding the firewall rules. (You should not have to make many assumptions, but feel free to write any if necessary.) You have one hour to complete this exercise.

**Weaknesses and/or Misconfigurations :**  
*(Please list one at a time and number them)*

**Overall Recommendations:**  
*(Please list one at a time and number them)*

## Appendix H: Firewall Test Comprehensive Finding Set

### Weaknesses and/or Misconfigurations:

1. Internet traffic allowed to company internal network: Traffic is allowed from the Internet to 172.16.1.42 and 172.16.0.204 on the Internal Network by rules 3 and 11. This does not provide layered security. A breach of the servers security would expose the entire internal network to the attacker. All traffic sessions should stop in the DMZ.
2. Unnecessary Rule: Rules 6 and 7 have source and destination IP addresses that are on the same network. Because they are on the same network, traffic between those systems will not pass through the firewall. These two rules do not have any effect implying that the rule writer does not understand the concept and is relying on this rule to block traffic that will not be blocked.
3. DNS Any Any: Rule 5 allows DNS traffic through the firewall under all circumstances. This rule should be changed to only allow outbound DNS queries, and only allow inbound DNS to the DNS servers.
4. DMZ Any Any: Rule 8 allows 192.168.0.2 and 192.168.0.12 Exchange servers to communicate to any other server using any protocol. This rule should be restricted, allowing only the necessary protocols and restricted to the necessary destinations because if this box was compromised, it could cause a compromise of the entire network.
5. DMZ Replication: The DNS server should really be broken into two parts, one in DMZ and one in internal and the internal could replicate data from the DMZ.
6. Move Internal Web Server: Traffic is allowed into the internal network to a web server. This web server should be moved to the DMZ.
7. Rule 10 allows any source IP address to contact 192.168.0.2 and 192.168.0.12 Exchange Servers via SMTP, http, https, and FTP. Outlook Web Access should only need http and https. If SMTP and FTP are not needed, they should be disabled. If they are needed, the need should be documented. The other thing to note on this rule is that if there is an FTP and SMTP server needed, they should be broken out and hardened as separate servers to provide defense in depth.
8. Rule 4, traffic from 172.16.1.40 (labeled "Internal Web Server") to any other network any service. Why does the web server need to initiate web traffic (http and https)? This is not necessary in a stateful firewall and only affords a tunnel for hackers to pipe data out of the system. The requirement for this traffic should be verified and documented or the rule should be removed.
9. PCAnywhere All All: Rule 11 allows any source IP address, including Internet addresses, to contact 172.16.0.204 using tcp high port (ports above 1023) or PcAnywhere. There would not normally be any reason to allow the TCP high ports to be destinations for a connection because this is a stateful firewall. Also, PcAnywhere connections from the Internet to a server on the internal network, without some additional form of authentication, is not generally considered an acceptable risk. The requirement for this traffic should be verified and documented or the rule should be removed.
10. No Antispoof: There are no antispoof rules in place
11. No Deny All: There is no deny all at the end of the rules, Nokia's by default have this rule at the end but someone must have gone in and deleted it. This means all traffic is routed making all rules null.
12. Egress Filtering: For the most part, traffic is not limited outbound and it should be. Being a stateful firewall, almost everything should be blocked outbound.
13. Do not allow external network to talk to private internal network. Ever.
14. An additional firewall or filtering device should be added to filter traffic at the DMZ level and then use this Nokia to protect the internal network.

15. Rule of thumb for the whole environment, only allow services needed, block everything else.

Overall Recommendations: (This was not counted because was out of scope)

A security policy to be written and cover, in general terms, what kind of connections and traffic are allowed between the internal network, the DMZ and the Internet.

## Appendix I: Vulnerability Test Data: NessusScanner Output

### Nessus Scan Report

This report gives details on hosts that were tested and issues that were found. Please follow the recommended steps and procedures to eradicate these threats.

### Scan Details

Hosts which were alive and responding during test

1

Number of security holes found

9

Number of security warnings found

13

### Host List

Host(s)

Possible Issue

[192.168.0.100](#)

Security hole(s) found

[\[ return to top \]](#)

Analysis of Host

Address of Host

Port/Service

Issue regarding Port

192.168.0.100

[ssh \(22/tcp\)](#)

Security hole found

192.168.0.100

[ftp \(21/tcp\)](#)

Security hole found

192.168.0.100

[http \(80/tcp\)](#)

Security hole found

192.168.0.100

[sunrpc \(111/tcp\)](#)

Security notes found

192.168.0.100  
[https \(443/tcp\)](#)  
Security hole found

192.168.0.100  
[kdm \(1024/tcp\)](#)  
Security notes found

192.168.0.100  
[x11 \(6000/tcp\)](#)  
Security warning(s) found

192.168.0.100  
[sunrpc \(111/udp\)](#)  
Security notes found

192.168.0.100  
[unknown \(1024/udp\)](#)  
Security hole found

192.168.0.100  
[general/tcp](#)  
Security warning(s) found

192.168.0.100  
[general/udp](#)  
Security notes found

192.168.0.100  
[general/icmp](#)  
Security warning(s) found

#### Security Issues and Fixes: 192.168.0.100

Type  
Port  
Issue and Fix

**Vulnerability**  
ssh (22/tcp)

You are running a version of OpenSSH which is older than 3.4

There is a flaw in this version that can be exploited remotely to give an attacker a shell on this host.

Note that several distribution patched this hole without changing the version number of OpenSSH. Since Nessus solely relied on the banner of the remote SSH server to perform this check, this might be a false positive.

If you are running a RedHat host, make sure that the command : rpm -q openssh-server

Returns :  
openssh-server-3.1p1-6

Solution : Upgrade to OpenSSH 3.4 or contact your vendor for a patch

Risk factor : High

CVE : [CVE-2002-0639](#), [CVE-2002-0640](#)

BID : [5093](#)

Nessus ID : [11031](#)

#### Vulnerability

ssh (22/tcp)

You are running a version of OpenSSH older than OpenSSH 3.2.1

A buffer overflow exists in the daemon if AFS is enabled on your system, or if the options KerberosTgtPassing or AFSTokenPassing are enabled. Even in this scenario, the vulnerability may be avoided by enabling UsePrivilegeSeparation.

Versions prior to 2.9.9 are vulnerable to a remote root exploit. Versions prior to 3.2.1 are vulnerable to a local root exploit.

Solution :  
Upgrade to the latest version of OpenSSH

Risk factor : High

CVE : [CVE-2002-0575](#)

BID : [4560](#)

Nessus ID : [10954](#)

#### Warning

ssh (22/tcp)

You are running OpenSSH-portable 3.6.1 or older.

There is a flaw in this version which may allow an attacker to bypass the access controls set by the administrator of this server.

OpenSSH features a mechanism which can restrict the list of hosts a given user can log from by specifying a pattern in the user key file (ie: \*.mynetwork.com would let a user connect only from the local network).

However there is a flaw in the way OpenSSH does reverse DNS lookups.

If an attacker configures his DNS server to send a numeric IP address

when a reverse lookup is performed, he may be able to  
Circumvent this mechanism.

Solution : Upgrade to OpenSSH 3.6.2 when it comes out

Risk Factor : Low

CVE : [CAN-2003-0386](#)

BID : [7831](#)

Nessus ID : [11712](#)

Warning

ssh (22/tcp)

You are running OpenSSH-portable 3.6.1p1 or older.

If PAM support is enabled, an attacker may use a flaw in this  
version

to determine the existence of a given login name by comparing  
the times

the remote sshd daemon takes to refuse a bad password for a  
non-existent

login compared to the time it takes to refuse a bad password  
for an existing login.

An attacker may use this flaw to set up a brute force attack  
against the remote host.

\*\*\* Nessus did not check whether the remote SSH daemon  
is actually

\*\*\* using PAM or not, so this might be a false positive

Solution : Upgrade to OpenSSH-portable 3.6.1p2 or newer

Risk Factor : Low

CVE : [CAN-2003-0190](#)

BID : [7482](#)

Nessus ID : [11574](#)

Warning

ssh (22/tcp)

The remote SSH daemon supports connections made  
using the version 1.33 and/or 1.5 of the SSH protocol.

These protocols are not completely cryptographically  
safe so they should not be used.

Solution :

If you use OpenSSH, set the option 'Protocol' to '2'

If you use SSH.com's set the option 'Ssh1Compatibility' to 'no'

Risk factor : Low

Nessus ID : [10882](#)

Informational

ssh (22/tcp)

An ssh server is running on this port

Nessus ID : [10330](#)

Informational  
ssh (22/tcp)  
Remote SSH version : SSH-1.99-OpenSSH\_3.1p1  
Nessus ID : [10267](#)

Informational  
ssh (22/tcp)  
The remote SSH daemon supports the following versions of the SSH protocol :

- . 1.33
- . 1.5
- . 1.99
- . 2.0

Nessus ID : [10881](#)

### Vulnerability

ftp (21/tcp)

The remote FTP server allows any anonymous user to browse the entire remote disk by issuing commands like :

```
LIST ../../../../  
LIST ..\..\..\.
```

Solution : Contact your vendor for a patch  
Risk factor : High

The command we found to escape the chrooted environment is : .../  
The root dir of the remote server contains :  
total 0

CVE : [CVE-2001-0680](#), [CAN-2001-1335](#), [CAN-2001-0582](#)  
BID : [2618](#), [2786](#)  
Nessus ID : [11112](#)

### Vulnerability

ftp (21/tcp)

The remote FTP server seems to be vulnerable to an integer conversion bug when it receives a malformed argument to the 'REST' command.

An attacker may exploit this flaw to crash the remote FTP daemon and possibly execute arbitrary code on this host.

Solution : if the remote FTP server is HP/UX ftpd, then apply patch PHNE\_21936.

Risk Factor : High  
Nessus ID : [11701](#)

### Vulnerability

ftp (21/tcp)

You seem to be running an FTP server which is vulnerable to  
The 'glob heap corruption' flaw.  
An attacker may use this problem to execute arbitrary  
commands on this host.

\*\*\* Nessus relied solely on the banner of the server to issue this  
warning,  
\*\*\* so this alert might be a false positive  
\*\*\* NOTE: must have a valid username/password to fully check  
this vulnerability

Solution : Upgrade your ftp server software to the latest version.  
Risk factor : High

CVE : [CAN-2001-0249](#), [CVE-2001-0550](#)  
BID : [2550](#), [3581](#)  
Nessus ID : [10821](#)

Warning  
ftp (21/tcp)

This FTP service allows anonymous logins. If you do not  
want to share data with anyone you do not know, then you  
should deactivate  
the anonymous account, since it can only cause troubles.  
Under most Unix system, doing :  
echo ftp >> /etc/ftpusers  
will correct this.

The content of the remote FTP root is :

```
total 32
d--x--x--x 2 root root 4096 Oct 21 12:55 bin
d--x--x--x 2 root root 4096 Oct 21 19:59 etc
drwxr-xr-x 2 root root 4096 Oct 21 19:59 lib
drwxr-xr-x 2 root 50 4096 Mar 22 2001 pub
```

Risk factor : Low  
CVE : [CAN-1999-0497](#)  
Nessus ID : [10079](#)

Informational  
ftp (21/tcp)  
An FTP server is running on this port.  
Here is its banner :  
220 localhost.localdomain FTP server (Version wu-2.6.1-16)  
ready.  
Nessus ID : [10330](#)

Informational  
ftp (21/tcp)  
Remote FTP server banner :  
220 localhost.localdomain FTP server (Version wu-2.6.1-16) r  
eady.  
Nessus ID : [10092](#)

## Vulnerability

http (80/tcp)

The remote host appears to be vulnerable to the Apache Web Server Chunk Handling Vulnerability.

If Safe Checks are enabled, this may be a false positive since it is based on the version of Apache. Although unpatched Apache versions 1.2.2 and above, 1.3 through 1.3.24 and 2.0 through 2.0.36, the remote server may be running a patched version of Apache

\*\*\* Note : as safe checks are enabled, Nessus solely relied on the banner to issue this alert

Solution : Upgrade to version 1.3.26 or 2.0.39 or newer

See also :

[http://httpd.apache.org/info/security\\_bulletin\\_20020617.txt](http://httpd.apache.org/info/security_bulletin_20020617.txt)

[http://httpd.apache.org/info/security\\_bulletin\\_20020620.txt](http://httpd.apache.org/info/security_bulletin_20020620.txt)

Risk factor : High

CVE : [CVE-2002-0392](#)

BID : [5033](#)

Nessus ID : [11030](#)

Warning

http (80/tcp)

Your webserver supports the TRACE and/or TRACK methods. It has been shown that servers supporting this method are subject to cross-site-scripting attacks, dubbed XST for 'Cross-Site-Tracing', when used in conjunction with various weaknesses in browsers.

An attacker may use this flaw to trick your legitimate web users to give him their credentials.

Solution: Disable these methods.

If you are using Apache, add the following lines for each virtual host in your configuration file :

```
RewriteEngine on
RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK)
RewriteRule .* - [F]
```

If you are using Microsoft IIS, use the URLScan tool to deny HTTP TRACE requests or to permit only the methods needed to meet site Requirements and policy.

See

[http://www.whitehatsec.com/press\\_releases/WH-PR-20030120.pdf](http://www.whitehatsec.com/press_releases/WH-PR-20030120.pdf)  
<http://archives.neohapsis.com/archives/vulnwatch/2003-q1/0035.html>

Risk factor : Medium  
Nessus ID : [11213](#)

Warning  
http (80/tcp)

The remote host appears to be running a version of Apache which is older than 1.3.27

There are several flaws in this version, you should upgrade to 1.3.27 or newer.

\*\*\* Note that Nessus solely relied on the version number  
\*\*\* of the remote server to issue this warning. This might  
\*\*\* be a false positive

Solution : Upgrade to version 1.3.27  
See also : <http://www.apache.org/dist/httpd/Announcement.html>  
Risk factor : Medium  
CVE : [CAN-2002-0839](#), [CAN-2002-0840](#), [CAN-2002-0843](#)  
BID : [5847](#), [5884](#), [5995](#), [5996](#)  
Nessus ID : [11137](#)

Informational  
http (80/tcp)  
A web server is running on this port  
Nessus ID : [10330](#)

Informational  
http (80/tcp)  
The following directories were discovered:  
/cgi-bin, /icons, /manual  
Nessus ID : [11032](#)

Informational  
http (80/tcp)  
The remote web server type is :

Apache/1.3.19

The 'ServerTokens' directive is set to ProductOnly however we could determine that the version of the remote server by requesting a non-existent page.

Nessus ID : [10107](#)

Informational  
http (80/tcp)  
An information leak occurs on Apache based web servers whenever the UserDir module is enabled. The vulnerability allows an external

attacker to enumerate existing accounts by requesting access to their home directory and monitoring the response.

Solution:

1) Disable this feature by changing 'UserDir public\_html' (or whatever) to 'UserDir disabled'.

Or

2) Use a RedirectMatch rewrite rule under Apache -- this works even if there

is no such entry in the password file, e.g.:

RedirectMatch ^/~(.\*)\$

[http://my-target-webserver.somewhere.org/\\$1](http://my-target-webserver.somewhere.org/$1)

Or

3) Add into httpd.conf:

ErrorDocument 404 <http://localhost/sample.html>

ErrorDocument 403 <http://localhost/sample.html>

(NOTE: You need to use a FQDN inside the URL for it to work properly).

Additional Information:

<http://www.securiteam.com/unixfocus/5WP0C1F5FI.html>

Risk factor : Low

CVE : [CAN-2001-1013](#)

BID : [3335](#)

Nessus ID : [10766](#)

Informational

sunrpc (111/tcp)

The RPC portmapper is running on this port.

An attacker may use it to enumerate your list of RPC services. We recommend you filter traffic going to this port.

Risk factor : Low

CVE : [CAN-1999-0632](#), [CVE-1999-0189](#)

BID : [205](#)

Nessus ID : [10223](#)

Informational

sunrpc (111/tcp)

RPC program #100000 version 2 'portmapper' (portmap sunrpc rpcbind) is running on this port

Nessus ID : [11111](#)

Vulnerability

https (443/tcp)

The remote host appears to be vulnerable to the Apache Web Server Chunk Handling Vulnerability.

If Safe Checks are enabled, this may be a false positive since it is based on the version of Apache. Although unpatched Apache versions 1.2.2 and above, 1.3 through 1.3.24 and 2.0 through 2.0.36, the remote server may be running a patched version of Apache

\*\*\* Note : as safe checks are enabled, Nessus solely relied on the banner to issue this alert

Solution : Upgrade to version 1.3.26 or 2.0.39 or newer

See also :

[http://httpd.apache.org/info/security\\_bulletin\\_20020617.txt](http://httpd.apache.org/info/security_bulletin_20020617.txt)

[http://httpd.apache.org/info/security\\_bulletin\\_20020620.txt](http://httpd.apache.org/info/security_bulletin_20020620.txt)

Risk factor : High

CVE : [CVE-2002-0392](#)

BID : [5033](#)

Nessus ID : [11030](#)

#### Vulnerability

https (443/tcp)

The remote host seems to be using a version of OpenSSL which is older than 0.9.6e or 0.9.7-beta3

This version is vulnerable to a buffer overflow which, may allow an attacker to obtain a shell on this host.

\*\*\* Note that since safe checks are enabled, this check  
\*\*\* might be fooled by non-openssl implementations and  
\*\*\* produce a false positive.  
\*\*\* In doubt, re-execute the scan without the safe checks

Solution : Upgrade to version 0.9.6e (0.9.7beta3) or newer

Risk factor : High

CVE : [CAN-2002-0656](#), [CAN-2002-0655](#), [CAN-2002-0657](#),  
[CAN-2002-0659](#), [CVE-2001-1141](#)

BID : [5363](#)

Nessus ID : [11060](#)

#### Warning

https (443/tcp)

Your webserver supports the TRACE and/or TRACK methods.

It has been

shown that servers supporting this method are subject to cross-site-scripting attacks, dubbed XST for 'Cross-Site-Tracing', when used in conjunction with

various weaknesses in browsers.

An attacker may use this flaw to trick your legitimate web users to give him their credentials.

Solution: Disable these methods.

If you are using Apache, add the following lines for each virtual host in your configuration file :

```
RewriteEngine on
RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK)
RewriteRule .* - [F]
```

If you are using Microsoft IIS, use the URLScan tool to deny HTTP TRACE requests or to permit only the methods needed to meet site requirements and policy.

See

[http://www.whitehatsec.com/press\\_releases/WH-PR-20030120.pdf](http://www.whitehatsec.com/press_releases/WH-PR-20030120.pdf)  
<http://archives.neohapsis.com/archives/vulnwatch/2003-q1/0035.html>

Risk factor : Medium

Nessus ID : [11213](#)

Warning

https (443/tcp)

The SSLv2 server offers 5 strong ciphers, but also 0 medium strength and 2 weak "export class" ciphers. The weak/medium ciphers may be chosen by an export-grade or badly configured client software. They only offer a limited protection against a brute force attack

Solution: disable those ciphers and upgrade your client software if necessary

Nessus ID : [10863](#)

Warning

https (443/tcp)

The remote host appears to be running a version of Apache which is older than 1.3.27

There are several flaws in this version, you should upgrade to 1.3.27 or newer.

\*\*\* Note that Nessus solely relied on the version number  
\*\*\* of the remote server to issue this warning. This might  
\*\*\* be a false positive

Solution : Upgrade to version 1.3.27

See also : <http://www.apache.org/dist/httpd/Announcement.html>

Risk factor : Medium

CVE : [CAN-2002-0839](#), [CAN-2002-0840](#), [CAN-2002-0843](#)

BID : [5847](#), [5884](#), [5995](#), [5996](#)

Nessus ID : [11137](#)

Informational

https (443/tcp)

A TLSv1 server answered on this port

Nessus ID : [10330](#)

Informational

https (443/tcp)

A web server is running on this port through SSL

Nessus ID : [10330](#)

Informational

https (443/tcp)

The following directories were discovered:

/cgi-bin, /icons, /manual

Nessus ID : [11032](#)

Informational

https (443/tcp)

The remote web server type is :

Apache/1.3.19

The 'ServerTokens' directive is set to ProductOnly however we could determine that the version of the remote server by requesting a non-existent page.

Nessus ID : [10107](#)

Informational

https (443/tcp)

An information leak occurs on Apache based web servers whenever the UserDir module is enabled. The vulnerability allows an external attacker to enumerate existing accounts by requesting access to their home directory and monitoring the response.

Solution:

1) Disable this feature by changing 'UserDir public\_html' (or whatever) to 'UserDir disabled'.

Or

2) Use a RedirectMatch rewrite rule under Apache -- this works even if there is no such entry in the password file, e.g.:

RedirectMatch ^~(.\*\$

[http://my-target-webserver.somewhere.org/\\$1](http://my-target-webserver.somewhere.org/$1)

Or

3) Add into httpd.conf:

ErrorDocument 404 <http://localhost/sample.html>

ErrorDocument 403 <http://localhost/sample.html>

(NOTE: You need to use a FQDN inside the URL for it to work properly).

Additional Information:

<http://www.securiteam.com/unixfocus/5WP0C1F5FI.html>

Risk factor : Low

CVE : [CAN-2001-1013](#)

BID : [3335](#)

Nessus ID : [10766](#)

Informational

https (443/tcp)

Here is the SSLv2 server certificate:

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 0 (0x0)

Signature Algorithm: md5WithRSAEncryption

Issuer: C=--, ST=SomeState, L=SomeCity, O=SomeOrganization, OU=SomeOrganizationalUnit, CN=localhost.localdomain/emailAddress=root@localhost.localdomain

Validity

Not Before: Oct 21 12:59:17 2003 GMT

Not After : Oct 20 12:59:17 2004 GMT

Subject: C=--, ST=SomeState, L=SomeCity, O=SomeOrganization, OU=SomeOrganizationalUnit, CN=localhost.localdomain/emailAddress=root@localhost.localdomain

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (1024 bit)

Modulus (1024 bit):

00:cb:65:67:50:32:03:7e:35:63:f7:16:56:3b:d0:

da:6b:40:7d:55:3a:ad:3b:63:21:73:d5:70:ef:16:

be:01:68:f3:a0:e3:48:fc:7c:c7:41:86:c4:9b:45:

fd:e1:28:9a:4c:24:3b:67:58:d4:4e:39:08:69:dc:

c6:ad:ac:3d:b5:aa:a1:8f:c8:cc:dc:ae:f8:71:fe:

02:c6:ae:a9:23:05:86:25:12:5d:e2:1b:eb:f2:ab:

ad:0c:80:1c:8b:a2:a2:d3:7a:39:ae:c2:18:1d:53:

4d:05:d0:41:56:aa:11:3e:92:52:6b:3f:c5:e2:9a:

e7:72:80:b7:c1:80:2e:39:6d

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Subject Key Identifier:

7A:5A:94:73:92:28:8F:6C:17:49:89:A7:19:CC:F6:0D:42:5C:75:

2E

X509v3 Authority Key Identifier:

keyid:7A:5A:94:73:92:28:8F:6C:17:49:89:A7:19:CC:F6:0D:42:  
5C:75:2E  
DirName:/C=--/ST=SomeState/L=SomeCity/O=Some  
Organization/OU=  
SomeOrganizationalUnit/CN=localhost.localdomain/  
emailAddress=root@localhost.localdomain  
serial:00

X509v3 Basic Constraints:

CA:TRUE

Signature Algorithm: md5WithRSAEncryption

40:fe:f2:32:36:d2:4a:48:5f:4e:2b:06:d6:19:3a:f7:21:b6:  
ab:42:04:7c:54:f5:9a:86:63:1f:80:4b:41:5b:8e:9b:7e:42:  
9b:a1:f9:12:82:10:c5:fb:84:75:fc:90:e5:ba:3a:cd:2a:67:  
ee:5e:95:d2:9e:ad:a8:d3:fb:71:86:ba:3d:5a:16:00:71:80:  
63:a2:cc:c6:28:2e:ab:8a:7f:89:28:42:8b:84:34:b7:2a:62:  
01:15:6d:ed:57:47:54:43:ea:90:1c:aa:a5:b5:91:fc:60:1d:  
0c:a8:cd:f9:3d:e6:67:1b:b8:b0:8f:a8:bf:18:3e:c6:9b:06:  
cf:17

Nessus ID : [10863](#)

Informational

https (443/tcp)

Here is the list of available SSLv2 ciphers:

RC4-MD5

EXP-RC4-MD5

RC2-CBC-MD5

EXP-RC2-CBC-MD5

DES-CBC-MD5

DES-CBC3-MD5

RC4-64-MD5

Nessus ID : [10863](#)

Informational

https (443/tcp)

This TLSv1 server also accepts SSLv2 connections.

This TLSv1 server also accepts SSLv3 connections.

Nessus ID : [10863](#)

Informational

kdm (1024/tcp)

RPC program #100024 version 1 'status' is running on this port

Nessus ID : [11111](#)

Warning

x11 (6000/tcp)

This X server does \*not\* allow any client to connect to it  
however it is recommended that you filter incoming connections  
to this port as attacker may send garbage data and slow down  
your X session or even kill the server.

Here is the server version : 11.0

Here is the message we received : Client is not authorized

Solution : filter incoming connections to ports 6000-6009  
Risk factor : Low  
CVE : [CVE-1999-0526](#)  
Nessus ID : [10407](#)

Informational  
sunrpc (111/udp)  
RPC program #100000 version 2 'portmapper' (portmap  
sunrpc rpcbind) is running on this port

Nessus ID : [11111](#)

**Vulnerability**  
unknown (1024/udp)

The remote statd service may be vulnerable to a format string attack.

This means that an attacker may execute arbitrary code thanks to a bug in this daemon.

\*\*\* Nessus reports this vulnerability using only information that was gathered. Use caution when testing without safe checks enabled.

Solution : upgrade to the latest version of rpc.statd  
Risk factor : High  
CVE : [CVE-2000-0666](#)  
BID : [1480](#)  
Nessus ID : [10544](#)

Warning  
unknown (1024/udp)

The statd RPC service is running. This service has a long history of security holes, so you should really know what you are doing if you decide to let it run.

\*\*\* No security hole regarding this program have been tested, so this might be a false positive.

Solution : We suggest that you disable this service.  
Risk factor : High  
CVE : [CVE-1999-0018](#), [CVE-1999-0019](#), [CVE-1999-0493](#)  
BID : [127](#), [450](#)  
Nessus ID : [10235](#)

Informational  
unknown (1024/udp)  
RPC program #100024 version 1 'status' is running on this port

Nessus ID : [11111](#)

Warning  
general/tcp

The remote host does not discard TCP SYN packets which have the FIN flag set.

Depending on the kind of firewall you are using, an attacker may use this flaw to bypass its rules.

See also :

<http://archives.neohapsis.com/archives/bugtraq/2002-10/0266.html>

<http://www.kb.cert.org/vuls/id/464113>

Solution : Contact your vendor for a patch

Risk factor : Medium

BID : [7487](#)

Nessus ID : [11618](#)

Informational

general/tcp

Remote OS guess : Linux Kernel 2.4.0 - 2.5.20

CVE : [CAN-1999-0454](#)

Nessus ID : [11268](#)

Informational

general/udp

For your information, here is the traceroute to 192.168.0.100 :  
192.168.0.100

Nessus ID : [10287](#)

Warning

general/icmp

The remote host answers to an ICMP timestamp request.  
This allows an attacker to know the date which is set on your machine.

This may help him to defeat all your time based authentication protocols.

Solution : filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14).

Risk factor : Low

CVE : [CAN-1999-0524](#)

Nessus ID : [10114](#)

---

*This file was generated by [Nessus](#), the open-sourced security scanner.*

## Appendix J: Vulnerability Test Data: Other Scanner Output

Report Date: Wednesday, October 22, 2003 10:03  
Scanner: 1.3.1; Oct 7 2003 15:26:27 [RedHat Linux 6.2 x86]  
Scanner Config: 2003/10/09

- Name: test.nameobfuscated.com (192.168.0.100)  
Operating System: UNIX  
Audit Date: *Wednesday, October 22, 2003 09:44*  
Auditor: *root@localhost.localdomain*

### Security Audit Summary

- [Configuration Settings](#) - 6
- [Potential Vulnerabilities](#) - 2
- [Information Leaks](#) - 2

### Security Audit Breakdown

#### Configuration Settings - 6

#### Potential Vulnerabilities - 2

- [Security Obsolesced Software Versions](#) - 2

#### Information Leaks - 2

- [Information About System Resources](#) - 2

### Security Audit Findings

- Configuration Settings
  - [\[HTTP/80/TCP\]](#) Method `MKCOL' is enabled.
  - [\[HTTP/80/TCP\]](#) Method `DELETE' is enabled.
  - [\[HTTP/80/TCP\]](#) Method `LOCK' is enabled.
  - [\[HTTP/80/TCP\]](#) Method `UNLOCK' is enabled.
  - [\[HTTP/80/TCP\]](#) Method `COPY' is enabled.
  - [\[HTTP/80/TCP\]](#) Method `MOVE' is enabled.
- Potential Vulnerabilities
  - Security Obsolesced Software Versions
    - [\[SSH/22/TCP\]](#) Server version `Protocol 1.99; Server OpenSSH\_3.1p1' is known to contain vulnerabilities.

- [\[FTP/21/TCP\]](#) Server version `wu-2.6.1-16' is known to contain vulnerabilities.
  - Information Leaks
    - Information About System Resources
      - [\[SunRPC/111/TCP\]](#) SunRPC program 100000 ([portmapper](#)) is active. [More...](#)
      - [\[SunRPC/111/UDP\]](#) SunRPC program 100000 ([portmapper](#)) is active. [More...](#)
  - Active Network Servers
    - [\[22/TCP\]](#) [SSH](#) is active. [More...](#)
    - [\[21/TCP\]](#) [FTP](#) is active. [More...](#)
    - [\[443/TCP/TLSv1\]](#) Unidentified-server is active.
    - [\[80/TCP\]](#) [HTTP](#) is active. [More...](#)
    - [\[6000/TCP\]](#) [X](#) is active. [More...](#)
    - [\[SunRPC/111/TCP\]](#) SunRPC program 100000 ([portmapper](#)) is active. [More...](#)
    - [\[SunRPC/111/UDP\]](#) SunRPC program 100000 ([portmapper](#)) is active. [More...](#)
  - Available Network Services
    - Unauthenticated File Service
      - [\[FTP/21/TCP\]](#) Anonymous FTP service is active.
    - User Login Services
      - [\[22/TCP\]](#) [SSH](#) is active. [More...](#)
      - [\[21/TCP\]](#) [FTP](#) is active. [More...](#)
  - Operating system is `UNIX'.
  - Server Version Strings
    - [\[SSH/22/TCP\]](#) Server version is `Protocol 1.99; Server OpenSSH\_3.1p1'.
    - [\[HTTP/80/TCP\]](#) Server version is `Apache'.
    - [\[FTP/21/TCP\]](#) Server version is `wu-2.6.1-16'.
  - Server Protocol Versions
    - [\[SSH/22/TCP\]](#) Protocol version is `1.99'.
  - Network Transport Information

- IP Transport Information
      - Host responded to ICMP Echo Request.
      - Host responded to ICMP Time Stamp Request.
  - Port Scan Information
    - TCP Port Scan Data
      - The following TCP ports were scanned: [More...](#)  
1-134, 136-65535
      - The following TCP ports were visible: [More...](#)  
1-134, 136-65535
      - The following TCP ports were active: [More...](#)  
21-22, 80, 111, 443, 1024, 6000
      - The servers on these TCP ports could not be identified:  
[More...](#)  
443, 1024
      - The following inactive TCP ports were visible: [More...](#)  
1-20, 23-79, 81-110, 112-134, 136-442, 444-1023, 1025-5999, 6001-65535
    - UDP Port Scan Data
      - The following UDP ports were scanned: [More...](#)  
1-1023, 1025, 1080, 1352, 1433-1434, 1494, 1512, 1524, 1527, 1645-1646, 1649, 1661-1672, 1718-1720, 1758, 1760, 1789, 1812-1813, 1895-1896, 1911, 1997, 2049, 2102-2104, 2140, 2150, 2401, 2430-2433, 2627, 2766, 2784, 2988, 3130, 3264, 3306, 3346, 3455, 4321, 4444, 5002, 5060, 5232, 5308, 5354-5355, 5432, 5999-6063, 6660-6669, 7000-7009, 8008, 8080-8081, 9359, 9876, 10080-10081, 10666-10667, 11371, 11720, 12345, 20011-20012, 24554, 26000, 26208, 26274, 26740, 27374, 27440, 29891, 31337, 32700-32900, 33434, 34555, 47262, 60177, 60179, 65534
      - The following UDP ports were active: [More...](#)  
111
  - Server Banners
    - [\[FTP/21/TCP\]](#) Server banner -  
220 localhost.localdomain FTP server (Version wu-2.6.1-16) ready.

---

## References

### HTTP

The **HTTP** (HyperText Transfer Protocol) is the protocol used by the World Wide Web. It is defined in [RFC 2068](#) and [RFC 1945](#). The standard port for HTTP is TCP port 80, but servers

can often be found on other TCP ports. The following advisories have been issued related to HTTP servers:

- [CERT Advisory CA-97.25](#) CGI meta characters
- [CERT Advisory CA-97.24](#) count.cgi
- S.N.I. 01-13-97 Apache HTTPD cookies
- L0pht Advisory Lotus Domino Server
- L0pht Advisory `test-cgi' script
- [CERT Advisory CA-96.06](#) CGI Example code
- [AUSCERT AA-96.01](#) CGI Example code
- [CERT Advisory CA-95.04](#) NCSA HTTPD

### **SSH**

The **SSH** (Secure Shell) service is a remote login and command execution service. Its use of encryption provides eavesdrop protection. It does offer features that may not be desirable in all environments, such as the ability to forward TCP ports across the encrypted channel, bypassing any filtering that may be done by a site. SSH is normally found on TCP port 22.

### **FTP**

The **FTP** (File Transfer Protocol) service is the standard service for transferring files between hosts. It is documented in [RFC 959](#). The standard port for this service is TCP port 21. The following advisories have been issued for servers implementing the FTP service:

- [CERT Advisory CA-97.27](#) FTP PORT command misuse
- [CERT Advisory CA-97.16](#) FTPD signal handling
- [AUSCERT Advisory AA-9703](#) FTPD signal handling
- [CIAC Advisory H-63](#) FTPD signal handling
- [CERT Advisory CA-95.16](#) wuarchive FTPD SITE EXEC
- [CERT Advisory CA-94.08](#) FTPD SITE EXEC
- [CERT Advisory CA-94.07](#) wuarchive FTPD Trojan
- [CERT Advisory CA-93.06](#) wuarchive Guest flag
- [CERT Advisory CA-92.09](#) AIX Anonymous FTP

In addition, CERT provides documents on configuring for anonymous FTP and how the service can be abused.

### **portmapper**

The **SunRPC portmapper** is used by all SunRPC programs for registration services. It is also known by the newer name **rpcbind**. This server should not be accessible from outside the local network. The following advisories have been issued about the portmapper:

- [CIAC Advisory H-70](#) Solaris rpcbind

- S.N.I. Advisory SNI-14 Solaris rpcbind

## **X**

The **X** protocol is used by the X Windowing system and is used to access a graphical display. The standard ports for X servers are TCP ports 6000-6063 (6000 + the display number).

### **.TCPPortsScanned**

The list of TCP ports that scanned. This can contain ports other than those provided to if it collects port registrations from servers such as the Sun RPC portmapper.

### **.TCPPortsVisible**

defines a "visible" TCP port as any port which it appears to be able to send packets to. It is intended for checking whether packet filtering rules are set up as expected. If is able to complete the TCP 3-way handshake, or if it receives a TCP RST when attempting to connect to the port, it will be marked as visible.

Note that certain packet filter configurations may forge TCP RST packets, or may intercept and complete the TCP 3-way handshake, causing to mis-report the status of ports.

### **.TCPPortsActive**

defines a "active" TCP port as any port which it appears to be able to connect to. If is able to complete the TCP 3-way handshake it will be marked as active. Any TCP port that is "active" is also considered "visible".

Note that certain packet filter configurations may intercept and complete the TCP 3-way handshake, causing to mis-report the status of ports.

### **.TCPPortsUnknown**

records any TCP port as "unknown" if it is unable to identify the server that is listening on that port. This happens if is unable to elicit any output from the server, or it is unable to recognize the output if any is available.

### **.TCPPortsNoServer**

These TCP ports were [visible](#) but did not have a server [active](#) on them. Of interest to anyone wanting to tighten firewalling packet filter rules down to only those ports that have active servers on them.

### **.UDPPortsScanned**

The list of UDP ports that scanned. This can contain ports other than those provided to if it collects port registrations from servers such as the Sun RPC portmapper.

### **.UDPPortsVisible**

defines a "visible" UDP port as any port which it appears to be able to send packets to. It is intended for checking whether packet filtering rules are set up as expected. If receives a UDP datagram from the port, or if it receives an ICMP Port Unreachable message for the port, it will be marked as visible.

Note that certain packet filter configurations may forge ICMP Port Unreachable messages, causing to mis-report the status of ports.

### **.UDPPortsActive**

defines a "active" UDP port as any port which receives a UDP datagram from in response to a datagram to that port. Any UDP port considered "active" is also considered "visible".

Note that most UDP based servers will not answer unless the received message is properly formatted. Thus, there may be additional UDP ports that have servers listening on them that are not listed. Also, some UDP based services, such as "syslogd" or "discard" never send out messages. The UDP ports such services are listening on should never show up as active.

***.UDPPortsQuiet***

will record any UDP port as "quiet" if it is unable to get any type of response from the port. This means that either the UDP port is blocked via a packet filter, or that the service on that port will not respond to the messages that is sending. There is no way to distinguish the reason.

***.UDPPortsNoServer***

These UDP ports were [visible](#) but did not have a server [active](#) on them. Of interest to anyone wanting to tighten firewalling packet filter rules down to only those ports that have active servers on them.

## Appendix K – Vulnerability Test Framework

OSSTMM 2.1 - The Open Source Security Testing Methodology Manual  
23 August 2003

### 4. Vulnerability Research and Verification

The focus of this module is in the identification, understanding, and verification of weaknesses, misconfigurations and vulnerabilities within a host or network.

Research involved in finding vulnerabilities is necessary up until the delivery of the report. This involves searching online databases and mailing lists specific to the systems and network being tested. Do not confine yourself to the web, consider using IRC, Newsgroups, and underground FTP sites.

Testing for vulnerabilities using automated tools is an efficient way to determine existing holes and system patch level. Although many automated scanners are currently on the market and in the underground, it is important for the tester to identify and incorporate the current underground scripts/exploits into this testing. However, manual verification is necessary for eliminating false positives, expanding the hacking scope, and discovering the data flow in and out of the network. Manual testing refers to a person or persons at the computer using creativity, experience, and ingenuity to test the target network.

<b>Expected Results:</b>	Type of application or service by vulnerability Patch levels of systems and applications
	List of possible denial of service vulnerabilities List of areas secured by obscurity or visible access List of actual vulnerabilities minus false positives
	List of internal or DMZ systems List of mail, server, and other naming conventions Network map

1. Integrate the currently popular scanners, hacking tools, and exploits into the tests.
2. Measure the target organization against the currently popular scanning tools.
3. Attempt to determine vulnerability by system and application type.
4. Attempt to match vulnerabilities to services.
5. Attempt to determine application type and service by vulnerability.
6. Perform redundant testing with at least 2 automated vulnerability scanners.
7. Identify all vulnerabilities according to applications.
8. Identify all vulnerabilities according to operating systems.
9. Identify all vulnerabilities from similar or like systems that may also affect the target systems.
10. Verify all vulnerabilities found during the exploit research phase for false positives and false negatives.
11. Verify all positives (be aware of your contract if you are attempting to intrude or might cause a denial of service).

Copyright 2000-2003 Peter V. Herzog, ISECOM – The Institute for Security and Open Methodologies – [www.isecom.org](http://www.isecom.org) - [www.osstmm.org](http://www.osstmm.org)  
ISECOM is the OSSTMM Professional Security Tester (OPST) and OSSTMM Professional Security Analyst (OPSA) certification authority.

## **Appendix L – Vulnerability Test Directions – Framework Used**

Directions: You are a security consultant for Llama corporation. You are asked to perform a vulnerability assessment of a business critical server hosted in an Internet accessible. As this machine is business critical, you are requested not to physically harm the box in any way. You are given two different vulnerability scanner outputs to work with. Please list any vulnerabilities, weaknesses or misconfigurations on the system. Please also use the Open Source Security Testing Methodology Manual as your guide in finding these vulnerabilities, weaknesses or misconfigurations. You will have one hour to complete the exercise. If you want to do any hands on testing, you can reach the server on 9.99.24.250.

### **Vulnerabilities, Weaknesses or Misconfigurations:**

*(Please list one at a time and number them)*

### **Overall Recommendations:**

*(Please list one at a time and number them)*

## **Appendix M – Vulnerability Test Directions – No Framework Used**

Directions: You are a security consultant for Llama corporation. You are asked to perform a vulnerability assessment of a business critical server hosted in an Internet accessible. As this machine is business critical, you are requested not to physically harm the box in any way. You are given two different vulnerability scanner outputs to work with. Please list any vulnerabilities, weaknesses or misconfigurations on the system. You will have one hour to complete the exercise. If you want to do any hands on testing, you can reach the server on 9.99.24.250.

**Vulnerabilities, Weaknesses or Misconfigurations:**  
*(Please list one at a time and number them)*

**Overall Recommendations:**  
*(Please list one at a time and number them)*

## Appendix N: Vulnerability Test Comprehensive Finding Set

### Vulnerabilities, Weaknesses or Misconfigurations:

- 1) SSH daemon accepts protocol version 1 connections which is thought by some to have cryptographic weaknesses. Recommend disable version 1 functionality.
- 2) FTP server accepts anonymous connections. This allows anyone to connect and transfer files without authentication.
- 3) FTP server is downlevel and vulnerable and flawed. Upgrade to a newer version.
- 4) Apache Downlevel (80, 443) Based on the version # in the identification banner, the Apache web servers running on ports 80 and 443 appear to be vulnerable. Recommend upgrade to a newer version.
- 5) The TRACE and/or TRACK HTTP methods are supported by the web server. These can enable cross-site scripting attacks. Disable these methods.
- 6) Directory traversal on the webserver is possible. The following directories were publicly visible: /cgi-bin, /icons, /manual. This is an information leak.
- 7) The web server allows user enumeration.
- 8) Filter traffic to RPC port 111 so that RPC portmapper will not leak information that may aid an attacker.
- 9) OpenSSL is outdated and vulnerable to a buffer overflow. Upgrade to a newer version
- 10) Configure web server on 443 to only use strong ciphers.
- 11) X windows should not be running on a DMZ box. Attackers can send garbage to slow down the X session or kill the server. Additionally, an attacker can attach to the X server and potentially compromise the X sessions of others.
- 12) The version of statd that is running is obsolete and is usually unnecessary. Turn off if unnecessary or upgrade to a newer version.
- 13) Drop ICMP traffic to/from the outside to avoid leaking the system date.
- 14) The remote host does not discard TCP SYN packets which have the FIN flag set. Make sure that firewall cannot be bypassed by taking advantage of this fact.
- 15) No need to have ftp if ssh is running, recommend disable FTP.
- 16) Portmapper should be disabled.
- 17) Rstatd should be disabled.
- 18) Apache is using a self signed certificate, recommend getting certified CA certificate
- 19) Many services are running on one host, recommend breaking the box out into ftp server, ssh server, web server as totally and wholly separate boxes so they can be hardened appropriately and provide for defense in depth.
- 20) SSH is not downlevel, that was a false positive but there was no way for a remote analyst to know so it was appropriate for them to recommend it be checked BUT this was not actually a finding so it was not counted.
- 21) Apache is running as root, recommend run as a user with lesser privileges.

### Overall Recommendations: (out of scope so not counted)

- 1) Run regular vulnerability scans on this box to ensure it stays patched.
- 2) Patching policy should be implemented.

## Appendix O: IDS Signature Cross Reference Comments

*This gives comments on each IDS signature – why each signature was or was not considered covered by the framework.*

SIGNATURE		Which Section	Comments
Nachi_Worm_ICMP_Echo_Request	yes	Vulnerability Research and Identification/Access Control Testing	Patching the windows systems would allow the environment not to be compromised by this worm. The problem with a lot of these worms is the systems are not patched fast enough.
IP_Localhost_Source_Spoof	yes	Access Control Testing	Anti-spoof rules would prevent this attack.
ICMP_Sweep_Echo	maybe	Access Control Testing/ Intrusion Detection System Testing	The firewall acl testing rule number one is to test the ACL against a "Deny all" rule, which means deny any that you do not specifically need. While there is no reason for ICMP from the Internet, the framework does not specifically mention that it is a bad idea. Also if the IDS was in place per "Intrusion Detection Systems Testing" this would be caught and reacted to. This is left as a maybe because there are too many variables.
MS_SQL_Control_Overflow	yes	Vulnerability Research and Identification/Access Control Testing	Patching the systems would allow the environment not to be compromised by this worm.
ICMP_Flood	maybe	Access Control Testing/ Intrusion Detection System Testing	The firewall acl testing rule number one is to test the ACL against a "Deny all" rule, which means deny any that you do not specifically need. While there is no reason for ICMP from the Internet, the framework does not specifically mention that it is a bad idea. Also if the IDS was in place per "Intrusion Detection Systems Testing" this would be caught and reacted to. This is left as a maybe because there are too many variables.
ICMP_Smurf	maybe	Access Control Testing/ Intrusion Detection System Testing	The firewall acl testing rule number one is to test the ACL against a "Deny all" rule, which means deny any that you do not specifically need. While there is no reason for ICMP from the Internet, the framework does not specifically mention that it is a bad idea. Also if the IDS was in place per "Intrusion Detection Systems Testing" this would be caught and reacted to. This is left as a maybe because there are too many variables.
Windows_SMB_RPC_NoOp_Sled	yes	Vulnerability Research and Identification/Access Control Testing	Patching the systems would allow the environment not to be compromised by this worm.

		s Control Testing	
Windows_RPC_DCOM_Overflow	yes	Vulnerability Research and Identification/Access Control Testing	Patching the systems would allow the environment not to be compromised by this worm.
Windows_Registry_Access	unknown		No information could be ascertained regarding this signature.
SNMP_Protocol_Violation	no	Access Control Testing/ Intrusion Detection System Testing	The firewall acl testing rule number one is to test the ACL against a "Deny all" rule, which means deny any that you do not specifically need. While there is no reason for ICMP from the Internet, the framework does not specifically mention that it is a bad idea. Also if the IDS was in place per "Intrusion Detection Systems Testing" this would be caught and reacted to. This is left as a maybe because there are too many variables.
Limewire_File_Request	no	Security Policy Review	The best way to address this peer to peer file sharing is by a security policy. Second is the use of a proxy which examine traffic and blocks such peer to peer traffic. This framework mentions security policy, not specific to peer to peer but does not mention proxies.
IIS_CGI_Double_Decode	yes	Vulnerability Research and Identification/Access Control Testing	Patching the systems would allow the environment not to be compromised by this worm.
Q_Mail_Length_Crash	no	Access Control Testing/ Intrusion Detection System Testing	The firewall acl testing rule number one is to test the ACL against a "Deny all" rule, which means deny any that you do not specifically need. While there is no reason for ICMP from the Internet, the framework does not specifically mention that it is a bad idea. Also if the IDS was in place per "Intrusion Detection Systems Testing" this would be caught and reacted to. This is left as a maybe because there are too many variables.
General_Loki	yes	Internet Application Testing	Internet Application Testing should find this bug and flag it.
WWW_WinNT_cmd_DOT_exe_Access	yes	Internet Application Testing	Internet Application Testing should find this bug and flag it.
IIS_DotDot_Crash_Bug	yes	Vulnerability Research and Identification/Access Control Testing	Patching the systems would allow the environment not to be compromised by this worm.

Windows_RPC_DCOM_Overflow	yes	Vulnerability Research and Identification/Access Control Testing	Patching the systems would allow the environment not to be compromised by this worm.
IP_Fragment_Overwrite_Data_is_Overwritten	yes	Access Control Testing	Access control testing specifically mentions checking for this.
Jolt2_Fragment_Reassembly_DoS_attack	no	access control testing/Routing/Intrusion Detection	These two sections specifically mention checking how the firewall handles fragments. Uncertain if this testing would actually predict and mitigate upcoming fragment reassembly attacks. Also if the IDS was in place per "Intrusion Detection Systems Testing" this would be caught and reacted to.
Gnutella_Server_Reply	no	Security Policy Review	The best way to address this peer to peer file sharing is by a security policy. Second is the use of a proxy which examine traffic and blocks such peer to peer traffic. This framework mentions security policy, not specific to peer to peer but does not mention proxies.
Gnutella_Client_Request	no	Security Policy Review	The best way to address this peer to peer file sharing is by a security policy. Second is the use of a proxy which examine traffic and blocks such peer to peer traffic. This framework mentions security policy, not specific to peer to peer but does not mention proxies.
IIS_DotDot_EXECUT_E_Bug	yes	Internet Application Testing	Internet Application Testing should find this bug and flag it.
Windows_RPCSS_Overflow_2	yes	Vulnerability Research and Identification/Access Control Testing	Patching the system should protect from this vulnerability
TCP_SYN_Port_Sweep	yes	Access Control Testing	Access control testing specifically mentions checking for this.
IP_Fragments_overlap	yes	Access Control Testing	Access control testing specifically mentions checking for this.
URL_with_XSS	yes	Internet Application Testing	Internet Application Testing should find this bug and flag it.
Lotus_Domino_database_DoS	yes	Vulnerability Research and Identification	Patching the system should protect from this vulnerability
DNS_Zone_Transfer_High_Port	yes	Access Control Testing	Access control should only allow firewall rules what is expressly permitted and thus would deny a transfer on a dns high port.
WWW_General_cgi_bin_Attack	yes	Internet Application Testing	Internet Application Testing should find this bug and flag it.
WWW_IIS_Internet_Printing_Overflow	yes	Internet Application Testing	Internet Application Testing should find this bug and flag it.

WWW_IIS_Unicode_Attack	yes	Internet Application Testing/Vulnerability Research and Identification	Internet Application Testing should find this bug and flag it.
FetchMail_Arbitrary_Code_Execution	no	Intrusion Detection System Testing	If the IDS was in place per "Intrusion Detection Systems Testing" this would be caught and reacted to.
Route_Up	unknown		No information could be ascertained regarding this signature.
Orphaned_Fin_Packet	yes	Access Control Testing / Router Testing	Access control testing and router testing should catch this.
Ident_Improper_Request	no	Access control testing/IDS	Access control testing will catch the high port part of this signature but would not catch a large malformed url, that is the part of the signature that is not covered so the signature as a whole will be considered not covered.
Sendmail_Invalid_Recipient	no		This flags a   in an email packet. No test in the framework would catch this. There are circumstances where email could be filtered by trusted host but that also is not covered in the framework.
IIS_CGI_Double_Decompile	yes	Internet Application Testing/Vulnerability Research and Identification	Internet Application Testing should find this bug and flag it.
Root_exe_access	yes	Internet Application Testing	Internet Application Testing should find this bug and flag it.
IP_Fragment_Attack	yes	Access Control Testing / Router Testing	Access control testing and router testing should catch this.
Missed_Packet_Count	no	Access Control Testing / Router Testing	Access control testing or firewall could catch this but is not specifically mentioned
KaZaA_v2_UDP_Client_Probe	no	Security Policy Review	The best way to address this peer to peer file sharing is by a security policy. Second is the use of a proxy which examine traffic and blocks such peer to peer traffic. This framework mentions security policy, not specific to peer to peer but does not mention proxies.
BadBlue_File_Disclosure	yes	Internet Application Testing/Vulnerability Research and Identification	Internet Application Testing should find this bug and flag it.
Route_Down	unknown		
UDP_Bomb	yes	Access Control Testing / Router Testing	Access control and router testing includes forming malformed packets

Cisco_Securce_ACS_Directory_Traversal	yes	Vulnerability Research and Identification	This involves flagging a // in a http request for cisco acs. An upgrade to the access control server should fix this.
UDP_Flood	yes	Access Control Testing / Router Testing	Access control testing and router testing should catch this.
SMTP_AUTH_Brute_Force_Attempt	no	IDS	This would be caught by an application test or possibly by limiting traffic to a trusted source but essentially is not covered by the framework.
IIS_CGI_Double_Decode	yes	Internet Application Testing/Vulnerability Research and Identification	Internet Application Testing should find this bug and flag it. Patching would fix this bug as well.
IOS_HTTP_Unauth_Command_Execution	yes	Internet Application Testing/Vulnerability Research and Identification	Internet Application Testing should find this bug and flag it. Patching would fix this bug as well.
Back_Orifice_Ping	yes	Access Control Testing / Router Testing	These ports should be blocked and the access control check should flag this.