INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

**A Strategy for Defending
Against
Distributed Denial of Service Attacks**

by
David L. Whyte

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of
Master of Computer Science

Ottawa-Carleton Institute for Computer Science
School of Computer Science
Carleton University
Ottawa, Ontario, Canada

March 2002

© Copyright
2002, David L. Whyte

**National Library
of Canada**

**Acquisitions and
Bibliographic Services**

**395 Wellington Street
Ottawa ON K1A 0N4
Canada**

**Bibliothèque nationale
du Canada**

**Acquisitions et
services bibliographiques**

**395, rue Wellington
Ottawa ON K1A 0N4
Canada**

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-72082-9

Canada

The undersigned hereby recommend to the
Faculty of Graduate Studies and Research
Acceptance of the thesis.

A Strategy for Defending Against Distributed Denial of Service Attacks

submitted by

David L. Whyte

in partial fulfillment of the requirements
for the degree of Master of Computer Science

Director, School of Computer Science

Thesis Supervisor

Carleton University

March 2002

Abstract

Distributed Denial of Service (DDoS) attacks are one of the most serious threats facing the Internet today. DDoS attacks overwhelm a system or network by sending large amounts of network traffic, thereby exhausting the target's computational or communication resources. A single attacker, using a DDoS attack tool, is able to direct thousands of systems in a coordinated Denial of Service (DoS) attack against a target system.

In this thesis, we developed three DDoS resistant network architectures. These DDoS resistant network architectures were designed to: (1) secure the network against participation in a DDoS attack, and (2) maintain mission critical service delivery while a DDoS victim. In this study, the DDoS resistant network architectures are achieved by developing a strategy that utilized two distinct but related approaches: layered network security and survivability. Additionally, this study provides a comparative analysis of two DDoS tools: TFN2K and Plague.

Acknowledgements

When one undertakes an endeavor such as this, it is rarely a solitary journey. There are numerous people that have influenced and helped me along the way. I owe them all a debt of gratitude.

To my thesis advisor, Dr. Kranakis, I thank you for your support, guidance, and encouragement. You were always patient and truly interested in my work.

To my parents, Donald and Sharon, thank you for instilling in me the importance of education and for always believing in my abilities.

To all of my colleagues, I appreciate your advice, comments, and support. I extend a special thanks to you Donald. You have always provided me with excellent technical advice both professionally and academically.

Finally to my wife, Michelle, this would have not been possible without your support and encouragement. As I completed this work, you spent many hours proofreading and many more nights and weekends alone. In this, as in all things, you enable me to realize my potential.

Table of Contents

Abstract	iii
Acknowledgements	iv
Table of Contents.....	v
List of Tables.....	viii
List of Figures	ix
Chapter 1 Introduction.....	1
1.1. The Need for Security.....	2
1.2. The Threat of Distributed Denial of Service (DDoS) Attacks	8
1.3. Objectives	10
1.4. Overview of the Thesis Work	10
1.5. Thesis Contributions	12
1.6. Outline	12
Chapter 2 Background Information.....	13
2.1. IP Address Spoofing	13
2.1.1. Following the Journey of a Spoofed IP Datagram	15
2.1.2. TCP/IP Overview.....	17
2.1.2.1. Ethernet	18
2.1.2.2. Internet Protocol	18
2.1.2.3. Transmission Control Protocol.....	19
2.1.2.4. User Datagram Protocol	19
2.1.2.5. Internet Control Message Protocol	20
2.1.2.6. IP Datagram Structure	20
2.1.3. Tracing Spoofed IP Datagrams Back to the Source	26
2.1.4. Stopping IP Address Spoofing	29
2.2. What is a DoS Attack?.....	30
2.2.1. The Underlying Causes of DoS Attacks.....	30
2.2.2. DoS Attack Implementation and Evolution.....	32
2.3. What is Distributed Denial of Service (DDoS)?	35
2.3.1. DDoS Architecture.....	36
2.3.2. DDoS Attack Scenario	38
2.3.3. Attacks	41
2.3.4. Communication Methods	44
2.3.5. DDoS Tools Survey	46
2.4. Anatomy of a Network Attack	47
2.4.1. What is a Network Attack?	48
2.4.2. Stages of a Network Attack.....	49
2.4.2.1. Reconnaissance.....	50
2.4.2.2. Target Acquisition	57
2.4.2.3. Attack	58
2.4.2.4. Compromise and Implantation	60

2.4.2.5.	Data Acquisition and Exfiltration	62
Chapter 3	Examination of DDoS Tools	64
3.1.	Test Environment.....	64
3.2.	Testing Methodology	65
3.3.	TFN2K.....	66
3.3.1.	Overview	66
3.3.2.	TFN2K Installation	66
3.3.3.	Communication Characteristics	67
3.3.4.	Attacks	68
3.3.5.	Attack Options	69
3.4.	Plague	69
3.4.1.	Overview.....	69
3.4.2.	Plague Installation.....	70
3.4.3.	Communication Characteristics	71
3.4.4.	Attacks	71
3.4.5.	Attack Options	72
3.5.	Intrusion Detection Signatures.....	72
3.5.1.	TFN2K Intrusion Detection Signatures.....	73
3.5.2.	Plague Intrusion Detection Signatures	74
3.5.3.	Shortcomings of Intrusion Detection Signatures.....	75
Chapter 4	Current Network Defence Strategies	76
4.1.1.	Threat and Risk Assessment	76
4.1.2.	Survivable Network Systems	79
4.1.3.	Layered Network Security	82
4.2.	Proposed DDoS Defence Strategies.....	84
4.3.	DDoS Detection Tools.....	86
Chapter 5	DDoS Defence Strategies	87
5.1.	Problem Definition	87
5.2.	Designing a DDoS Resistant Network Architecture	88
5.3.	Securing the Network Against Participating in a DDoS Attack.....	89
5.3.1.	Network Security Overview	90
5.3.1.1.	Router.....	90
5.3.1.2.	Firewall.....	91
5.3.1.3.	Split DNS server.....	92
5.3.1.4.	Virtual Private Network (VPN).....	93
5.3.1.5.	Intrusion Detection Systems (IDS)	93
5.3.2.	Host Security Overview	96
5.3.2.1.	Limit Access to Network Resources	96
5.3.2.2.	Limit Resource Sharing	97
5.3.2.3.	Host Hardening.....	97
5.3.2.4.	TCP/IP Stack Hardening	99
5.3.3.	Network Security Posture Maintenance Overview.....	100
5.3.3.1.	Penetration Testing.....	100
5.3.3.2.	Update Systems	101
5.3.3.3.	Disaster Recovery Plan.....	101
5.3.3.4.	Logging Strategy	102
5.3.3.5.	Virus Scanning	103
5.3.3.6.	Backups	103

5.3.3.7.	Periodic Network Security Audits	104
5.4.	Maintain Mission Critical Service Delivery while a DDoS Victim	104
5.4.1.	Load Balancing Using Server Clusters	105
5.4.2.	Redundant ISP Connections.....	105
5.4.3.	Good Working Relationship with ISP.....	106
5.4.4.	Firewall Load Balancing	106
5.4.5.	Fail-Over Routers.....	107
5.4.6.	Additional Bandwidth	107
5.5.	DDoS Resistant Network Architectures – A Case Study	108
5.5.1.	Sample Network Requirements.....	109
5.5.2.	Network A.....	110
5.5.3.	Network B.....	115
5.5.4.	Network C.....	121
Chapter 6	Conclusions and Future Work.....	127
6.1.	Summary.....	127
6.2.	Conclusions.....	128
6.3.	Recommendations and Future Work.....	129
Acronyms.....		131
Appendix A TFN2K.....		133
1.	TFN2K Screen Shot	133
2.	TFN2K Network Data Captures	133
3.	TFN2K Communications.....	133
4.	TFN2K Attacks	135
Appendix B Plague		139
1.	Plague Screen Shot.....	139
2.	Plague Network Data Captures.....	139
3.	Plague Communications	139
4.	Plague Attacks	140
References.....		142

List of Tables

Table 1: CERT/CC Incident Statistics	3
Table 2: Evolution of DoS Attacks	34
Table 3: DDoS Tools Survey	46
Table 4: Stages of a Network Attack	49
Table 5: Nmap Scan Results	57
Table 6: DDoS Network Testbed Configuration	65
Table 7: TFN2K Attacks	68
Table 8: TFN2K Attack Options	69
Table 9: Plague Attacks	72
Table 10: DDoS Detection Tools	86
Table 11: Sample Network Requirements	109
Table 12: Network A – Security Domains	110
Table 13: Network A – Network Security Overview	111
Table 14: Network A – Host Security Overview	112
Table 15: Network A – Network Security Posture Maintenance Overview	113
Table 16: Network A – Availability Overview	113
Table 17: Network B – Security Domains	116
Table 18: Network B – Network Security Overview	117
Table 19: Network B – Host Security Overview	118
Table 20: Network B – Network Security Posture Maintenance Overview	119
Table 21: Network B – Availability Overview	119
Table 22: Network C – Security Domains	122
Table 23: Network C – Network Security Overview	123
Table 24: Network C – Host Security Overview	124
Table 25: Network C – Network Security Posture Maintenance Overview	124
Table 26: Network C – Availability Overview	125

List of Figures

Figure 1: Valid Source IP Address	16
Figure 2: Spoofed Source IP Address	17
Figure 3: IP Datagram Structure over Ethernet	21
Figure 4: Simplified Network Schematic	22
Figure 5: Datagrams Routed by Network A's Edge Router	23
Figure 6: Datagrams Routed by the ISP's Backbone Router	24
Figure 7: Datagrams Routed by the ISP's 2 nd Backbone Router	25
Figure 8: DDoS Tiers	37
Figure 9: DDoS Attack – Phase 1	39
Figure 10: DDoS Attack – Phase 2	40
Figure 11: DDoS Attack – Phase 3	41
Figure 12: DDoS Communications	44
Figure 13: Stages of a Network Attack	50
Figure 14: Screen Capture of Nmap	55
Figure 15: Network Schematic	56
Figure 16: TFN2K Test Network	67
Figure 17: Plague Test Network	70
Figure 18: Network A Architecture	114
Figure 19: Network B Architecture	120
Figure 20: Network C Architecture	126

Chapter 1 Introduction

The Internet is a complex and heterogeneous environment that enables information to flow between networks almost instantaneously. It is an inexpensive, efficient, reliable, and rapid communications medium. Internet usage is growing at an exponential rate as organizations, governments, and citizens continue to increase their reliance on this technology.

The Household Internet Use Survey, Electronic Commerce Release [1] prepared in July 2000 by Industry Canada, stated that 51% of Canadian households are now connected to the Internet. In a single year, between 1999 and 2000, Canadian household purchases over the Internet increased 264%. In terms of global connectivity, Canada rates only behind the United States in Internet usage and e-commerce adoption [2].

Canada's reliance on the Internet will continue to grow as the Government of Canada enters a time of unprecedented levels of networking. In the 37th Speech from the Throne, the Government of Canada set a goal of providing all government services to its citizens electronically by 2004.

“The Government has helped to make Canada one of the most connected countries in the world, yet the speed of change continues to accelerate. Canada must continue to develop and strengthen its information infrastructure. ... The Government will continue to work toward putting its services on-line by 2004, to better connect with citizens. [3]”

E-government by 2004 is indeed an aggressive goal. Providing government services electronically presents formidable technical challenges not only in regards to scalability, but also in the deployment and implementation of appropriate security countermeasures.

Government is not alone in realizing the benefits of the Internet. Private industry regards Internet connectivity as a business requirement. By 2005, business to business (B2B)

Internet commerce is projected to reach \$8.5 trillion [4]. The Internet allows organizations to conduct business in previously inaccessible markets thus creating a truly global economy. As society moves to unprecedented levels of connectivity, the need for security must be addressed. This is an area of deep concern to the public, government, and businesses. Despite these deep concerns, there is no planned or well-formulated security policy for the Internet.

1.1. The Need for Security

Citizens have an expectation of privacy and secure access to electronic services. Organizations are dependent on information technology. As their interconnectivity and interoperability increases, so too does their need for security. Network owners have a responsibility to secure their systems not only to protect their own networks, but also to prevent their systems from being used as part of a larger cyber attack.

Designing secure network architectures is a complex and challenging undertaking. The threats found on the Internet are as diverse as the networks that comprise it. Threats vary in expertise, resources, motivation, and access to specialized equipment. They can range from script kiddies [5] performing the equivalent of "digital spray painting" on the Internet to individuals working for well-funded nations involved in espionage.

"The CERT ® Coordination Center (CERT/CC) is a center of Internet security expertise, at the Software Engineering Institute, a federally funded research and development center operated by Carnegie Mellon University [6]."

CERT/CC is a clearinghouse for Internet security vulnerability information. In this capacity, it: issues computer security alerts and advisories, performs IT security research, offers training in computer security, and manages computer incidents. It also encourages organizations, government, academic institutions, and businesses to report computer incidents. CERT/CC can then provide their constituents with guidance on how to avoid,

minimize, and recover from potential threats. The focus of the organization is on the technical issues related to computer security and not on the location, motivation, and monitoring of intruders. In the course of managing computer incidents, CERT/CC has compiled statistics that reveal trends about malicious computer activity. The number of computer incidents reported to CERT/CC has tripled over the last two years [7].

Year	Incidents
1996	2,573
1997	2,134
1998	3,734
1999	9,859
2000	21,756
2001 Q1-Q3	34,754

Table 1: CERT/CC Incident Statistics

The honeynet project [8] is a computer security research project whose main objective is to understand the capabilities of computer hackers or blackhats.

“A blackhat is the attacker, someone attempting unauthorized access or activity with one of your resources [9]”.

The project consisted of deploying vulnerable systems on the Internet in the hopes of luring potential hackers. They did not advertise the existence or location of these systems nor did they issue penetration challenges. The systems were simply deployed and heavily monitored. This collection of systems is called a honeynet. The attacks, intrusions, and penetrations detected and then analyzed offer a glimpse into the capabilities, tools, motivations, and techniques of the blackhat community.

The honeynet project published their research findings that provided statistics about the

amount of malicious activity on the Internet and the capabilities of attackers.

“...we estimate the life expectancy of a default installation of a Red Hat 6.2 server to be less than 72 hours [10].”

Their study revealed that one could expect a vulnerable system connected to the Internet to be detected and then successfully attacked within 72 hours. Similar security studies and research initiatives have corroborated these findings.

In 1999, a security study was performed at San Diego's Supercomputer Center.

“The purpose of the experiment is to determine the “life expectancy” of a popular commercial operating system when attached to the public Internet [11].”

In late December 1999, a system with a default install of the Linux Red Hat operating system version 5.2 was connected to the Internet. In less than eight hours after the system was deployed, it was probed for Remote Procedure Call (RPC) vulnerabilities. Twenty-one days after installation the system experienced twenty unsuccessful attacks. These attacks failed because they were designed for systems running the Linux Red Hat operating system version 6.0. After forty days, the system was finally compromised using a Post Office Protocol (POP) vulnerability. Once the system was compromised, the attacker(s) installed a rootkit, a network sniffer, and then attempted to delete log files to hide their activity. A second intruder attacked the machine with another exploit and posted defaced web pages on the system's web server. No attempt was ever made to advertise, attract, or entice people to penetrate the system.

The amount of malicious activity on the Internet is both significant and increasing. This hostile environment can be attributed to a number of factors: the advancement of network exploitation tools, the Internet environment, increased complexity of networks, and the

attackers themselves as discussed in [12].

- Network exploitation tools:

- Increasing sophistication: Network exploitation tools are becoming increasingly sophisticated. Many tools build upon previous versions of the software both improving and adding functionality. Security countermeasures developed to neutralize previous versions of the tool are circumvented by newer versions of the tool. New network exploitation tools are also employing advanced techniques such as encryption and polymorphic code to avoid detection.
- Automation: Network exploitation tools have been developed that combine automated vulnerability scanning, dynamic exploit selection (based on scanning results), attacking, rootkit installation, and covert communication methods. Automated tools allow attackers to execute attacks on thousands of systems within minutes.
- Automatic propagation: Some network attacks only require human intervention for the initial launch of the attack. Attack scripts can then automatically launch new attacks unbeknownst to the system owner.
- Easy to use: The technical knowledge to develop and build network exploitation tools is significant. The creators must have a comprehensive knowledge of networking, networking protocols, vulnerability analysis techniques, and programming languages. While the skill and technical knowledge involved in developing these tools is significant, it typically requires very little technical skill to operate them. Some network exploitation tools are GUI and run on a Windows operating system. They come complete with tutorials, help files, and user mailing lists.
- Easy to find: Network exploitation tools are not kept and shared amongst a small community of skilled developers. These tools are found on hundreds of well-visited blackhat and security websites and distributed under the guise of

“educational purposes”. These tools are also distributed via Internet Relay Chat (IRC) rooms, mailing lists, and through news groups.

- The Internet environment:
 - Increasingly complex environment: The Internet accommodates a myriad of applications and computing platforms. Each newly developed protocol, product, service and technology implementation is layered on top of the existing infrastructure. Administrators face the daunting task of maintaining and understanding an ever changing and increasingly complex operational environment in order to properly secure their systems.
 - The number of users is increasing: The estimated number of worldwide users of the Internet as of August 2001 was approximately 513 million. One year earlier, the estimated number of worldwide users of the Internet was approximately 369 million [13].
 - Lack of central governance and policing: The Internet is ubiquitous and cannot be thought of in the traditional paradigm of political and geographic boundaries. No single authority owns the Internet. Information passed from one network to another may pass through several geopolitical zones. Each zone may have its own respective cultural, language, and criminal law differences.
 - Anonymity: The Internet infrastructure as it exists today allows communication without authentication. This enables a practice known as IP address spoofing. A sender can specify a bogus IP address when sending communications. Determining the true source of spoofed IP datagrams can be a difficult and sometimes impossible task.
 - Logical connectivity: The Internet paradigm is best described as everything connected to everything else. Telephone infrastructures, utilities, mobile devices, and household products all form the unbounded network we call the Internet.

- Increased complexity of networks:
 - Mobile work force: The modern workforce is mobile. Workers require access to a corporate network when traveling on business or when working from home. To provide security for such an endeavor a Virtual Private Network (VPN) solution is often used. VPNs allow secure remote access into the network so internal resources such as email, file servers, and internal websites can be accessed. It is an effective means of increasing productivity, but it blurs the network boundary. Accommodating a mobile workforce extends the network boundary both logically and physically making security more challenging to implement.
 - Business integration (trusted partners): A good security posture dictates that countermeasures should be deployed at all access points into the network. Trusted business partners often access networks from a leased or shared line. There may be a temptation to deploy less security at this access point because it connects to a trusted partner. This trust may be misplaced unless you are confident that the security posture of your business partner is at least equal to your own. If a successful network attack occurs on your business partner's network, the only thing standing between the attacker and your network is the countermeasures deployed at the business partner's access point.
 - Technology advances and integration: As technology advances and new "killer-apps" are developed, there is pressure from users to implement the latest technology. Unfortunately, quickly adopting the newest technology is often done at the expense of IT security.
 - Reliance on networks: Internet connectivity has grown into an essential part of most organization's mission critical functions. Network downtime often translates into thousands of dollars lost in business sales and opportunities.
- Attackers:
 - Information sharing: Traditionally the intruder community has been very skilled at

sharing information. Organizations are typically reluctant to share information about vulnerabilities and incidents as disclosure may bring loss of consumer confidence. Attackers only have to find one security hole to get in, but defenders have to close all holes to keep the attackers out.

1.2. The Threat of Distributed Denial of Service (DDoS) Attacks

DDoS attacks are one of the most serious threats facing the Internet today. A single attacker using a DDoS attack tool is able to direct thousands of systems in a coordinated DoS attack against a target system. DDoS attacks overwhelm a system or network by sending large amounts of network traffic, thereby exhausting the target's computational or communication resources. To exacerbate the problem, the network traffic is spoofed making it difficult to locate the attacking systems.

DDoS tools are becoming increasingly sophisticated and designed to overcome network security countermeasures. The latest DDoS tools make use of: encrypted one-way command and control communications, automated code updates, and a variety of self-protection mechanisms. These tools have the capability to evade detection and implement a variety of attacks.

Implementing an effective network security posture to defend against DDoS attacks is a challenge. The best network security practices may protect your network from being a source of a DDoS attack, but it will not stop you from being a victim of a DDoS attack. As long as there are weakly defended systems on the Internet, the potential for DDoS attacks exists. If all network owners implemented a strong network security posture, the amount of systems compromised would be reduced, as would the number of DDoS attacks. Unfortunately, relying on the security of other networks is not practical. It is

infeasible to secure the entire Internet; therefore, DDoS attack mitigation strategies need to be put in place for protection.

In February 2000, DDoS attacks were launched on several large and highly visible Internet portals such as Yahoo, Amazon, Etrade, and CNN [14, 15]. The flood of network traffic generated by the DDoS attack overwhelmed the network infrastructure of these portals denying access to thousands of users for hours. The Yankee Group estimates the cumulative cost of the February DDoS attacks to be approximately 1.2 billion dollars [16]. These large e-commerce sites plan and expect to service thousands of user requests simultaneously. These attacks were successful despite the fact that some of these portals employed availability technology such as: server farms, load balancing, redundant network devices, and high bandwidth capabilities to ensure legitimate users access.

The key to a successful DDoS attack is the number of systems participating in the attack. The more systems involved in the attack the greater the impact. Non-distributed denial of service attacks that rely only on resource exhaustion are usually not successful. If a single system attempted these types of flooding attacks, it would have little to no appreciable effect on a network. Even the most security aware organizations can fall prey to these types of attacks. The CERT/CC website was disrupted over a three-day period by a DDoS attack in May 2001 [17, 18]. This incident was of particular concern because the Internet community relies on CERT/CC to provide it with the latest computer vulnerability information.

At first consideration, one may be tempted to focus on technologies that safeguard availability as a means to defend against DDoS attacks. However, it quickly becomes apparent that all security services must be addressed. Lack of service availability is the symptom and not the cause of DDoS attacks. A DDoS resistant network architecture must protect network assets against compromise and subsequent use in a coordinated

attack. Additionally, it must implement the necessary technical and non-technical countermeasures to effectively manage a DDoS attack with an overall objective to deliver uninterrupted mission critical services.

1.3. Objectives

The primary purpose of this study was to develop practical DDoS-resistant network architectures. Drawing upon varying requirements specified in a case study, three different DDoS resistant network architectures were developed. In the course of producing these network architectures, a number of specific sub-goals were included in this study:

- To develop a network design strategy to counter DDoS attacks.
- To implement this strategy in the form of practical DDoS resistant network architectures.
- To provide guidance on the use of security technologies and availability strategies that could be applied to any network regardless of site specific requirements.
- To experiment with two DDoS tools to better understand their capabilities and limitations.
- To observe network traffic patterns caused by the two DDoS tools in order to create intrusion detection signatures.

1.4. Overview of the Thesis Work

The primary effort of this thesis work was to develop DDoS resistant network architectures. This undertaking allowed us to explore a variety of topics such as:

- The level of malicious activity on the Internet
- The difficulties in determining the true source of spoofed Internet Protocol (IP) datagrams
- How network attacks occur

- How DDoS attacks occur
- A detailed examination of two DDoS tools
- The development of a network design strategy to counter DDoS attacks
- Network security technologies
- Availability strategies

The DDoS resistant network architectures developed in this study are a result of utilizing two distinct but related approaches to provide network security. These two approaches are layered network security and survivability. Layered network security seeks to improve the overall security of a network by applying layers of security thus removing single points of failure. Survivability is the ability of a system to deliver mission critical functions in the event of system attacks, accidents, or failures. The resulting network design strategy secures the network against participating in a DDoS attack and maintains mission critical service delivery while a DDoS victim.

Included with the DDoS resistant network architectures are descriptions of the applicable network security countermeasures and availability strategies. These descriptions provide a framework that can be applied to a variety of networks regardless of site specific operational requirements.

The work in this thesis allowed us to examine in detail two DDoS tools, TFN2K and Plague. These tools were tested to discover any observable patterns in network traffic. These patterns are used by intrusion detection systems (IDS) to detect DDoS activity on a network.

1.5. Thesis Contributions

The three DDoS resistant network architectures developed in this study are complete network schematics. The architectures can be used to implement fully functional DDoS resistant networks for organizations with similar operational requirements.

The security technologies and availability strategies used to build the DDoS resistant network architectures are accompanied by significant guidance. This guidance allows the reader to apply the DDoS mitigation strategy to networks of varying operational environments thus extending the three network architectures developed in this study.

The comparative analysis of the DDoS tools: TFN2K and Plague, gives insight into the differing capabilities and limitations of this type of network attack tool. The examination allowed us to observe patterns in network traffic that can be used to detect a DDoS tool's activity on a network.

1.6. Outline

This thesis is organized as follows. Chapter one is an introduction to the topic of network security and DDoS. Chapter two provides the reader with background on DDoS and malicious network activity. Chapter three is a detailed analysis and comparison of two DDoS tools. Chapter four contains current approaches on secure network design and recommendations on how to defend against DDoS attacks. Chapter five contains a case study and a network design strategy to mitigate the effect of DDoS attacks. Finally, the thesis concludes in chapter six.

Chapter 2 Background Information

DDoS attacks are one of the most serious and sophisticated threats that currently face the Internet. Understanding how these attacks are orchestrated and the ability to properly evaluate proposed network defence strategies requires knowledge of a variety of concepts. The reader must be familiar with TCP/IP protocol suite fundamentals and also understand basic networking theory. Additionally, a solid working knowledge of computer security basics is required such as:

- DoS
- DDoS
- IP address spoofing
- How network attacks occur

This chapter provides the reader with the necessary information to understand how DDoS attacks occur. It will reveal weaknesses found in the Internet infrastructure that allow these attacks to occur with relative anonymity. It will also explore DDoS attacks in detail and reveal the methodology used by malicious individuals to attack and compromise vulnerable computer systems.

2.1. IP Address Spoofing

In order to send and receive data across the Internet, a universally agreed upon set of protocols is required to govern these activities. Although there are a number of communication protocols supported by the Internet, the principle method of communication is the TCP/IP protocol suite [19, 20]. The TCP/IP protocol suite is used to specify how data is transmitted across packet switched networks over the Internet.

The TCP/IP protocol suite uses numeric identifiers called IP addresses to uniquely identify computers on a network. Typical network communications such as email, file

transfers, or web page requests all require the use of valid source and destination IP addresses. The source IP address identifies the data sender and the destination IP address identifies the data receiver. The basic unit of data transfer in a packet switched network is called an IP datagram. The terms: packet, IP datagram, and datagram are often used interchangeably. Datagrams are composed of two basic components: a header portion that contains the necessary information to route the datagram and a payload portion that contains the data [21]. Two 32-bit fields in the datagram header contain the source and destination IP addresses.

The TCP/IP protocol suite has proven to be an effective communication method, but it does contain inherent security flaws. Steven Bellovin's paper entitled, *Security Problems in the TCP/IP Protocol Suite*, describes some of these flaws in detail [22]. A number of these flaws exist because some systems rely on source IP addresses as a means of authentication. Access to a system or services provided by a system is decided based on the claimed source IP address contained in the datagram. This means of authentication has proven to be unreliable. There have been many documented cases of intruders spoofing source IP addresses to gain unauthorized access to systems and services. IP address spoofing can be defined as the intentional misrepresentation of the source IP address in an IP datagram in order to conceal the identity of the sender or to impersonate another computing system. The US Department of Energy Computer Incident Advisory Capability (CIAC) [23] and CERT/CC [24] have both released advisories that define IP address spoofing in the context of gaining root or privileged access to a system by an intruder impersonating a local system's IP address. These advisories explain how systems that rely on session based IP address authentication are susceptible to IP address spoofing which, if successful, can allow an intruder to gain unauthorized access to a system or services.

In addition to using spoofed IP addresses as a means to gain unauthorized access to a system, it can also be used to conceal the identity of the datagram sender. CERT/CC has issued advisories about the use of spoofed IP addresses as a method to conceal an attacker's identity during flooding attacks [25] and to purposely cause systems to crash [26], both resulting in successful DoS attacks.

IP address spoofing is possible because the network devices, called routers, which provide connectivity between individual networks, only require inspection of the destination IP address in the datagram to make routing decisions. The source IP address is not required by routers and an invalid source IP address will not affect the delivery of datagrams. Spoofed IP addresses are an effective way to conceal an attacker's identity during a DoS attack. When a DoS attack is launched against a system, the attacker is not interested in retrieving any information from the intended victim and thus the return or source IP address can be spoofed. In a DoS attack, the overall goal is to deny the use of services the system provides to authorized users without being discovered. One class of DoS attacks includes sending large volumes of network traffic to a system until it becomes overwhelmed and cannot respond to new requests for service. This type of DoS attack is successful because computational resources are finite and every honored spurious request means one less legitimate request that can be serviced. Typically, the datagrams that comprise the flooding attack contain spoofed datagrams so that determining the source of the attack will be more difficult.

2.1.1. Following the Journey of a Spoofed IP Datagram

Figure 1: Valid Source IP Address, illustrates a typical interaction between a workstation with a valid source IP address requesting web pages and the web server executing the requests. When the workstation requests a page from the web server, the request contains both the workstation's IP address (i.e. source IP address 192.168.0.5) and the address of

the web server executing the request (i.e. destination IP address 10.0.0.23). The web server returns the web page using the source IP address specified in the request as the destination IP address (192.168.0.5) and its own IP address as the source IP address (10.0.0.23).

Figure 1: Valid Source IP Address

Figure 2: Spoofed Source IP Address, illustrates the interaction between a workstation requesting web pages using a spoofed source IP address and the web server executing the requests. If a spoofed source IP address (i.e. 172.16.0.6) is used by the workstation, the web server executing the web page request will attempt to execute the request by sending information to the IP address of what it believes to be the originating system (i.e. the workstation at 172.16.0.6). The system at the spoofed IP address will receive unsolicited connection attempts from the web server that it will simply discard.

Figure 2: Spoofed Source IP Address

Basic networking concepts are discussed in the following section in order to provide the necessary context to discuss the technical aspects of IP address spoofing. Locating the origin of DoS attacks is very complex when IP address spoofing is used.

2.1.2. TCP/IP Overview

The Open Systems Interconnection Reference Model (OSI model) groups network: protocols, communications, and applications into seven distinct layers [27]. It is a model that is used to understand how protocols or applications can work to interconnect networks. The model represents all the specifications, functions, and activities that need to occur for successful networking. Each layer in the model represents a group of related: functions, specifications, and activities. TCP/IP is not a single protocol but rather a suite of protocols. It is comprised of the Internet Protocol (IP), Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and the Internet Message Control Protocol (ICMP). Using a suite of protocols, rather than a single protocol, simplifies the design and implementation of the hardware and software that allow computing platforms to be

connected. For a detailed explanation of the exact format and description of the TCP/IP protocol suite, please refer to RFC 791 [20] and RFC 793 [19].

2.1.2.1. Ethernet

The TCP/IP protocol suite governs how data is transported across networks from host to host, but does not specify how data is transmitted across different physical media. Layer 2 of the OSI model governs how the raw signals on a physical line are interpreted and converted into bits and then organized in frames for transport. To transmit these frames over physical media most networks use Ethernet. Ethernet is an IEEE 802.3 series standard that specifies how two or more systems sharing a common cabling system can interact [28]. Ethernet uses its own addressing scheme that consists of a unique 48-bit number. This address is known as the Media Access Controller (MAC) address and is assigned to network interface cards (NIC) by the manufacturer. Each manufacturer is assigned a block of these addresses by the IEEE Registration Authority [29]. MAC addresses encapsulate datagrams transmitted over networks that use Ethernet.

2.1.2.2. Internet Protocol

IP is a network layer protocol that is used to deliver packets across connected networks to their intended destination. IP operates at layer 3 of the OSI model, the network layer. IP is connectionless. It does not generate or maintain state information about the datagrams during delivery. The implication of connectionless delivery is that the delivery of datagrams is unreliable and only transported on a best-effort basis. Datagrams may arrive out of sequence, be delayed, duplicated, or not arrive at all [20]. Additionally, since each datagram is delivered independently of the others, they may not travel along the same route to their intended destination. This best-effort delivery scheme allows IP to have the necessary level of abstraction to deliver datagrams over the heterogeneous environments they may encounter on the way to their destination.

Although IP provides the necessary functionality to transmit packets over heterogeneous computing platforms and environments, it does not provide the necessary services for reliable datagram delivery. This best-effort delivery scheme, utilized by IP, presents a problem for applications that rely on the integrity of the data they transmit or receive.

2.1.2.3. Transmission Control Protocol

TCP operates at layer 4 of the OSI model, the transport layer. TCP provides the necessary functionality to guarantee reliable datagram delivery. In contrast to IP, TCP provides reliable connection-based data stream delivery that ensures error-free, sequential, and non-duplicated packet delivery. This is achieved by the use of a number of techniques such as: flow control mechanisms, sequence numbers, required acknowledgements, and adaptive retransmission techniques [30]. TCP requires that the data sender and receiver establish a connection before the data can be sent and received.

2.1.2.4. User Datagram Protocol

UDP like TCP operates at layer 4 of the OSI model. UDP provides the necessary functionality to establish datagram delivery between two computing systems. However, like IP, UDP datagrams may arrive out of sequence, be delayed, duplicated, or not arrive at all [21]. UDP depends upon the IP protocol to move the datagrams that it produces. UDP is typically used for applications where speed of data transmission is a priority and integrity is not critical for proper functioning. Examples of some of the uses of UDP are streaming audio and video.

2.1.2.5. Internet Control Message Protocol

ICMP is used to send messages between computing systems for diagnostic or management purposes [20]. Most ICMP is used internally by IP applications, but there are some applications that use its functionality. A popular example of this is ping. Ping is a program included with most operating systems and uses ICMP echo request datagrams to allow a user to determine if a host is active on a network. ICMP depends upon IP to move the datagrams that it produces.

2.1.2.6. IP Datagram Structure

Figure 3: IP Datagram Structure over Ethernet illustrates the anatomy of a typical IP datagram that traverses over Ethernet. Like the layers of an onion, a datagram's fields can be stripped away until just the data portion remains. Network applications generate datagrams by encapsulating data with protocol information. The protocol information appended to the data field is called the datagram header. The data portion is first encapsulated by the desired transport protocol, then the IP protocol, and finally by the MAC address.

Figure 3: IP Datagram Structure over Ethernet

To understand how to trace spoofed datagrams to their origin, it is necessary to understand how they travel through networks from source to destination. Network A and B, as shown in Figure 4: Simplified Network Schematic, connect to the same Internet backbone. Their network boundaries can be defined as the interfaces on their routers that receive and transmit information from their respective networks to the Internet Service Provider's (ISP) backbone.

Figure 4: Simplified Network Schematic

An ISP backbone is a high-speed data transmission line that connects networks to the Internet. When systems in Network A communicate with systems in Network B, their

datagrams pass through four routers. Routers 1 and 4 are the edge routers of the respective networks and routers 2 and 3 are the two ISP backbone routers. When a system on Network A transmits a datagram, it encapsulates it with the MAC address of the NIC installed on the system. When the datagram reaches the edge router of Network A (i.e. Router - 1), the router inspects the datagram and replaces the current MAC address with its own MAC address (i.e. 0050DAC6C95F). Then Router - 1 forwards the datagram to Router - 2 (see Figure 5: Datagrams Routed by Network A's Edge Router).

Figure 5: Datagrams Routed by Network A's Edge Router

Further inspection of the colored datagram reveals that the destination IP address of 24.12.36.5 is a valid IP address within Network B. The source IP address of 121.34.5.171 is not part of the valid IP address range of Network A. Valid source IP addresses for Network A fall within the range of 216.34.12.0 to 216.34.12.255 inclusively. Therefore, this datagram is spoofed. The backbone ISP router (i.e. Router - 2 in Figure 6: Datagrams Routed by the ISP's Backbone Router) receives the datagram from Network A's router, checks the destination IP address, replaces the MAC address with its own MAC address (i.e. 0232FE07A123), and then forwards the datagram to Router - 3.

Figure 6: Datagrams Routed by the ISP's Backbone Router

Router – 3 repeats this process and forwards the datagram to the edge router of Network – B. Router – 4 (see Figure 7: Datagrams Routed by the ISP's 2nd Backbone Router). Router – 4 then forwards the datagram to Network B's firewall. Finally, Network B's firewall inspects the datagram and sends it to the system with the IP address of 24.12.36.5.

Figure 7: Datagrams Routed by the ISP's 2nd Backbone Router

Even though the datagram contained an invalid source IP address, it was still able to arrive at the intended destination. Routers only need to inspect the destination IP address to determine the best path for efficient delivery. The source IP address does not have to

be examined. The previous example made use of a simplified network architecture to illustrate how datagrams are moved between networks. In practice, tracing the path of spoofed IP datagrams can be a difficult endeavor.

2.1.3. Tracing Spoofed IP Datagrams Back to the Source

Tracing spoofed IP datagrams back to the true source presents a number of challenges that are both technical and non-technical in nature. Currently, the best method to determine the true source of a spoofed datagram is to manually trace the path of the datagram from router to router until you reach the network of origin.

As stated in a Cisco technical notes document,

“The only reliable way to identify the source of an attack is to trace it back hop-by-hop through the network. This process involves reconfiguring routers and examining log information, and therefore requires cooperation by all network operators along the path from the attacker to the victim [31].”

This is not a trivial undertaking as datagrams typically pass through several routers before they reach their destination. Even datagrams sent from the same source and going to the same destination do not have to follow the same path; they may pass through different routers. By inspecting the MAC address of the incoming datagrams, you can determine the location of the forwarding router or last hop. This assumes that the router is logging and recording the MAC address of the incoming datagrams. As shown in the previous section, regardless of the source IP address in the datagram being spoofed, the MAC address of the datagram source has to be correct. If you were able to retrieve the MAC address from every single hop between the source and destination systems, you could trace the datagram's network of origin.

Following the path of an datagram through a network using only MAC addresses supplied by router logs presents some interesting difficulties as discussed in [32, 33]:

- It is a slow and manual process: Responding to a computer incident in a timely fashion is critical to limiting damage to systems. Harvesting logs from each router along the transmission path and examining specific datagrams takes time. Network personnel for each router have to be contacted and be willing to send the relevant log information. The time required to trace the spoofed datagrams will be significant. The logs of several routers may have to be examined and the results of the search verified through correlation.
- Reliance on upstream router personnel: A successful investigation to determine the source of spoofed datagrams will depend on the co-operation of every ISP or administrator in the path between the sender and receiver. Failure to provide all of the necessary information will make tracing the origin of the spoofed datagram impossible. Even if all router administrators are willing to cooperate, they may not have the technical skill or time required to assist in the investigation. Interpreting router logs and enabling logging on router interfaces requires some measure of technical skill.
- The system sending the spoofed datagrams may be compromised: If you are able to trace the datagrams back to an actual system, it may itself be the victim of an attack. However, the owner of the system may be unaware that they have been successfully compromised and that their system is being used to send spoofed datagrams as part of a denial of service attack. You may succeed in stopping the flow of spoofed datagrams from this source, but you would not have discovered the identity of the attacker.
- Multiple sources of spoofed datagrams: Tracing the path of spoofed datagrams is a labor intensive undertaking. There may be multiple sources of incoming spoofed datagrams. Each source of spoofed datagrams will mean a separate investigation as the datagrams may take different routes to the system.

- Routers may not be logging: Not all organizations will have logging enabled on their routers. An organization could be using older routers that do not support the logging required or there could be significant performance issues when logging is enabled. It is not uncommon for datagrams to travel through nine or more intermediate hops until they reach their destination. Failure of just one router to log will make tracing the spoofed datagrams impossible.
- Network traffic bursts or variances: If the incoming flow of spoofed datagrams is stopped periodically or is sent from different systems at different times, tracing them to the source will be problematic. If the source of the datagrams currently being traced stops sending spoofed datagrams, then determining the last hop point may not be possible. Without the flow of spoofed datagrams to guide the way, the MAC addresses will have to be retrieved exclusively from the logs of all routers between the source and destination. If the appropriate logging was not enabled on even a single router, the investigation time devoted to that flow will have been wasted.
- Crossing jurisdictions/national boundaries: Tracing spoofed datagrams relies on being able to communicate and gain cooperation from network administrators responsible for the routers between you and the source of spoofed datagrams. The Internet knows no geographical boundaries and datagrams may traverse a number of network boundaries in different countries on the way to their destination. Assistance from network administrators will typically be on a voluntary basis. In those cases where law enforcement becomes involved, warrants may be issued to compel ISPs to reveal their log files for the purpose of tracing network attacks. These writs may prove ineffective in getting information from countries that do not recognize their authority.
- Language barriers: Tracing spoofed datagrams relies on being able to communicate and gain cooperation from network administrators responsible for the routers between you and the source of spoofed datagrams. If you can not communicate with the network administrators, it will be hard to let them know what information is required.

- **Network failure:** A successful DoS attack causes the target system, or the network infrastructure that the system relies on, to become overwhelmed and fail. If the router has failed as a result of the DoS attack, harvesting its logs may be problematic.

2.1.4. Stopping IP Address Spoofing

The router that connects a network to another network is known as a border router. One way to mitigate the threat of IP spoofing is by inspecting datagrams for invalid source IP addresses as datagrams leave and enter a network. If this type of filtering were performed on all border routers, IP address spoofing would be greatly reduced.

Egress filtering checks the source IP address of datagrams to ensure they come from a valid IP address range within the internal network [34]. When the router receives a datagram that contains an invalid source address, the datagram is simply discarded and does not leave the network boundary. However, this type of filtering may not prevent a system from participating in a DoS attack as the spoofed IP address used could fall within the valid internal address range. However, since the system will have to use a source IP address within the valid IP range of the network, it will simplify the process of tracing spoofed datagrams.

Ingress filtering checks the source IP address of datagrams that enter the network to ensure they do not come from sources that are not permitted to access the network [35]. At a minimum, all private, reserved, and internal IP addresses should be discarded by the router and not allowed to enter the network.

2.2. What is a DoS Attack?

The objective of a DoS attack is to impair or render services offered by a system inaccessible by legitimate users. CERT/CC defines a denial of service attack as:

“A “denial of service” attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. Examples include

- attempts to “flood” a network, thereby preventing legitimate network traffic
- attempts to disrupt connections between two machines, thereby preventing access to a service
- attempts to prevent a particular individual from accessing a service
- attempts to disrupt service to a specific system or person [36]”.

DoS attacks are designed to adversely impact service delivery in an effort to cause a significant reduction in quality of service or to bring about total service delivery failure. Unfortunately, DoS attacks are a frequent occurrence on the Internet. The Computer Security Institute and the FBI produce and jointly release the Computer Crime and Security Survey. This survey is designed to raise the level of Information Technology (IT) security awareness and help determine the extent of computer crime in the United States. In the 2001 Computer Crime and Security Survey, 38% of respondents reported detecting DoS attacks against their systems. In a separate initiative, the University of California undertook a study to estimate the amount of DoS activity occurring on the Internet. Their study consisted of collecting and analyzing network traffic on a quiescent Class A network. Using a new technique, called backscatter analysis, the researchers observed 12,000 DoS attacks against 5,000 distinct targets in three weeks [37].

2.2.1. The Underlying Causes of DoS Attacks

The *Trends in Denial of Service Attack Technology* paper [38] produced by CERT/CC states that all Internet connected systems face a real and constant threat from DoS attacks

because of two fundamental characteristics of the Internet:

- The Internet is composed of finite resources: The entire Internet infrastructure is comprised of finite and thus exhaustible resources. Processing power, bandwidth, disk, and storage are all valid targets for DoS attacks.
- Internet security is highly interdependent: DoS attacks are typically launched from systems external to the victim's own system or network. The attacking systems are usually compromised in an effort to protect the identity of the attacker. If more care were taken to secure systems on the Internet, fewer compromises would occur and fewer systems would be available to participate in DoS attacks. Unfortunately, even the best security precautions only protect you from being a participant in a DoS attack and not from being the victim.

One could argue there is a third characteristic of the Internet that also contributes to the threat of DoS attacks:

- The proliferation of homogeneous technology: The technology landscape of the Internet is uniform. The Internet will continue to remain homogeneous as long as only a few large technology vendors such as: Cisco, Sun, Red Hat and Microsoft continue to dominate the technology marketplace. Although homogeneous technology leads to network and system administration cost savings for organizations, it also fosters an environment where security vulnerabilities pose a greater threat. Security vulnerabilities can have a severe impact in a homogenous environment. The discovery of a single exploitable vulnerability can place the entire Internet at risk. An example of this scenario is the Code Red worm [39, 40]. The Code Red worm appeared in July of 2001 and exploited a vulnerability in Microsoft's IIS web servers. Within 14 hours, on July 19th, 2001, 359,000 Windows systems were infected with a variant of the worm (i.e. Code Red v2). At its peak, Code Red was able to infect 2,000 systems a minute [41].

2.2.2. DoS Attack Implementation and Evolution

There are a number of ways that DoS attacks can be implemented as discussed in [36] such as:

- Resource exhaustion:
 - Computational resources: Systems have a finite amount of resources that they use to manage all resource requests. These resources may take the form of: kernel processes, CPU cycles, network data structures, disk space, or process state tables. Every granted resource request means fewer resources available for subsequent resource requests.
 - Flooding: Sending a flood of bogus IP datagrams to a system reduces bandwidth for legitimate network traffic and exhausts networking applications.
- Implementation flaws:
 - Protocol weaknesses: Examination of Request for Comments (RFCs) for protocols may reveal implementation flaws that can introduce exploitable vulnerabilities.
 - Malformed datagrams: Vulnerabilities in the way that some TCP/IP network stacks handle and interpret non-standard network traffic may cause a system to become unresponsive or to crash.
- Physical destruction: If physical access to computer assets is accomplished, hard drives may be physically destroyed or files deleted.

DoS attacks have been around for many years and are evolving over time. At one time, network-based DoS attacks consisted of one system sending malformed IP datagrams to another system. The receiving system, unable to interpret the malformed IP datagrams, would either reboot or shutdown causing a DoS attack [26]. Crafting malformed IP datagrams to successfully execute this type of DoS attack requires considerable technical skill. Only by careful analysis of networking protocols and applications can an

exploitable vulnerability be discovered that can produce the desired effect. In recent times, these types of DoS attacks have been replaced by remote, highly distributed, and coordinated DoS attacks [38]. A single attacker controlling hundreds or possibly thousands of systems sends a flood of spurious network traffic until the target becomes overwhelmed and unresponsive to legitimate user requests. This type of DoS attack is called a Distributed Denial of Service (DDoS). DoS evolution can be characterized as follows [42]:

- Resource exhaustion attack: Locally requesting an inordinate amount of finite resources from a system until resource exhaustion occurs.
- Remote resource exhaustion attack: Remotely requesting an inordinate amount of finite resources from a system until resource exhaustion occurs.
- Remote implementation flaw attack: Remotely exploiting a configuration or software implementation flaw causing system or application failure.
- Combination attack: Combining a number of DoS attacks into one attack.
- Distributed attack: Using numerous distributed systems in a coordinated remote resource exhaustion attack.

Examples of some of the more popular DoS attacks in their respective categories are listed in Table 2: Evolution of DoS Attacks. All of the programs that implement these attacks are available for download at [43].

Resource exhaustion attack	
Attack	Description
Fork() Bomb	A process that recursively spawns copies of itself until it fills the process table and no other processes can be spawned.
Recursive Directory Attack	It creates a series of recursive directories until they cannot be deleted. The Windows FAT file system was susceptible to this type of attack.
Remote resource exhaustion attack	
Attack	Description
Bonk	A modified version of the teardrop (see below for description) attack. The attack sends corrupt UDP packets to port 53 (DNS).
Pepsi	It sends a flood of UDP datagrams to specified ports.
Smurf	It spoofs ICMP echo requests from a system to various broadcast addresses (typically active in routers). This results in echo replies to that system from all systems in the broadcast address range.
UDP Flood	It sends a flood of UDP datagrams to either the small services ports (i.e. chargen and echo) or random ports.
TCP SYN Flood	It sends a flood of half open TCP connection requests to a system.
Remote implementation flaw attack	
Attack	Description
Land	It uses a forged IP datagram that specifies the same source and destination IP address.
Teardrop	This is a fragment reassembly attack for Windows operating systems.
Jolt	This is a ICMP echo fragmentation attack.
Nestea	This is similar to the teardrop, except it affects Linux operating systems.
Newtear	It sends a malformed UDP datagram.
Boink	It manipulates the fragment offset field in the TCP/IP header, causing the target system to reassemble a datagram that is too large to reassemble.
Combination DoS attack	
Attack	Description
Targa	This is a tool that launches: Bonk, Jolt, Land, Nestea, Newtear, SYNdrop, Teardrop, and Winnuke attacks.
Distributed DoS attack	
Attack	Description
Trinoo	Also known as Trin00, it utilizes UDP flooding attacks.
Plague	Contains only two types of attacks: SYN and Stream flood.
TFN	Tribe Flood Network flooding attacks include: UDP, TCP, ICMP, and Smurf.
TFN2K	Tribe Flood Network 2000 flooding attacks include: UDP, TCP, ICMP, and Smurf. It also can execute the combination attack Targa.
Stacheldraht	This is the German word for barbed wire. Its flooding attacks include: UDP, TCP, ICMP, and Smurf.

Table 2: Evolution of DoS Attacks

2.3. What is Distributed Denial of Service (DDoS)?

DDoS attacks are the most sophisticated form of DoS attacks that exist today. A single attacker using a DDoS attack tool is able to direct thousands of systems in a coordinated DoS attack against a target system. These tools can support a variety of remote resource exhaustion and remote implementation flaw attacks to overwhelm a targeted system. DDoS attacks pose one of the most serious threats to the proper functioning of the Internet. However, work has been undertaken to understand and develop effective countermeasures against this threat.

The CERT/CC has released a number of alerts and advisories concerning the capabilities and countermeasures for DDoS tools [44, 45]. Additionally, CERT/CC sponsored a workshop attended by thirty computer experts to address the threat of distributed attack tools. During the resulting Distributed-Systems Intruder Tools (DSIT) Workshop [46], participants discussed and proposed a number of approaches to preventing, detecting, and responding to DDoS attacks. David Dittrich, a senior security engineer at the University of Washington, has produced an in depth analysis of many DDoS tools. His website [47] provides links to many research papers and articles regarding DDoS. The System Administration, Networking, and Security (SANS) Institute posts practical advice concerning DDoS attacks [48, 49] as well as step-by-step procedures [50] to help stop these attacks.

2.3.1. DDoS Architecture

There have been a number of articles and research papers that provide in-depth analysis of various DDoS tools. However, the terminology used to describe the DDoS architecture is often inconsistent. For the purpose of this paper the DDoS architecture will be defined with the following nomenclature [51]:

- Attacker: The host that initiates all attacks.
- Handler: An intermediary between the attacker and the agent. The handler is responsible for command and control of the attack.
- Agent: The systems that launch the attacks against the target(s).
- Daemon: The DDoS process on the agent.
- Target: The host or network that is the focal point of the attack by the constellation.
- Cell: The agent cluster that is associated to a specific handler.
- Constellation [52]: The typical architecture of a DDoS tool (refer to Figure 10: DDoS Attack – Phase 2).

DDoS implementations employ a client-server architecture and can best be modeled as a constellation. Typically a three-tier architecture (refer to Figure 8: DDoS Tiers) is used to describe the DDoS architecture, but attackers are not limited to this implementation. Tier 1 contains the attacker. The attacker communicates to the handlers the attack type and target information. Tier 2 contains the handlers that receive attack instructions from the attacker. Handlers relay the attack information to the agents under their control. Tier 3 contains the agents that receive attack instructions from the handlers and implement the attack.

Figure 8: DDoS Tiers

The more tiers that exist between the attacker and the agents, the harder it will be to determine the attacker's identity. During a flooding attack, the target will receive spoofed network traffic from multiple systems. Every attacking agent represents an incident that the target will have to manage. With hundreds of attacking agents, the victim will have to manage hundreds of incidents simultaneously and will quickly become overwhelmed. Tracing spoofed IP datagrams to their true origin is a time consuming and complex task (refer to section 2.1.3 Tracing Spoofed IP Datagrams Back to the Source). Even if the target manages to trace the network traffic back to an attacking system, they will have not succeeded in locating the attacker. The investigation will have simply revealed one of the compromised systems participating in the attack. Every system that lies between the attacker and the target makes it more difficult for the target to determine the identity of the attacker.

2.3.2. DDoS Attack Scenario

In order to execute a DDoS attack, an attacker must have at their disposal a number of previously penetrated systems with installed DDoS software. This involves locating and penetrating weakly secured systems across the Internet. This activity is often referred to as the mass intrusion phase [53]. The mass intrusion phase usually occurs soon after a new exploitable vulnerability is discovered. For instance, in 1999 several exploitable vulnerabilities were discovered in the UNIX Remote Procedure Call (RPC) service. This vulnerability allowed remote and local users the ability to execute arbitrary code on a vulnerable system (in some cases with root privilege) [54, 55]. Soon after, both the National Information Protection Center (NIPC) and CERT/CC issued warnings of DDoS software installation occurring on systems compromised with these vulnerabilities [56, 57].

Once the mass intrusion phase is completed, the systems can be organized into the DDoS architecture. This involves an attacker installing DDoS software on compromised systems and building the DDoS constellation. The DDoS attack is usually carried out in three distinct phases:

- Phase 1: The attacker makes contact with the handlers to begin the attack.
- Phase 2: The handlers communicate with the agents the attack information such as: attack type, victim address, and length of attack.
- Phase 3: The agents receive the communications from the handlers and begin the attack.

Referring to Figure 9: DDoS Attack – Phase 1, the attacker has selected two of the compromised systems to act as handlers (i.e. Handler –1 and Handler –2). The attacker communicates to the handlers the attack type and target information.

Figure 9: DDoS Attack – Phase 1

The handlers act as an intermediary between the agents that carry out the attack and the attacker. Figure 10: DDoS Attack – Phase 2, illustrates the communications between the handlers and their respective agents. Typically, each handler will have a unique list or group of agents they control called a cell. Cell organization is important because if any handler is identified and examined only the agents they control will be discovered. Although these agents may be located and stopped, the agents in the other cells will remain hidden and can continue attacking. Handler -1's cell is comprised of all the agents labeled Agent -1. Handler -2's cell is comprised of all the agents labeled Agent -2.

Figure 10: DDoS Attack – Phase 2

Once the agents have received the attack information from their handler, they will commence the attack. Figure 11: DDoS Attack – Phase 3 illustrates all the agents sending a flood of network traffic to the victim. During a typical DDoS attack, spoofed traffic from hundreds or thousands of agents enters the network directed at the victim. Most networks are not designed to accommodate the service demand or network traffic generated from this type of attack thus causing either system or network failure.

Figure 11: DDoS Attack – Phase 3

2.3.3. Attacks

Most DDoS tools have a variety of attacks at their disposal. These attacks can be characterized as either remote implementation flaw attacks or remote resource exhaustion attacks. Remote implementation flaw attacks exploit either configuration errors in systems and software applications or coding errors/omissions in specific software applications. An example of this type of attack is the teardrop attack [26]. The teardrop attack exploits a software-coding flaw in some implementations of the TCP/IP network stack application. An attacker could craft an IP datagram with an invalid fragmentation field that would cause data to overlap during datagram reassembly. Vulnerable TCP/IP network stack applications can not process this type of datagram causing the system to crash or reboot. Although remote implementation flaw attacks can be an effective type of DoS attack, not all DDoS tools implement this functionality. The majority of DDoS tools

employ remote resource exhaustion via flooding [38]. The main reason for selecting this type of attack is to mitigate against countermeasure development. Implementation flaw attacks rely on flaw discovery in software code that in turn reveals exploitable vulnerabilities. Just as tools can be rapidly developed to exploit newly discovered vulnerabilities, software patches and fixes can also be developed to fix the coding errors or implement effective countermeasures. Countermeasures against flooding attacks are much more difficult to implement and thus these types of attacks are more robust. An example of a remote implementation flaw attack used by a DDoS tool is:

- Targa: A DoS attack that employs a combination of remote implementation flaw attacks. The Targa tool alternates between the Bonk, Jolt, Land, Nestea, Newtear, SYNdrop, Teardrop, and Winnuke attacks [43].

Remote resource exhaustion attacks consume finite resources until none are left to service legitimate resource requests. Flooding involves sending inordinate amounts of network traffic to a system until servicing the bogus requests overwhelms the system. Then, the network infrastructure becomes overwhelmed or all network bandwidth is consumed. The datagrams that comprise the flooding attack are spoofed so that determining the source of the attack is also more difficult. An example of this type of attack is the UDP flood [58]. The UDP flood attack involves sending a constant stream of UDP datagrams to random ports on a target system. Eventually, the target system becomes overwhelmed, the network infrastructure becomes overwhelmed, or the network bandwidth is consumed by the spurious UDP datagrams.

Examples of remote resource exhaustion attacks employed by DDoS tools are:

- TCP SYN flooding: TCP flooding has been determined to be the most prevalent form of DoS on the Internet [37]. It consists of sending a stream of spoofed TCP SYN datagrams to a listening TCP port on a system. The SYN flood attack exploits a resource limitation in the TCP three-way handshake mechanism used to establish

connections between the systems [25]. A typical TCP connection involves: one system, the client, requesting a connection to a system, the server, and then sending a datagram with the SYN flag set. The server responds with a datagram with the SYN and ACK flags set. Finally, the client responds to the server with a datagram with the ACK flag set and the connection between the two systems is established. This process is called the 3-way handshake. A SYN flood attack occurs when the client sends connection requests to the server without completing the third part of the three-way handshake. According to the TCP/IP protocol, the server maintains half-open connections waiting for a finite period of time before the half-opened connections expire and the resources are released. If enough TCP SYN datagrams are sent, either the server will not be able to respond to legitimate connection requests or all the network bandwidth will be consumed.

- **Smurf attack**: The broadcast address on a network allows datagrams to be broadcast to all systems within a subnet. A smurf attack involves an attacker sending an ICMP echo request datagram to the broadcast address using the spoofed IP address of the intended victim. The victim will receive a flood of ICMP echo reply datagrams from all systems within the broadcast address range [58].
- **ICMP flooding**: This is similar to a smurf attack, except the ICMP echo request datagrams are sent to the victim by multiple systems instead of the broadcast address.
- **UDP flooding**: This attack involves sending UDP datagrams to random ports on a system. When the system receives UDP datagrams and determines there is no application listening on the port, it will generate ICMP unreachable messages that it will send to the source IP address found in the UDP datagrams [59].

2.3.4. Communication Methods

The first generation of DDoS tools relied on unencrypted session-based communications. Normal TCP connections were used by the attackers, handlers, and agents to exchange attack information.

Figure 12: DDoS Communications

This communication method, although the most reliable, posed a greater risk to the attacker. Handler and agent discovery could occur by routine security monitoring of network traffic. Patterns in DDoS communication streams can be discovered through analysis. IDS signatures can then be developed and used to alert network administrators that DDoS communications are occurring on the network.

These factors led to the development of techniques to improve communications and add increased stealth and security to the DDoS tools such as:

- **Encryption**: Encryption is one method to defeat casual security monitoring and network-based IDS. Most IDSs are signature-based and can detect identifiable patterns associated with certain attacks. Encryption ensures that the communication patterns are unreadable by the IDS and thus seen as normal network traffic.
- **Password protection**: Any handler wishing to send commands or communicate with an agent must provide the correct password. To ensure they are not taken over by other handlers or detected by DDoS detection tools, agents are password protected.
- **Oblivious control**: Oblivious control [60] occurs when a DDoS agent can receive command and control information from a handler, but does not have to respond. One method to achieve oblivious control is through the use of ICMP datagrams. ICMP is a connectionless protocol. Among the services offered by ICMP is the Ping command that utilizes the ICMP echo request and ICMP echo reply datagram. ICMP echo reply datagrams can contain hidden or covert data [61]. Since ICMP is connectionless, the source IP address can be spoofed with no requirement for a reply and thus conceals the identity of the attacker.
- **Decoy datagrams**: Decoy datagrams are intermixed with actual DDoS command and control datagrams and then sent to random IP addresses. If handler communications are intercepted, these decoy datagrams are useful in hiding the IP addresses of actual DDoS agents.
- **IRC**: IRC, used as a DDoS communications bridge, negates the need for agents to listen on a system port. Unless IRC traffic is restricted on the network or it is being monitored for suspicious activity, DDoS agents that utilize IRC will be difficult to detect.

2.3.5. DDoS Tools Survey

A few of the more popular DDoS tools are listed in Table 3: DDoS Tools Survey. All of these DDoS tools are available for download from [43].

DDoS Tool	Attacks	Communications	Miscellaneous Functionality
Mstream	<ul style="list-style-type: none"> TCP flooding 	<ul style="list-style-type: none"> Attacker to handler – unencrypted TCP Handler to agent – unencrypted UDP Agent to handler – unencrypted UDP 	<ul style="list-style-type: none"> Agents are informed of all handler access attempts (successful or not)
Trinoo	<ul style="list-style-type: none"> UDP flooding 	<ul style="list-style-type: none"> Attacker to handler – unencrypted TCP Handler to agent – unencrypted UDP Agent to handler – unencrypted UDP 	<ul style="list-style-type: none"> Source IP addresses of UDP datagrams are not spoofed
TFN	<ul style="list-style-type: none"> ICMP flooding TCP flooding UDP flooding Smurf 	<ul style="list-style-type: none"> Attacker to handler – requires third party program Handler to agent – unencrypted ICMP Agent to handler – none 	<ul style="list-style-type: none"> Makes use of spoofed IP addresses in attack datagrams Bind a root shell to a specified port
TFN2K	<ul style="list-style-type: none"> ICMP flooding TCP flooding UDP flooding Smurf Mix flood Targa attack 	<ul style="list-style-type: none"> Handler to agents – can be a mixture of encrypted ICMP, TCP and UDP Agents to handler – none 	<ul style="list-style-type: none"> Oblivious control
Plague	<ul style="list-style-type: none"> TCP flooding 	<ul style="list-style-type: none"> Handler to agents – requires third party program, usually unencrypted TCP Agent to Handler – unencrypted TCP 	<ul style="list-style-type: none"> 2 tier architecture
Stacheldraht and variants	<ul style="list-style-type: none"> ICMP flooding TCP flooding UDP flooding Smurf 	<ul style="list-style-type: none"> Attacker to handler – encrypted TCP Handler to agent – encrypted UDP Agent to handler – none 	<ul style="list-style-type: none"> Upgrade agents on demand Oblivious control
Shaft	<ul style="list-style-type: none"> UDP flooding TCP flooding ICMP flooding Mix flood 	<ul style="list-style-type: none"> Attacker to handler – unencrypted TCP Handler to agent – unencrypted TCP or encrypted ICMP Agent to handler – unencrypted UDP 	<ul style="list-style-type: none"> Permits the attacker to compile statistics on agents
Trinity	<ul style="list-style-type: none"> TCP flooding UDP flooding 	<ul style="list-style-type: none"> Uses IRC as it communication method 	

Table 3: DDoS Tools Survey

2.4. Anatomy of a Network Attack

The following quote was taken from Lance Spitzner's *Know your enemy* series:

"My commander used to tell me that to secure yourself against the enemy, you first have to know who your enemy is. This military doctrine readily applies to the world of network security. Just like the military, you have resources that you are trying to protect. To help protect these resources you, need to know who your threat is and how they are going to attack [5]."

Designing, deploying, and maintaining a proper security posture for a network is an expensive, time consuming, and complicated undertaking. The challenge of securing a network is made more difficult due to an ever-changing operational environment. Modern networks exist in a constant state of evolution. Changes to a network topology occur for a variety of reasons. The emergence of new threats and vulnerabilities may require adjustments or augmentation to network security devices. The normal system life cycle requires the periodic addition, replacement, or reconfiguration of systems. As an organization matures the user requirements, business model, information processed and stored, and asset valuations change. As a result, periodic assessments of the security posture have to be performed to determine their effectiveness and adequacy. Security has to be regarded as an ongoing process. Therefore, the resources required to adequately protect a network can place a significant burden on an organization.

An organization should perform a cost benefit analysis to determine the security level it requires. In order to determine the proper amount of security required by a network, a thorough understanding of the operational environment is necessary. This includes, but is not limited to, understanding the attack methodologies that may be employed against network protection mechanisms.

2.4.1. What is a Network Attack?

There are a number of definitions of what constitutes a computer attack or incident.

“An attack is a single unauthorized access attempt, or unauthorized use attempt, regardless of success [62].”

This definition is somewhat limiting, as it does not address DoS attacks. DoS attacks are designed to deny the use of a system or service to legitimate users. A broader definition of a network attack is:

“A network attack or security incident is defined as a threat, intrusion, denial-of-service, or other attack on network infrastructure, computer system(s), or user account(s). Computer security incidents can vary from annoying email directed at an individual to intrusion attacks on sensitive data and computer systems. Some security incidents are inherently computer-based; in others, the electronic medium is coincidental to the crime or policy violation [63].”

Network attacks are intentional attempts to gain unauthorized access to, or prevent legitimate use of, electronically stored information or computational resources. Network attacks can be broadly categorized as either vulnerability exploitation or DoS:

- **Vulnerability exploitation**: An attack that successfully exploits a vulnerability in a system leading to unauthorized access. Vulnerabilities typically occur in systems due to configuration errors, implementation errors, or software flaws [64]. Vulnerability exploitation leads to:
 - the extraction or modification of information on the compromised machine or associated network segment, and/or
 - the subversion of computational resources for malicious activities (e.g. scanning, sniffing, DDoS attacks).
- **DoS**: An attack that successfully exploits a vulnerability in a system leading to the impairment or denial of services offered by the system to legitimate users. DoS attacks can also occur by flooding.

2.4.2. Stages of a Network Attack

Regardless of whether a network attack is used for vulnerability exploitation or DoS, a number of stages in the network attack methodology are shared. A typical network attack occurs in five distinct stages described in Table 4: Stages of a Network Attack.

Stage	Description
1. Reconnaissance	Information gathering about the target(s) in order to determine: the vulnerability to be exploited, the attack to be employed, and the likelihood of successful compromise.
2. Target Acquisition	The selection of the host/network for active exploitation or a denial of service attack.
3. Attack	A vulnerability exploitation or denial of service attack.
4. Compromise and Implantation	A successful attack results in unauthorized access called a compromise. Once compromise occurs, the attacker will typically install software on the target to enable covert control and communications.
5. Data Acquisition and Exfiltration	After control and communications with the target have been gained, the attacker can export information or use the target as an attack launch point.

Table 4: Stages of a Network Attack

The overall goal of a DoS attack is to deny use of the system or services the system provides to legitimate users. In a DoS attack an attacker has no interest in accessing resources on or maintaining control of the target. A DoS attack does not involve a compromise and subsequent subversion of the system to gain unauthorized access. DoS attacks employ only stages one to three. Vulnerability exploitation attacks employ all five stages of a network attack. The difference is depicted in Figure 13: Stages of a Network Attack.

Figure 13: Stages of a Network Attack

2.4.2.1. Reconnaissance

The goal of reconnaissance is to identify attributes about a specific target or a range of targets in order to enable a successful attack. The target is the system (network and/or host) that the reconnaissance is directed against. Target-specific reconnaissance occurs when the information gathering activities are restricted to a predetermined target. The targeted system contains: information, privileged access, computational resources, or

some other attribute of interest to the attacker. This type of reconnaissance is precise, deliberate, and focused.

Wide-range reconnaissance is used to scan large blocks of Internet address space to locate systems running a particular service or containing a specific vulnerability. The criteria for this type of attack are not based on a specific asset of interest to the attacker, but rather a specific vulnerability that can be exploited. Typically, there is little human interaction in this type of reconnaissance. Automated tools are much more efficient at detecting large numbers of vulnerable systems. This "shot-gun" approach to reconnaissance can be characterized as opportunistic scanning. Multitudes of systems are quickly scanned to locate systems with a specific vulnerability. An example of this phenomenon is the Ramen worm discovered in January 18, 2001 [65]. The Ramen worm is an automated exploitation tool that consisted of a collection of generally available hacking tools. Using simple methods to determine vulnerable systems, it scanned and exploited systems on the Internet at an extremely fast rate.

Reconnaissance can either be technical or non-technical in nature. It involves collecting as much information about the target as possible. Non-technical reconnaissance can consist of a number of traditional information gathering techniques such as:

- Open-source research: This includes, but is not limited to: dumpster-diving, newspapers articles, websites, and annual reports. Researching publicly available information such as: assets, organizational security posture, network services, network technologies, and authorized user population may offer intelligence that can be used in an attack.
- Social engineering: This is a non-technical intrusion of the organizations security posture performed exclusively by human interaction. It is the means of gaining information from organizational personnel by tricking them into breaking normal security procedures.

- Covert physical access: Surreptitiously gaining physical access to an organization's buildings or information systems.

Technical reconnaissance relies on the use of scanning tools. Scans are done so that an adversary can determine the network topology, organizational assets, and security countermeasures. Depending on the type of scanning tools used and the security posture of the network, the following information about the network may be determined:

- Number of systems: Accessible systems present on the network.
- Type of systems: Operating systems, versions of accessible network systems (e.g. UNIX, Linux, Windows, etc.), including function (e.g. workstation, server, network security device, etc.).
- IP addresses: The specific IP addresses of the systems on the network.
- Security patches applied: The latest software patches installed on the system.
- Offered services: Publicly accessible services (e.g. Telnet, HTTP, FTP, email, etc.).
- Installed software: The applications installed on the system.

Network scanning can be undertaken in a number of ways, but it usually occurs in two phases:

- Phase 1 - Network discovery: This phase involves determining the IP addresses and relative function of the systems on a network. The goal is to construct, as accurately as possible, a depiction of the target network. Often referred to as determining "live hosts" or locating systems that respond to network queries. The Ping command that implements the ICMP protocol is often used to determine live hosts. Once systems are discovered, they can be probed to reveal attributes about the system. System probes can be accomplished using a variety of techniques:
 - Whois: The Whois command reveals the owner of the domain name for a range of IP addresses.

- DNS zone transfers: These requests prompt authoritative domain name servers to return information contained in their records corresponding to the IP addresses.
- Traceroute: This program determines the number, IP addresses, and relative locations of routers between two systems.
- Sam Spade [66]: A very useful web-based tool for general-purpose research on the Internet. It provides many of the techniques listed above.
- Phase 2 - Host Discovery: Once the location and general function of the system has been determined, a more through investigation can be undertaken. Some of the techniques used to accomplish this investigation are:
 - Port scans: There are 65535 TCP and 65535 UDP ports on every system. RFC 1700 specifies the port location of well-known services or applications [67]. Every open port means that an application is listening and waiting for connections. Each open port reveals another possible entry point into the system for the attacker. A few well-known services and their corresponding ports are:
 - HTTP: port 80
 - SMTP: port 25
 - Telnet: port 23
 - FTP: port 20 and port 21
 - Banner grabbing: Some services display a banner when a connection attempt is made. For instance some versions of the Telnet application can return a banner that displays the operating system type and version to anyone that tries to initiate a Telnet session.
 - Operating system fingerprinting: RFCs define how systems should react to standard connection attempts. However, they don't specify how systems should react to non-standardized connection attempts. If illegal or unexpected TCP/IP flag combinations are sent to a system, it will react based on its particular software implementation of the TCP/IP stack. Each operating system has a corresponding

unique software implementation of the TCP/IP stack that can be remotely determined when it responds to connection attempts.

- **Firewalk [68]:** A tool used to determine the filtering rules of datagram forwarding network devices. It analyzes responses from network devices to determine the access control lists so that a mapping of the network can be made.
- **Ping:** A command that can often be used to accurately determine the operating system type and version. A complete explanation of how to use this command for this purpose can be found in the paper *ICMP Usage in Scanning* [69].

Methodical scanning of the Internet is a frequent occurrence. Examination of most computer logs, from systems that connect directly with the Internet, will reveal scans or probes from multiple sources that are trying to determine network boundaries.

Attackers use a number of scanning techniques that attackers use in order to avoid detection:

- **Decoy scanning:** An attacker can use a previously compromised system as a platform to perform scanning. If scanning activities are detected, they will be attributed to the compromised system not the attacker's local system.
- **Distributed scanning:** Instead of scanning from a single system, scanning can be done concurrently by multiple systems. The scanning results are gathered and tabulated into a single report by the attacker. To a system administrator it appears the scans are coming from multiple sources thus looking less suspicious.
- **Low and slow scans:** An entire network can be scanned in a matter of seconds. Most IDS easily detect this type of scanning. Instead of scanning an entire network in a matter of seconds, scans can be slowed to take minutes, hours, or days. The advantage to the attacker is the scan becomes intermixed with other network traffic thus looking less suspicious.

What is the capability of today's network scanning tools? To determine the amount of information that can be gathered by the use of such tools, we will examine one in greater detail.

"Nmap ("Network Mapper") is an open source utility for network exploration or security auditing [70]."

Nmap is a software tool that allows a user to scan IP address ranges to determine open ports, live systems, and perform OS fingerprinting. It is a robust scanning tool that can avoid detection and circumvent firewalls and routers. Figure 14: Screen Capture of Nmap reveals its functionality.

```
root@ATTACKER /root]# nmap
nmap V. 2.53 Usage: nmap [Scan Type(s)] [Options] <host or net list>
Some Common Scan Types ('*' options require root privileges)
  -sT TCP connect() port scan (default)
  * -sS TCP SYN stealth port scan (best all-around TCP scan)
  * -sU UDP port scan
  -sP ping scan (Find any reachable machines)
  * -sF, -sX, -sN Stealth FIN, Xmas, or Null scan (experts only)
  -sR/-I RPC/Identd scan (use with other scan types)
Some Common Options (none are required, most can be combined):
  * -O Use TCP/IP fingerprinting to guess remote operating system
  -p <range> ports to scan. Example range: '1-1024,1080,6666,31337'
  -F Only scans ports listed in nmap-services
  -v Verbose. Its use is recommended. Use twice for greater effect.
  -P0 Don't ping hosts (needed to scan www.microsoft.com and others)
  * -Ddecoy_host1,decoy2[...] Hide scan using many decoys
  -T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane> General timing policy
  -n/-R Never do DNS resolution/Always resolve [default: sometimes resolve]
  -oN/-oM <logfile> Output normal/machine parsable scan logs to <logfile>
  -iL <inputfile> Get targets from file; Use '-' for stdin
  * -S <your_IP>/-e <devicename> Specify source address or network interface
  --interactive Go into interactive mode (then press h for help)
Example: nmap -v -sS -O www.my.com 192.168.0.0/16 '192.88-90.*.*'
SEE THE MAN PAGE FOR MANY MORE OPTIONS, DESCRIPTIONS, AND EXAMPLES
```

Figure 14: Screen Capture of Nmap

To test the functionality of Nmap, it was executed on a small test network as depicted in Figure 15: Network Schematic. The test network was comprised of a laptop that

contained the Nmap software (i.e. 192.168.0.55) and the default installation of six Linux and two Windows systems.

Figure 15: Network Schematic

Using the command `nmap -O 192.168.0.*`, TCP/IP fingerprinting is performed on the entire network to remotely guess the operating systems. The results are listed in Table 5: Nmap Scan Results.

Hosts	Nmap results																					
192.168.0.1 192.168.0.2 192.168.0.3 192.168.0.4 192.168.0.5 192.168.0.8	Interesting ports on (192.168.0.1): (The 1517 ports scanned but not shown below are in state: closed) <table><tr><th>Port</th><th>State</th><th>Service</th></tr><tr><td>25/tcp</td><td>open</td><td>smtp</td></tr><tr><td>111/tcp</td><td>open</td><td>sunrpc</td></tr><tr><td>113/tcp</td><td>open</td><td>auth</td></tr><tr><td>515/tcp</td><td>open</td><td>printer</td></tr><tr><td>587/tcp</td><td>open</td><td>submission</td></tr><tr><td>1024/tcp</td><td>open</td><td>kdm</td></tr></table> TCP Sequence Prediction: Class=random positive increments Difficulty=5254530 (Good luck!)	Port	State	Service	25/tcp	open	smtp	111/tcp	open	sunrpc	113/tcp	open	auth	515/tcp	open	printer	587/tcp	open	submission	1024/tcp	open	kdm
Port	State	Service																				
25/tcp	open	smtp																				
111/tcp	open	sunrpc																				
113/tcp	open	auth																				
515/tcp	open	printer																				
587/tcp	open	submission																				
1024/tcp	open	kdm																				
192.168.0.6 192.168.0.7	Remote operating system guess: Linux 2.1.122 - 2.2.14 Interesting ports on (192.168.0.7): (The 1520 ports scanned but not shown below are in state: closed) <table><tr><th>Port</th><th>State</th><th>Service</th></tr><tr><td>135/tcp</td><td>open</td><td>loc-srv</td></tr><tr><td>139/tcp</td><td>open</td><td>netbios-ssn</td></tr></table> TCP Sequence Prediction: Class=trivial time dependency Difficulty=7 (Trivial joke) Remote operating system guess: Windows NT4 / Win95 / Win98	Port	State	Service	135/tcp	open	loc-srv	139/tcp	open	netbios-ssn												
Port	State	Service																				
135/tcp	open	loc-srv																				
139/tcp	open	netbios-ssn																				
Nmap run completed -- 256 IP addresses (9 hosts up) scanned in 13 seconds																						

Table 5: Nmap Scan Results

256 IP addresses or a full class C network was scanned in 13 seconds returning the information listed in the table above. Nmap was able to accurately determine all of the operating systems and the open ports on every system.

2.4.2.2. Target Acquisition

Target acquisition involves the selection of the intended host or network for active exploitation or a DoS attack. A number of possible scenarios for target selection are possible:

- Defacement target: Usually high profile websites that are a main corporate or government Internet point of presence.
- DoS target: Hosts or networks that offer services that the attacker wishes to disrupt or deny for a period of time.

- DoS amplifier: Networks that are susceptible to being Smurf amplifiers. These networks will respond to ping requests on the broadcast address of 255.255.255.255 thus causing a Smurf attack against a victim.
- Hop point: An attacker may use more than one compromised system to launch an attack.
- Random target: As a result of network reconnaissance against a potential target, a new target is discovered that was not originally considered. This is an example of opportunistic scanning.
- Information theft or modification: Hosts that store information that the attacker wishes to acquire or modify.
- Infrastructure control targets: Systems that control critical parts of the infrastructure.

2.4.2.3. Attack

Reconnaissance provides the necessary information to enable a successful attack. The attacker uses the information obtained from reconnaissance activities to determine the best way to attack the target. This may involve choosing between launching the attack from their local system or attacking remotely via a previously compromised system. An attacker will consider many factors when determining the attack launch point. The two primary factors are the security posture of the network and the type of attack to be launched.

If the attacker determines that the security posture of the network is low, then they may chose to take fewer precautions against detection. It may not be necessary to use previously compromised systems to attack the new target, thus the attack can be launched from the attacker's local system. Attacks using remote systems are attractive because every system between the attacker and the target adds an additional layer of anonymity.

This will introduce additional complexity to the attack. Additionally, it presupposes the attacker has previously compromised systems at their disposal.

If the security posture of the target is high, then extra care must be given when launching the attack. The network may have an IDS capability and the network administrators may regularly check their server, router, firewall, and security logs. The attacker may choose to launch the attack using multiple compromised systems which adds additional layers of abstraction. Every hop away from the attacker adds another layer between the attacker and the target and makes it harder to trace the attack to its true origin.

The type of attack also influences the selection of the attack launch point. If the goal of the attack is to compromise a system to gain access, then communication between the attacker and the target must occur. The attacker needs to be able to send and receive command and control communications between their system and the target system in order to maintain access. If denial of service is the goal of the attack, then no command and control communication between the target and the attacker is required.

Attacks vary in scope and sophistication. There are a number of website repositories of network attack tools that can be used by attackers with limited skills. Successful targeted attacks normally use exploits chosen as a result of the information gathered from reconnaissance. For instance, exploits developed specifically for UNIX operating systems will not be successful against Windows systems. Sophisticated attackers may have their own private exploits that they have developed that are not public knowledge. Typically, these exploits are developed after extensive vulnerability analysis has occurred on the operating system or targeted applications. A successful attack occurs when an attacker gains unauthorized access to the target or, as in the case of a DoS attack, denies its use to legitimate users.

The objective during a compromise is to obtain privileged access to a system. That is to obtain root or administrator privileges on a system. With this level of privilege, the attacker can view and modify audit logs, start-up scripts, and system files. Some applications attackers need to run and install in privileged mode. If an attack results in non-privileged access to the system, the attacker will attempt to use this foothold to escalate their privilege on the system.

Attacks can be carried out in many different ways. They can involve a substantial amount of human intervention to execute or be totally autonomous. Instead of manually installing and compiling applications, they can be done automatically by using composite tools and attack scripts. Autonomous attack propagation methods include [38]:

- Central source propagation: After the compromise of a system occurs, the attacking program executes an instruction to transfer a copy of the attack toolkit from a central location. The attack program can be located on a server and transferred via the HTTP, FTP, and RPC protocols.
- Back-chaining propagation: After the compromise of a system occurs, the system executes an instruction to transfer a copy of the attack toolkit from the attacking host. Back-chaining propagation is more robust than central source propagation because it avoids single points of failure.
- Autonomous propagation: After the compromise of a system occurs, the victim directly executes the attack instructions. These instructions include launching the attack to other systems without any file retrieval from an external source.

2.4.2.4. Compromise and Implantation

A successful attack leads to a compromise of the target. Once the attacker has gained unauthorized access to a system, their next goal is to maintain that access. Implantation usually involves installation of a number of tools such as: back channels, rootkits, log

cleaners, and covert communication channels. Log files must be edited, programs to maintain and hide access must be installed, and covert communication channels must be established.

Maintaining access to the system must involve using a network communications utility. Netcat [71] is a UNIX application that enables network communications between systems using the TCP or UDP protocol. Data can be written back and forth between systems on a network. For example, consider two systems A and B. Netcat is run as a server application on system A and listens on a specified port. Netcat is run as a client application on system B. System B initiates the connection to the specified port on system A and the subsequent data transfer occurs. This program is a useful way to communicate and transfer files between systems. Netcat can even provide a login shell when connection to the listening port is made.

Rootkits are extremely useful to the attacker because they afford hidden access to the compromised system.

“A hacker security tool that captures passwords and message traffic to and from a computer. A collection of tools that allows a hacker to provide a backdoor into a system, collect information on other systems on the network, mask the fact that the system is compromised, and much more. Rootkit is a classic example of Trojan Horse software. Rootkit is available for a wide range of operating systems [72].”

Rootkits are really a just collection of programs useful to an attacker. The term rootkit is used because these programs are installed after the attacker has broken root and has taken over the system with administrator privilege. Many include sniffers to capture passwords as they are passed through the network. Sniffers are programs that can intercept, monitor, and analyze traffic on a network segment. Rootkits replace key files on the victim's system with their own versions. This hides the attacker's unauthorized access to the system from the system administrator. A rootkit DIR command will show all files on the

system with the exception of files owned by the attacker. It replaces files surreptitiously with other files that appear to behave as the original, but have hidden functionality, this is called trojaning. Although rootkits vary in functionality and complexity, some common files that are replaced by rootkits with trojaned files are: ls, login, find, du, crontab, ps, tcpd, netstat, and killall. An example of a rootkit is T0rnkit [73]. T0rnkit is a rootkit for Linux and Solaris operating systems. It is very easy to use and provides exceptional functionality to an attacker such as: log cleaning, sniffer, backdoor program, trojaned files, and communications utilities.

Log cleaning tools make it more difficult to detect that unauthorized access has occurred. These tools remove all evidence of connections between the attacker and the target.

2.4.2.5. Data Acquisition and Exfiltration

Rootkits provide a method for stealth control locally on a host. The attacker also needs to hide communications from the victim to the attacker; this is achieved through the use of covert channels. Secret communications with the compromised systems are required in order to avoid detection. Additionally, data of interest to the attacker in the form of files, user accounts, or information gathered from network sniffer programs has to be sent to the attacker.

A covert channel may be defined as any communication channel that can be exploited by a process to transfer information in a manner that violates a system's security policy [74]. The goal of a covert channel is to move data or communicate secretly. Tunneling technology has typically been used to circumvent restrictive security practices on a network. If a network does not allow Telnet through the organization's firewall, there are programs available that allow you to use that service through other protocols.

A few examples of tunneling programs are:

- Reverse www shell: This tool allows an attacker to access the command prompt of a system remotely after the installation of the reverse www shell server [74]. The attacker simply sends their requests embedded in HTTP to the reverse www shell server and then the system responds by sending out the command line prompt embedded in standard HTTP traffic. To casual observers on the network this appears to be standard HTTP connections.
- Loki: A tool that uses ICMP echo network traffic [61]. Loki exploits the fact that ICMP traffic is ubiquitous and typically allowed to flow freely between networks and that there exists an optional data payload portion within ICMP echo datagrams. By placing data in the optional data portion of the ICMP traffic, a covert channel exists.

Chapter 3 Examination of DDoS Tools

Detecting DDoS activity on a network can be accomplished by three methods. The first method involves local examination, of the systems that comprise the network, for DDoS agents and handlers. There are tools available that inspect the contents of a system's main memory and file system to detect installed DDoS software. This method is time intensive and does not scale well in large networks. Additionally, it does not address detecting DDoS attack activity. The second method involves mimicking DDoS communications to locate or shutdown DDoS agents. There are a number of DDoS detection programs that can simulate DDoS communications in order to elicit a response from or send a command to a DDoS agent or handler. A listing of DDoS detection tools can be found in Table 10: DDoS Detection Tools. The third method involves monitoring network traffic for DDoS activity. The detection of DDoS communications or attack traffic is a good indication of network DDoS activity.

The purpose of this section is to apply the third method, monitoring network traffic for DDoS activity. Two separate DDoS tools will be examined, TFN2K and Plague. If unique characteristics in observable network behavior exist, then it may be possible to develop signatures that can be used to detect the presence of this DDoS activity on a network. Additionally, this investigation will lead to a better understanding of the capabilities of these two DDoS tools.

3.1. Test Environment

The test network consisted of eight PCs, running the Linux Red Hat 6.2 operating system, connected together by a hub. Six of the computers had DDoS agent software installed, one computer had the handler application installed, and one computer had network monitoring software installed. A full description of the test environment is found in Table 6: DDoS Network Testbed Configuration.

IP address	Operating System	Function	Installed software
192.168.0.1	Linux Red Hat 6.2	Target	No additional software
192.168.0.2	Linux Red Hat 6.2	DDoS Agent	DDoS agent
192.168.0.3	Linux Red Hat 6.2	DDoS Agent	DDoS agent
192.168.0.4	Linux Red Hat 6.2	DDoS Agent	DDoS agent
192.168.0.5	Linux Red Hat 6.2	DDoS Agent	DDoS agent
192.168.0.6	Linux Red Hat 6.2	DDoS Agent	DDoS agent
192.168.0.7	Linux Red Hat 6.2	DDoS Handler	DDoS handler (Ncat was also installed for phase 2 testing)
192.168.0.8	Linux Red Hat 6.2	Network Monitor	Libpcap version 0.6.1, Tcpdump version 3.6.1, Ethereal network analyzer version 0.8.15, Snort version, 1.6.3 – patch2

Table 6: DDoS Network Testbed Configuration

3.2. Testing Methodology

Two DDoS attack tools were chosen for testing: TFN2K and Plague. Testing was undertaken in two phases. Phase one consisted of installing TFN2K on the test network and then individually launching all available attacks against the target computer. The network traffic was monitored using Tcpdump [75] and the resulting communications and attack traffic was captured. Once testing on TFN2K was completed, the network was brought back to its original configuration by reinstalling the operating systems on the computers containing the DDoS software. Phase two testing consisted of repeating the same test procedures for the second DDoS attack tool, Plague. The observed network traffic from both phases was then examined to determine if observable patterns in DDoS network activity could be used for DDoS detection. Testing was restricted to observation of DDoS network activity only.

3.3. TFN2K

3.3.1. Overview

TFN2K or Tribe Flood Network 2000 is an updated version of a previous DDoS tool called Tribe Flood Network (TFN) created by an individual using the pseudonym of Mixer. It is a two-tiered command line interface DDoS tool. An attacker uses the command line provided by the handler to communicate with the agents. The agents receive the commands and then execute the attacks.

3.3.2. TFN2K Installation

TFN2K installation involved downloading the program [76], untaring the file, and compiling the agent and handler programs. The agent program is named `td` and the handler program is named `tfn`. TFN2K prompts the user for a password during the compilation of the program. This password is used as the key value for the encryption algorithm [77]. The `td` program was installed on the DDoS agents systems and once executed it forked to the background and listened for commands from the handler. The `tfn` program was then installed on the handler system. The test network diagram is shown in Figure 16: TFN2K Test Network.

Figure 16: TFN2K Test Network

3.3.3. Communication Characteristics

Communications between the handler and agents are password protected. Every time the handler wishes to communicate with the agents, a password must be supplied before the commands can be sent. This authentication mechanism protects the agents from responding to commands from a non-authorized handler.

Handler to agent communications is encrypted with CAST-256 algorithm [78] and can be intermixed with decoy datagrams. The encrypted commands are then Base-64 encoded so that the data portion of the IP datagram will be ASCII printable characters. The discovery of this communications characteristic, as well as the general behavior of this tool, was made in the analysis of TFN2K performed by the AXENT Security Team [77]. The commands can be sent via UDP, ICMP, and TCP, or by default a mixture of all three protocols. Agents do not communicate to the handlers and therefore do not acknowledge receipt of the commands. This one-way communication is used to lessen the chance of a handler discovery. To ensure the agents receive the commands, the handler sends the

command twenty times to each agent relying on the probability that at least one will reach the agent. The source IP address of the command and control datagrams is spoofed by default to hide the identity of the handler. Additionally, if required, the source IP address can be spoofed within a specified range of IP addresses to circumvent egress filtering on routers.

The handler can determine the location of the agents in one of two ways. Agent location can be specified in a file or at the command line as part of the attack parameters.

Additionally, TFN2K allows the handler to execute commands on a specified agent. The handler can bind a root shell to a specified port on an agent and then execute commands. This will allow a handler to execute commands at root level privilege on an agent system. The root shell binding functionality was not tested.

3.3.4. Attacks

TFN2K employs a number of different attacks that can be launched against a victim. A typical attack scenario involves the handler sending the attack parameters and the target IP address to the agents. The agents receive this attack information and begin attacking the target with the specified attack. The handler can stop the attack at any time by sending the stop command to the agents. A screen shot of the TFN2K functionality is included in Appendix A TFN2K. The attacks that TFN2K can execute are included in Table 7: TFN2K Attacks.

Attack	Description
SYN flood	Sends a flood of TCP SYN datagrams to a target.
UDP flood	Sends a flood of UDP datagrams to a target.
ICMP flood	Sends a flood of ICMP datagrams to a target.
Smurf flood	Sends a Smurf attack to a target.
Mix flood	Sends a mixture of SYN, UDP, and ICMP flood in a single attack
Targa	Sends a Targa attack to a target.

Table 7: TFN2K Attacks

3.3.5. Attack Options

Attacks can be launched with a variety of options that are specified, as part of the attack parameters, at the command line. A listing of the attack options is included in Table 8: TFN2K Attack Options.

Attack options	Description
[-P protocol]	Specifies the protocol used for handler to agent communication. It can be ICMP, UDP or TCP. The default is a random usage of all three protocols.
[-D n]	Sends out n bogus commands for each real command to evade detection.
[-S hos/ip]	Allows the handler to specify a source IP address used to send the attack commands. It is random by default, but spoofing a source IP address within a limited range may be required if the network routers are performing egress filtering.
[-f hostlist]	The file that contains a list of agents.
[-h hostname]	Used to contact only a single agent.
[-i target string]	Any attack options separated by '@'.
[-p port]	A TCP destination port can be specified for SYN floods.

Table 8: TFN2K Attack Options

3.4. Plague

3.4.1. Overview

Plague is a DDoS tool created by two individuals that use the pseudonyms Blazinweed and Datawar. It is a two-tiered DDoS tool that contains a handler that issues the attack commands and agents that receive the commands and then execute the attack. Plague is still under development and lacks some basic communications functionality. An attacker cannot launch an attack directly from the command line of the handler as is done with TFN2K. The DDoS tool requires the installation of an additional network communications program to enable communication with the handler. This installation of communications software can occur on the handler itself or on another system. If the communications software is installed on a separate system, that separate system can be considered the attacker and the new Plague DDoS architecture would extend to three-

tiers. If the communications software is installed on the handler, it simply sends commands to itself on its own listening port and it remains a two-tiered architecture. For the purposes of our experiment, the latter configuration was chosen so that the network traffic analysis with TFN2K would be more compatible.

3.4.2. Plague Installation

Plague installation involved downloading the program [76], untaring the file, and then compiling the agent and handler programs. The agent program is named ghost and the handler program is named server. The communications program used to enable handler communication was Netcat [71]. The Netcat program was downloaded, compiled, and installed on the handler. The test network diagram is shown in Figure 17: Plague Test Network.

Figure 17: Plague Test Network

3.4.3. Communication Characteristics

Communication between Plague handlers and agents is password protected, but not encrypted. Every time the handler wishes to communicate with the agents, the password must be supplied before the commands can be sent. This authentication mechanism protects the agents from responding to commands from a non-authorized handler. The agent program defines the listening port and password in its source files. The default values are 6969 and `ld.so.l` respectively. Agent programs, once executed on a system, fork into the background and listen on the specified port for commands from the handler.

A file called the `server.list` is required by the handler to identify the location of the agents. The `server.list` file contains the IP address, listening port, and password for every agent. When the handler program is executed it reads the `server.list` file and forks into the background listening on a specified port for commands. The Netcat program was used to connect to handler via the listening port. When connection to the handler is initiated, the attacker is prompted for a login string. This login string is essentially a password and the default value (`ld.so.l`) is specified in the handler source file. Once connection to the handler is made, the attack can be executed. Additionally, Plague allows the handler to execute commands on a specified agent. The handler can bind a root shell to a specified port on an agent and then execute commands. This will allow a handler to execute commands at root level privilege on an agent system. The root shell binding functionality was not tested.

3.4.4. Attacks

Plague employs two different attacks that can be launched against a target. A typical attack scenario involves the attacker connecting to a handler. The handler then sends the attack parameters and the target's IP address to the agents. The agents receive this attack information and begin attacking the target with the specified attack. The handler can stop

the attack at any time by sending the quit command to the agents. A screen shot of the Plague functionality is included in Appendix B Plague. The attacks that Plague can execute are included in Table 9: Plague Attacks.

Attack	Description
Stream	Sends a flood of TCP datagrams to a target.
SYN	Sends a flood of TCP SYN datagrams to a target.

Table 9: Plague Attacks

3.4.5. Attack Options

Plague does not have an advanced attack capability. Attacks cannot be launched with a variety of options included as part of the attack parameters.

3.5. Intrusion Detection Signatures

The focus of the DDoS tool examination is to discover unique characteristics in observable network behavior. These characteristics form patterns that become the signatures in network traffic used by an IDS to reveal potential DDoS activity. A network IDS signature can be defined as:

“A signature defines or describes a traffic pattern of interest [79].”

Network IDS signatures can be created for just about any network activity. This includes: network attacks, connection attempts to specific ports, datagrams with illegal flag combinations, in fact any definable attribute in network traffic. Therefore, network IDS signatures can be complex or simple. Complex signatures are comprised of patterns that contain several characteristics that must be observed before a match occurs. An example of a complex signature is one that looks for a connection attempt to a specific port, on a specific system, and a particular hex value, in the data payload portion of the datagram. If the signature is simple, the pattern may be as uncomplicated as a single characteristic. For

instance, the signature could be any connection attempt to a specific port on any system in the network.

Snort [80] is a software-based IDS that can be used to notify a network or system administrator when a potential intrusion attempt is occurring. It is free, easy to use, and allows the user to create custom intrusion signatures. The signatures created in the following sections are in the Snort format that is described in detail in [81]. An excellent reference that describes how to create network-based IDS signatures is [82].

3.5.1. TFN2K Intrusion Detection Signatures

The network traffic data captures for TFN2K can be found in Appendix A. During testing, all network traffic was captured using Tcpdump and saved to separate files for each test. The resulting data capture files were processed through Snort which gives them the format found in Appendix A. Using the traffic captures, the following Snort intrusion detection signatures were created for TFN2K.

The following three Snort signatures detect TFN2K handler to agent communications:

1. alert icmp any any -> any any (msg:"TFN2K ICMP communication"; itype: 0; icmp_id: 0; content: "AAAAAAAA";)
2. alert udp any any -> any any (msg:"TFN2K UDP communication"; content: "AAAAAAAA";)
3. alert tcp any any -> any any (msg:"TFN2K TCP communication"; content: "AAAAAAAA";)

No signatures were created for any of the TFN2K attacks. The network traffic generated during these attacks did contain some interesting characteristics such as: incorrect header lengths and non-standard ttl values which are discussed in Appendix A TFN2K.

However, these characteristics are insufficient to warrant attributing them to TFN2K

DDoS signatures. These characteristics can be generated by generic non-DDoS versions of the same attacks or improperly constructed datagrams. For instance, the SYN flood attack generated by TFN2K resembles a normal SYN flood attack. Creating an IDS signature for the TFN2K SYN attack and attributing it a possible DDoS attack may cause false or inaccurate alarms if a normal SYN flood attack triggers the alarm.

3.5.2. Plague Intrusion Detection Signatures

The network traffic data captures for Plague can be found in Appendix B. All network traffic during testing was captured using Tcpdump. The resulting data capture files were processed through Snort which gives them the format found in Appendix B. Using the traffic captures, the following Snort intrusion detection signatures were created for Plague. The following six Snort signatures detect Plague DDoS activity on a network:

Pingall signatures

1. alert tcp any any -> any 6969 (msg:"DDOS Plague handler PING agent"; content:"ld.so.1"; flags:AP; fragbits:D;)
2. alert tcp any 6969 -> any any (msg:"DDOS Plague agent PONG handler"; content:"PONG"; flags:AP; fragbits:D;)

SYN flood signatures

1. alert tcp any any -> any 6969 (msg:"DDOS Plague handler send SYN attack command to agent"; content:"ld.so.1"; depth:10; flags:AP; fragbits:D;)
2. alert tcp any 6969 -> any any (msg:"DDOS Plague agent notifying handler it is SYN flooding the target"; content:"flooding target"; flags:AP; fragbits:D;)

Stream signatures

1. alert tcp any any -> any 6969 (msg:"DDOS Plague handler send stream attack command to agent"; content:"ld.so.1"; depth:10; flags:AP; fragbits:D;)

2. alert tcp any any -> any any (msg:"DDOS Plague stream attack"; flags:!FSRPAU21; ack:0; dsize:0;)

3.5.3. Shortcomings of Intrusion Detection Signatures

Observable patterns in network traffic generated by both DDoS tools formed the basis of the intrusion detection signatures created in the previous sections. Although this is an effective method to create intrusion detection signatures, it relies on the fact that these patterns remain static. Most DDoS tools are provided with source code. It would be possible to modify this code and change default values such as: listening ports, process names, login strings, or passwords in the programs. These values comprise the patterns used to create intrusion detection signatures. If any of these values are modified, the signatures will not match and the attack will occur without being detected by the IDS. For instance, the two signatures created for the Plague Pingall command make use of the strings `ld.so.1` and `PONG` and the default port of 6969. If the attacker was to modify the source code to alter these strings or change the default port, then an IDS using the proposed signatures would not detect this activity on a network.

Chapter 4 Current Network Defence Strategies

Implementing a proper security posture for a network requires considerable planning through the use of a proven network security strategy. This chapter describes three different approaches for developing a network security posture. These three approaches are:

- Threat and risk assessment
- Layered network security
- Survivability

This chapter's discussion of the correspondence between layered network security and survivability has been taken directly from my previous publications on the subject. In turn, this previous work has been used to develop the network design strategy in Chapter five.

We also investigate initiatives by government and private industry to deal with DDoS attacks. The chapter concludes with a survey of DDoS detection tools that are currently available.

4.1.1. Threat and Risk Assessment

Choosing the correct security countermeasures to protect a network requires a complete understanding of the operational environment. This includes, but is not limited to defining:

- The type of services the network must provide.
- The network assets that must be protected.
- The authorized user population.
- The perceived threats to the assets.

- The acceptable level of risk that can be tolerated by the organization.

One way to ascertain these factors is to perform a threat risk assessment (TRA). A TRA examines current and/or proposed safeguards to ensure they are adequate to protect the network within an acceptable level of risk [83]. The four steps of a TRA are [84]:

- Step 1: Preparation (*what needs protection?*): defining the scope of the process
- Step 2: Threat assessment (*identification and consequences of a threat*): identifying the threats and associated consequences to the network
- Step 3: Risk Assessment (*safeguard adequacy*): determining the amount of risk an organization will be exposed to when the existing and/or proposed safeguards are deemed insufficient to protect an asset against a threat
- Step 4: Recommendations (*reducing the risk*): suggestions to reduce network vulnerabilities that in turn reduce the level of risk

Step 1 Preparation

The scope of the TRA is determined by the network boundary. The network boundary is the logical separation between the organization's network and all other interconnected networks. Information gathering for this step should be restricted to within the identified network boundary. One primary task that must be completed during the information gathering activity is asset identification and valuation. Assets are organizational goods or resources of value that require protection. Asset valuation must consider not only the replacement value, but also the loss of goodwill. In addition to assets, any information processed or stored by the assets also requires protection. In order to determine the required level of protection for the information, a Statement of Sensitivity (SoS) is performed. A SoS determines the security services required to protect the information. The security services are defined as confidentiality, integrity, and availability [85]. Confidentiality services ensure that the data is only divulged to authorized users.

Integrity services ensure the fidelity of the information. Availability services ensure that the information can be provided when required by the authorized users [83]. Through the requirements gleaned from the SoS, the sensitivity of the information can be determined.

Step 2 Threat assessment

The five main classes of threats are disclosure, modification, interruption, destruction, and removal or loss [84]. Properly identifying and articulating threats to the network provides the necessary information required to select the appropriate countermeasures. A useful characterization of threats includes identifying the following attributes: threat agent, asset target, motivation, expertise, previous reconnaissance activities, and suspected attack methods. Additionally, the likelihood and subsequent consequences as a result of a successful compromise by each threat is estimated and given a value. This value is called an exposure rating [83]. The SoS, undertaken in step one, provides a useful starting point for estimating the value of the assets that in turn is used to calculate the exposure rating.

Step 3 Risk assessment

Countermeasures are implemented to protect assets from compromise by threats. In order to determine if assets are vulnerable, the current and proposed countermeasures must be examined. If the countermeasures are not sufficient to protect an asset, a vulnerability exists. The goal of a risk assessment is to estimate the likelihood that a threat, in the presence of existing or proposed security countermeasures, will exploit a vulnerability. Risk is a function of the impact of compromise and the likelihood it will occur [84]. If the proposed countermeasures are sufficient to counter the threats to a level of risk acceptable to the organization, the network is in an equilibrium state. If the proposed countermeasures are not sufficient to counter the threats to a level of risk acceptable to the organization, the network is in a vulnerable state.

Step 4 Recommendations

Recommendations are intended to improve the security posture of the network. If the level of risk has to be reduced to meet organizational requirements, the appropriate countermeasures would be proposed in this section. This includes countermeasure implementations that offer sufficient protection at optimal cost.

4.1.2. Survivable Network Systems

Survivability is an emerging discipline developed at Carnegie Mellon University. This initiative seeks to improve computer security by advocating the design of systems that can deliver mission critical services in a timely manner in the presence of attacks, failures, or accidents [85]. The term system can include not only a single system, but also a network or even a collection of networks. The mission of the system is the high-level set of objectives that the system strives to provide. Systems contain two mutually exclusive types of services dictated by the mission specification and are defined as either essential or non-essential. Essential services are those services that must be maintained even while the system is under attack. Non-essential services can be interrupted by the attack or even by the system itself to ensure essential service delivery continues.

Missions will vary among networks depending on business requirements. Uninterrupted world wide web connectivity is crucial for organizations that service customers via on-line purchasing of their product. Other organizations will consider remote network connectivity as essential so that their customers, business partners, and employees can access crucial business functions. It is important that the overall mission of the system is clearly defined so that the appropriate security posture can be adopted.

Survivability is a concept that augments traditional security assessments like the TRA to provide increased protection and increased incident recovery capabilities. As previously

stated, a TRA examines current and/or proposed safeguards to ensure the network is protected within an acceptable level of risk. Risk acceptance acknowledges that the possibility of a successful attack and subsequent compromise exists. Defining essential or mission critical services and their associated availability requirements are a part of a TRA. The security countermeasures, typically specified in a TRA, ensure mission critical service delivery by resisting attacks. After a successful attack, a TRA does not define or assess how uninterrupted mission critical service delivery will occur. The TRA process seeks to provide a solution that avoids compromise altogether. Survivability looks at how systems can continue delivering mission critical services even after a compromise has occurred.

Survivability is an important concept because it is widely recognized that no amount of security can make a system that is connected to an unbounded network impenetrable to attacks and potential compromise [85]. This is a departure from more traditional view of the network fortress model of security. The network fortress security model viewed a network almost as a physical structure. This led to networks that had strong network boundary defences that typically consisted of firewalls and IDS. Little thought was given to securing within the network perimeter. Survivability seeks to transcend the fortress model of security. The fortress model of a computer is no longer valid in modern network environments because [86]:

- Systems are open: Internet connectivity ensures rapid and efficient access to suppliers, customers, and business partners. This is essential in a competitive business market. However, Internet connectivity exposes businesses to open and unsecured systems.
- Central security administration: System administrators cannot impose security restrictions on users outside of their specific administrative domain.
- Untrusted insiders: In an unbounded network, everyone can be considered an untrusted insider.

Survivable systems do not view computer security as containing only two states: compromised or secure [87]. Survivable systems possess four key components [85]:

- Be resistant to attacks: Systems must have the proper countermeasures employed to be able to resist attacks. Adopting a layered network security approach is one way to effectively resist attacks [88].
- Recognize attacks and extent of damage: Systems must be able to recognize when an attack is occurring, instigate protection mechanisms, and then assess the extent of a compromise.
- Recover full and essential services after attack: If an attack is successful, systems must have the ability to recover from the attack to deliver essential services.
- Adapt and evolve to reduce effectiveness of attacks: A system should learn from previous attacks and modify itself to lessen the effects of subsequent attacks.

The Survivable Network Analysis (SNA) method assesses the survivability properties of an existing system, a proposed system, and the impact of system modifications. This method was developed by the CERT/CC and provides a mechanism for an organization to understand the concept of survivability in the context of their operational environment. The method is composed of four steps [89]:

- System definition: Articulate the mission objectives and risks associated with the operational environment.
- Essential capability definition: Identify the essential services that must be maintained and the essential assets that have to be protected.
- Comprimisable capability definition: The development of intrusion scenarios based on the operational risk and intruder capabilities.
- Survivability analysis: Identify essential and comprimisable weaknesses in the system. Weaknesses are analyzed for the key survivability properties of recognize, resist, recover, and adapt.

Once completed, the SNA method provides strategies to improve survivability's four key components of recognize, resist, recover, and adapt.

4.1.3. Layered Network Security

"Layered network security involves the development of sound organizational security policies and the strategic deployment of appropriate risk-based security measures thereby reducing the possibility of circumvention through single points of failure [90]."

Layered network security moves beyond the fortress model of security. The fortress model of security only works until one of the security devices fails, contains an exploitable vulnerability, or a new type of network attack is executed. If these network border security devices are circumvented, then unauthorized access to the internal network can be gained. It is obvious that an extension of the perimeter deployment approach is required to adequately safeguard the network. Adding layers of security throughout the network, from the network perimeter to the host level, mitigates the risk of unauthorized network access.

The use of multiple security technologies deployed in parallel, at all network access points, is considered best practice. Network defenses must protect the network by: monitoring, restricting, and alerting malicious network traffic before it passes the network boundary. Security implemented at both the host and network level increases overall security. Layered network security aids in achieving system survivability [88]. Layered network security strives to remove single points of failure. This increases the chance of detection, resistance, and containment of attacks.

One strategy, to achieve a layered network security posture, is to define logical security perimeters within the network [88]. Defining perimeters makes it easier to determine the security countermeasures required to defend the assets within. Security perimeters will

differ for a variety of reasons such as: threat exposure level, perceived threats, and asset valuation. Security perimeters typically require perimeter specific security countermeasures. Security countermeasures can be IT or non-IT in nature and together they form the security posture for the network perimeter.

Several security perimeters may exist within a single organization. If each perimeter is considered in isolation, security countermeasure deployment may occur without consideration of the security requirements of other perimeters. The deployed security countermeasures work in isolation defending their respective perimeters. This is known as the island security perimeter model [88]. This model is especially apparent when divergent networks have been brought together as a single network as a result of a business merger or for information sharing requirements.

Another method to deploy security countermeasures is to consider the security requirements of all network security perimeters before deploying countermeasures. This is known as the ring security perimeter model [88]. This model is more effective than the island security perimeter because the security countermeasures deployed in each perimeter work together to increase the overall level of security in the network.

Interactions between security countermeasures must also be considered. A security countermeasure used to mitigate a threat can actually weaken another security countermeasure. For instance [91]:

- VPN: A VPN allows secure remote access to a network by providing authentication and confidentiality services. Data transmissions are encrypted to ensure confidentiality. Encryption impedes the effective use of IDS. Most IDSs rely on pattern matching to detect malicious activity. If the network traffic is encrypted, it will not be possible to perform pattern matching.

- **Multiple Internet connections:** In order to provide redundancy in a network infrastructure, some organizations may install a second Internet connection. However, this second data line increases the complexity of deployment and administration of network security devices such as firewalls and IDS.
- **Encrypted email:** Encrypted email provides confidentiality to both the sender and receiver of the email. However, it impedes virus scanning at the gateway.

A layered network security approach is one method for a network to achieve system survivability. Implemented correctly, it reduces the possibility of network compromise through single points of failure.

4.2. Proposed DDoS Defence Strategies

The seriousness of DDoS attacks has prompted action within the security community. Consortiums consisting of private industry, government, and academia have all come together to develop guidance on how to stop DDoS attacks.

Mitre Corporation is an American not-for-profit technology organization that provides systems engineering, research and development, and information technology support to the US government. In late February 2000, Mitre Corporation hosted an ISP security summit [92] to develop technical solutions to defeat DoS attacks. The outcome of the summit was a list of immediate security actions for all organizations connected to the Internet. These security actions consisted exclusively of technical countermeasures designed to stop the use of spoofed IP addresses and amplification attacks. No technical advice was offered: to secure systems, maintain service delivery, or limit damage during a DDoS attack.

On November 2, 1999, CERT/CC hosted a Distributed – Systems Intruder Tools Workshop [46]. The purpose of the workshop was to gather security experts in a forum

to address distributed attack tools. It was a 2-day workshop that resulted in a valuable information sharing opportunity for participants. Participants had a chance to discuss past and current experiences dealing with these types of attack tools. There was a general consensus that better training, forensic techniques, and cooperation was required by all organizations to deal with these types of attacks. The workshop resulted in a paper that contained a description of distributed attack tools and mitigation suggestions targeted to four major audiences within the Internet community:

- Managers
- System administrators
- Internet service providers
- Incident response teams

Each audience was given suggestions on how to protect, detect, and react to this threat in the immediate, short, and long term. These groups represent the main stakeholders of the Internet infrastructure. These suggestions, if properly implemented by all groups, would provide a comprehensive protection plan against distributed attack tools. The paper provided an overview of the actions required to stop DDoS attacks. However, the paper fell short of providing the specific details required to achieve these actions.

In February 2000, SANS released a paper entitled, *Consensus Roadmap for Defeating Distributed Denial of Service Attacks* [49]. The purpose of the paper was to examine the factors that make these types of attacks possible in the hopes of providing solutions to reduce their threat. The paper suggested immediate actions that needed to be taken by the Internet community. These suggestions mimicked those proposed in both the CERT/CC workshop and Mitre ISP summit. Additionally, it gave guidance and direction for long-term research and development to manage these threats.

4.3. DDoS Detection Tools

Security devices such as commercial virus scanners, file integrity checkers, and IDS all have the capability to detect DDoS tools. These tools look for known DDoS signatures (in network activity) or variations (in file systems) to detect the presence of DDoS activity. There are also a number of programs specifically designed to detect DDoS tools or DDoS activity. They vary in method and capabilities and are listed in Table 10: DDoS Detection Tools. All of these DDoS detection tools are available for download at [43].

DDoS Detector	Description	DDoS tools detected
RID	A C program that sends out datagrams that mimics DDoS communications and then listens for appropriate replies.	<ul style="list-style-type: none"> • Trinoo attacker • TFN attacker • Stacheldraht attacker
Gag	A C program that mimics stacheldraht handler datagrams and then listens for a response.	<ul style="list-style-type: none"> • Stacheldraht agents
Dds	A program that mimics various handler datagrams to detect DDoS agents.	<ul style="list-style-type: none"> • Trinoo agents • TFN agents • Stacheldraht agents
Find_DDoS	A program that looks at all 32-bit ELF format files in a directory tree and compares this against a set of known signatures for TFN and trinoo tools.	<ul style="list-style-type: none"> • Mstream handler • Mstream server • Stacheldraht attacker • Stacheldraht agent • Stacheldraht handler • TFN attacker • TFN agent • TFN2K attacker • TFN2K agent • Trinoo agent • Trinoo handler
Zombie Zapper	A program that mimics the shutdown commands of DDoS clients. Assuming the default passwords for the tools have not been changed, the agents will be disabled.	<ul style="list-style-type: none"> • Trinoo agent • WinTrinoo agent • TFN agent • Stacheldraht agent • Shaft agent

Table 10: DDoS Detection Tools

Chapter 5 DDoS Defence Strategies

5.1. Problem Definition

DDoS attacks are one of the most serious threats facing the Internet today. A single attacker using a DDoS attack tool is able to direct thousands of systems in a coordinated DoS attack against a target system. DDoS attacks overwhelm a system or network by sending large amounts of network traffic, thereby exhausting the target's computational or communication resources.

Implementing an effective network security posture to defend against these types of attacks is a challenge. The best network security practices may protect a network from being the source of a DDoS attack, but it will not stop the network from being a victim of a DDoS attack. As long as there are weakly defended systems on the Internet, the potential for DDoS attacks exists. If all network owners implemented a strong network security posture, the number of DDoS attacks would be reduced. Unfortunately, relying on the security of other networks is not practical. It is infeasible to secure the entire Internet; therefore, DDoS attack mitigation strategies need to be put in place to protect the network.

It is difficult to defend against DDoS attacks at the individual network level. Once a DDoS attack begins, network traffic floods the network which in turn exhausts network resources. The result is wide scale system failures that make a network based response impossible. Defending a network from these types of attacks requires active cooperation from the owners and operators of the upstream network infrastructure. ISPs may be able to stop some of the attack by implementing filters on their core routing infrastructure. To exacerbate the problem, network traffic is typically spoofed which makes it difficult to locate attacking systems. As discussed in the previous section (refer to section 2.1.4

Tracing Spoofed IP Datagrams Back to the Source), tracing spoofed IP datagrams to their true origin is a slow, manual, and difficult process. Although there are proposed solutions to aid in tracing spoofed IP datagrams, these solutions are still academic and usually involve substantial changes to the infrastructure. Even if attacking systems are located, they are typically compromised and unwitting participants in the attack.

DDoS tools are becoming increasingly sophisticated and are designed to overcome existing network security countermeasures. The latest DDoS tools make use of encrypted one-way command and control communications, automated code updates, and a variety of self-protection mechanisms. These tools have the capability to evade detection and implement a variety of attacks. The network attack tools used to gather systems for use during the mass intrusion phase are automated, easy to use, and efficient.

5.2. Designing a DDoS Resistant Network Architecture

A DDoS-resistant network architecture has to accomplish two main objectives:

1. To secure the network against participating in a DDoS attack by protecting network assets against compromise and subsequent use in a coordinated attack.
2. To maintain mission critical service delivery, while a DDoS victim, by implementing the necessary technical and non-technical countermeasures to effectively manage a DDoS attack.

The first objective can be met by employing layered network security. A layered network security approach ensures that the organization does not rely upon only one type of technology to solve all security issues. Layered network security strives to reduce the possibility of compromise by identifying and eliminating single points of failure. The security countermeasures implemented, as a result of adopting this approach, will serve to protect the network from DDoS specific attacks and other classes of network attacks.

The second objective is a much harder to achieve. Even under a DDoS attack, the network architecture has to be robust enough to deliver mission critical services. In essence, the network must become a survivable system. Every network has a failure point where it will cease to provide network service, regardless of the availability strategy undertaken. Networks are comprised of finite resources and in a large enough attack will become overwhelmed by system failure or bandwidth consumption. Addressing availability in the network design will substantially increase a network's resistance against DDoS attacks.

Fortunately, adopting a layered network security approach makes a network secure and survivable. As discussed in a paper I co-authored, *The role of Layered Network Security in Achieving System Survivability* [88], additional layers of defence enhance the survivability of a network. Defending a network requires the proper deployment of security countermeasures. Separating the network into logical security perimeters allows for the proper selection of security countermeasures. This approach was used to develop the DDoS resistant network architectures proposed in the following sections.

5.3. Securing the Network Against Participating in a DDoS Attack

The mass intrusion phase of a DDoS attack involves locating and penetrating weakly secured systems across the Internet. This section describes the security countermeasures used by the proposed network architectures to secure the network against compromise and subsequent use as part of a DDoS attack. It involves three distinct, but related activities:

1. Securing the network
2. Securing the hosts that comprise the network
3. Security posture maintenance

Care must be taken to ensure that the overall security of the network is improved with the deployment of each additional countermeasure.

“Security services designed to work in concert, not isolation achieve a higher security posture. [88]”

5.3.1. Network Security Overview

The following section describes the security devices used to secure the three DDoS resistant network architectures proposed in this study.

5.3.1.1. Router

A router is a device that joins multiple networks together. It is also referred to as a layer 3 gateway because it operates at layer 3 of the OSI model. Routers determine optimal paths between networks and send datagrams accordingly. Routers are the first entry points into a network and are positioned on the outermost edge of the network boundary. They receive all datagrams sent to the network and permit or deny datagram entry based on their access control lists (ACL). Routers are the first lines of defence for the network. They can prevent malicious traffic from entering the network boundary and stop certain types of spoofed datagrams from entering or leaving the network. Configuration recommendations for the router are [93, 94, 95]:

- Software: Use the latest version of applicable router software.
- Router security policy: Create and maintain a router security policy. This policy should specify how router maintenance and configuration occurs.
- Access control list: Implement the most restrictive security stance. Deny services and protocols not expressly permitted by the network security policy.
- Egress and ingress filtering: Implement anti-spoofing measures through egress and ingress filtering.

- Use the IP verify unicast reverse-path interface command: Every datagram received on the router interface should be examined to determine if the datagram's source IP address corresponds to a valid route in the Cisco Express Forwarding (CEF) table. Properly implemented, this configuration will prevent attacks that rely on source IP address spoofing (e.g. Smurf attacks).
- Use rate limiting filters: Implement rate limiting, via the committed access rate (CAR) service, to control the amount of ICMP, TCP, and SYN datagrams that enter the network. Rate limiting may assist in stopping some forms of a DDoS attack flood from entering a network. Normal rates of these network datagrams should be determined before a limit is set.
- Logging: Log and record the security events as dictated by the network security policy. Events can include interface status changes, configuration changes, and privilege level changes.
- Suppress admin prohibited messages: All admin prohibited messages should be suppressed. If admin prohibited messages are sent the router will notify senders of network prohibited protocols.
- Block broadcasts: Broadcast or limited broadcast requests should be blocked and not allowed to leave or enter the network.

5.3.1.2. Firewall

A firewall is a network security device designed to prevent unauthorized access between networks. All datagrams entering or leaving the network should pass through a firewall. Firewalls examine and compare each datagram against its rule set. Any datagrams that do not comply with the rule set will be blocked from either entering or leaving the network. Two main categories of firewalls are [96]:

- Circuit layer gateways: These firewalls operate at the transport and network layers. The decision to allow or block datagrams is based on the content of the TCP/IP

header. The source IP address, destination IP address, protocol type, and port are all used to make datagram forwarding decisions.

- Application gateways/proxies: These firewalls operate at the application layer and inspect application data for disallowed commands or suspicious data. These types of firewalls work on behalf of internal systems. They intercept connection requests and forward them to the destination system. These firewalls act as a server accepting datagrams from a client. In order to accept the datagram, the firewall must be running a proxy for the application (e.g. HTTP, FTP, Telnet) used by the client. If the connection is allowed, the firewall recreates the datagram and sends it to the real server. When the server sends the response, the firewall intercepts it and returns the results to the internal system.

Another important function that firewalls provide is network address translation (NAT). NAT translates IP addresses used within one network into IP addresses known within another network. The firewall replaces all internal source IP addresses in datagrams destined to leave the network with its own IP address. When datagrams leave the internal network, their source IP address is now the firewall's IP address. An outsider will only be able to determine the IP address of the firewall and not the number and IP addressing scheme of internal systems.

5.3.1.3. Split DNS server

The domain name system (DNS) server translates domain names into IP addresses. A single DNS server can be used on a network for both internal and external IP address translation. However, this scheme exposes the structure of the internal network to the outside world. An attacker can use this information to gain valuable insight into the network topology. Using a split DNS server model helps keep internal network addresses from being revealed outside the internal network boundary. Two independent DNS servers are used to service the internal and external DNS requests respectively. The

internal DNS server contains the database of all DNS names within the organization. The external DNS server contains only DNS names dealing with the external presence such as: the firewall, router, and web server. There are firewalls that can also implement a split DNS functionality.

5.3.1.4. Virtual Private Network (VPN)

Remote access allows users to connect directly into an organization's internal network to access systems or execute transactions. This technology lends itself well to a mobile workforce by allowing any authorized user to access the network from any location at any time thus increasing productivity. This creates what is known as a VPN. However, every remote access solution will have to address the critical security elements of authentication and confidentiality. Only authorized users should be able to access the network; therefore, some form of authentication will have to be implemented. Additionally, the data they access and send to the network must be protected from unauthorized disclosure through the use of encryption.

5.3.1.5. Intrusion Detection Systems (IDS)

IDS are an integral part of any layered network security solution. There are three main categories of IDS [79]:

- **Host:** Suspicious connections to the host are monitored and logged as an alarm. IDS software is installed on the system and alarms are sent to a system called the IDS console for viewing. The IDS typically utilizes the pre-existing audit subsystem residing on the system to generate their alarms.
- **Application:** It is similar to a host-based IDS. However, it is focused on a specific application or subsystem resident on a system.
- **Network:** A network-based IDS monitors a network segment for malicious activity. IDS software is installed on a system called a sensor and is connected to the network

to be monitored. Most commercially available network IDS are signature based. A signature based network IDS contains a database of known network attacks. It uses its database of signatures to compare against the network traffic it is monitoring. If the network traffic matches an attack signature, an alarm is generated and sent to the IDS console.

IDS deployment considerations for network-based IDS:

- Network perimeter:
 - In front of the network firewall: Placing a sensor in front of the firewall will allow the IDS to detect malicious network traffic directed at the network. This type of deployment typically generates a large number of alarms that may require significant technical resources for subsequent incident investigation.
 - In back of the network firewall: Placing a sensor in back of the firewall will allow the IDS to detect only the malicious activity that passed through the firewall. This will reduce the amount of alarms generated and subsequently the resources required to investigate them. However, the IDS will not detect malicious activity that does not pass through the firewall.
 - In front and back of the network firewall: Placing an IDS sensor in front and in back of the firewall allows a determination of the total amount of malicious activity on the network and the intrusion attempts that succeeded in penetrating the firewall.
- Network access points:
 - Multiple ingress/egress points: Large networks often have multiple ingress/egress points for Internet traffic. Sensors will have to be placed at all ingress/egress points to ensure that all network activity is monitored. All network access points must be identified to ensure the IDS is monitoring all network traffic.
 - Modems: If the modem pool is located within the protected network perimeter, an IDS sensor may have to be deployed at this access point.

IDS deployment considerations for host-based IDS:

- **Critical systems:** Unlike a network-based IDS sensor that can monitor multiple systems on a network segment, a host-based IDS sensor only monitors one system. Every host-based IDS sensor will require a separate installation of the software on the system it will be monitoring.

IDS deployment constraints:

- **Encryption:** VPN's and systems that use encryption add an additional layer of complexity for network-based IDS. Without knowing the private key that will unlock the session, the IDS will be unable to inspect the data portion of the datagram. In order to maintain the effectiveness of the IDS, either a network IDS sensor must be placed where the session is decrypted or the device that does the encryption must have a host-based IDS.
- **Filtering routers:** A router with an extensive ACL will block only certain malicious network traffic before it reaches the internal network. Tight ACL's are part of a layered network security model. However, the people administering the IDS must be aware that they are not detecting all malicious traffic directed at the network.
- **Switches:** On a switched network, each port of the switch receives only network traffic bound for a specific system assigned to that port. For network-based IDS to work on a switch, one of the ports must be configured to receive a copy of all traffic going to all ports.
- **High speed lines:** Large organizations often utilize high-speed Internet connections. Modern commercial IDS systems face resource limitations as the speed of the line increases. Effective monitoring is dependant upon the speed of the line which influences the choice of: the IDS used, the hardware the sensor is running on, and the alarms that are being used.

IDS vary in terms of cost and capabilities. They are ultimately only as effective as the skills of the analyst(s) interpreting the results. Intrusion detection data analysis is not a trivial undertaking; it requires a considerable breadth and depth of IT security knowledge.

5.3.2. Host Security Overview

The following section describes the host security activities used to secure the three DDoS resistant network architectures proposed in this study.

5.3.2.1. Limit Access to Network Resources

Limiting access to network resources enforces tight control of network assets. Reducing the number of connection and resource requests should reduce the size of log files. Then, it can be easier to detect malicious or suspicious activity. Some ways to restrict access on a network are:

- **VLAN**: A VLAN is an Ethernet switch that allows you to deploy multiple IP subnets within a single LAN. This is an effective method of separating network traffic generated by systems on a network.
- **Ipchains**: Iptables [97] is a program that permits kernel level datagram filtering based on the IP datagram header. Systems can use Iptables to block network traffic that does not conform to its rule set.
- **TCP wrapper**: TCP wrapper [98] is a program that provides a method of access control based on the source and destination IP address of connection requests. It also can log successful or unsuccessful connection attempts.
- **SSH**: Secure Shell [99] is a program that allows secure access to another system over a network. The login, file transfers, even system commands, can be executed using strong authentication and encrypted communications.

5.3.2.2. Limit Resource Sharing

Access to resources such as file shares and printers should be controlled and monitored. By enforcing tight control on these network assets, it ensures asset availability when needed. Limiting access to a smaller pool of authorized users reduces the possibility of abuse. The reduced usage has the added benefit of making malicious activity easier to detect.

5.3.2.3. Host Hardening

The term hardening means to modify a system's default operating system configurations and associated applications to increase its security and performance [100]. Hardening network systems makes the network more resistant to attack. Although the goal is to increase security without sacrificing performance, this can not always be achieved. In some instances, one of these objectives will have to be lessened in order to increase the other objective to a level required by the environment. The overall balance between security and performance is best determined by considering the requirements placed on the system by the network environment in which it will be deployed.

The process of host hardening consists of several steps:

1. System mission identification: Specify in as much detail as possible the services the system will provide to the network.
2. Mission supporting system components and application identification: Identify only those system components and applications that help achieve the identified services.
3. Non-mission supporting system components and application identification: Identify, then disable or delete those system components or applications that do not directly or indirectly support the overall mission of the system.
4. Build and configure: Install the operating system and required applications identified in step 2. Change the default installation of the system by disabling, deleting or never

installing unnecessary applications and files identified in step 3. In addition, modify default configurations, file permissions, and accounts of remaining mission supporting system components and applications. Cryptographic checksums should be generated for all critical system files. These checksums can be used to detect any unauthorized file changes to verify system integrity.

5. Pre-deployment testing: Verify that the required configuration changes have been properly implemented and the security posture of the system is adequate.
6. Deployment: Install the system in the intended environment.
7. Post-deployment testing: Verify that the required configuration changes have been properly implemented and the security posture of the system is adequate. This activity is performed as part of penetration testing (refer to section 5.3.3.1 Penetration Testing).
8. Maintenance: Install the required application version changes, security patches, and updates as they are released.

Simply put, hardening is the process of taking the default insulation of a system, customizing it to reflect a more secure stance, then verifying and maintaining the intended secure state. The process of host hardening makes compromise more difficult. If a hardened system becomes compromised, an attacker may find it challenging to maintain access. Many of the services and applications required for communication and data transfer may not be present (e.g. FTP, compilers, and Telnet). In a layered network security posture, the host is the last line of defense.

System hardening is an essential activity because most software packages, including operating systems, contain a multitude of features that are enabled as part of a default installation. This allows the majority of users to perform successful installations without having to encounter special configuration issues. The problem with feature rich default installations is that many of these features, while contributing immensely to ease of use

and speed of deployment, will produce a system with too many points of possible compromise. This is especially true where the system owner is unaware of the services included in the default installation. Without critical security vendor patches, these “hidden” services will become out of date and insecure.

System hardening should be performed on all systems within a network. The amount of specific hardening procedures will be dictated by the location, importance, and service requirements of the system. There are a number of automated hardening scripts available such as Bastille Linux [101] and YASSP [102].

5.3.2.4. TCP/IP Stack Hardening

Network devices communicate using hardware and software that implements the TCP/IP stack. The default TCP/IP stack configuration is tuned to handle normal network traffic. Normal network traffic is assumed to be non-malicious and non-disruptive to the proper functioning of the system. The TCP/IP stack can be modified to be more resistant against DoS attacks. The modifications can simply mean changing default options such as the number of half-open or idle connections that can be maintained. The automated hardening scripts listed in the previous section provide some TCP/IP stack hardening options.

5.3.3. Network Security Posture Maintenance Overview

The following section describes the security posture maintenance activities that need to be implemented in order to maintain the security of the three DDoS resistant network architectures proposed in this study.

5.3.3.1. Penetration Testing

Penetration testing is an important activity in the layered network security deployment process. It is used to verify that the proposed security posture of the network is adequate and that the implementation was completed properly. Penetration testing ensures that network security components perform their intended functions as expected and that their respective configurations are correct. A thorough penetration test will contain a test plan that includes test cases that fully exercise the security capabilities of the network. This involves a careful consideration of the network devices deployed, identification of potential weak spots, and a thorough knowledge of the operational environment.

Penetration testing should reveal any inconsistencies, omissions, and inadequacies of the proposed network security posture.

Penetration testing can be generalized into two broad categories:

- **Red teaming:** Testing is undertaken without the knowledge of systems administration staff of the target network. It can cover most methods to gain access to system resources including: dumpster diving, social engineering, illicit physical entry, and finally more classic remote technical network penetration techniques. This is a more adversarial approach, but perhaps more realistic because the red team will test from the same vantage as an actual attacker.
- **Blue teaming:** A less adversarial method of testing, it is one that elicits the full system administration staff as part of the testing process. The testing personnel will meet with the staff, map the network, and then involve the staff with the penetration testing.

After the testing, the testing team will work with the systems administration staff to help mitigate any vulnerabilities.

Penetration testing usually employs an automated tool or suite of tools to identify and rate vulnerabilities found in a network. The methodology employed will be similar to the stages of a network attack in section 2.9.2. of this paper. Testing can be performed by any qualified professional; however, it is best practice to employ a third party who did not participate in the design or implementation of the network. Penetration testing is invaluable for identifying weaknesses, improving the standard operating procedures, and training the systems administration staff. Some form of penetration testing should be undertaken as part of regular security auditing to ensure the network security posture is adequate.

5.3.3.2. Update Systems

The emergence of new threats and computer vulnerabilities will require the network to be updated. To detect the most recent attacks, virus scanners and IDS must contain the latest security vulnerability information. These devices can usually be configured to automatically download the latest security updates from their respective vendor websites. In addition to security devices, network hosts must also be updated. When a software vendor detects a security problem in their software, they usually post a security fix or application update on their website. If these security fixes are not implemented, the host will continue running the vulnerable software.

5.3.3.3. Disaster Recovery Plan

After a disaster or unscheduled business interruption, a good recovery plan will ensure graceful business resumption. A disaster recovery plan considers the potential impacts of

each type of event and implements the appropriate contingency recovery strategy. Every organization should have a disaster recovery plan.

5.3.3.4. Logging Strategy

Detecting a network attack is a crucial element to the implementation of a layered network security posture. Attacks must be detected so that an informed damage assessment can be made and an appropriate response taken. Logging can detect malicious activity, verify system integrity and behavior, and reveal violations of the network security policy.

An understanding of the logging capabilities of the network is required in order to develop an adequate logging strategy. All network assets need to be examined for their logging mechanisms to ensure they can fulfill the logging requirements specified in the network security policy. If the logging capabilities of the network are deemed inadequate, additional systems will have to be acquired to supplement the amount and type of logging data collected.

Continual examination of system logs will provide a baseline for individual systems and the entire network. This baseline will be the benchmark that will be used to periodically compare the integrity of the systems. This baseline can include many different types of logs such as: network traffic, system and network performance, processes, files, and user activity.

Logging, while an integral part of layered network security, can be challenging to efficiently implement. Even the smallest network can generate enormous amounts of logging data. Additionally, someone has to collect logs from all the systems on the network. One solution to alleviate some of the overhead introduced by logging is to use a Syslog server [103].

Syslog was originally used on UNIX systems to collect application, operating system, and network logging. Many network devices can be configured to generate Syslog messages. A Syslog server can create a central log repository for the network. The use of a Syslog server is not only beneficial for its automatic log harvesting, but also for data protection. Once critical security data is removed from individual systems, the data can no longer be deleted or modified during a system attack.

5.3.3.5. Virus Scanning

Virus scanning is used to detect malicious code entering the network. Virus scanning, at the network gateway, stops the user from receiving suspicious emails and attachments. Gateway virus scanning is transparent to the end user and infected emails are stopped before they are delivered. In addition, virus scanning should also be performed at the user desktop. Scanning at the desktop will detect viruses that did not originate from the network. Viruses can also be found on infected media (e.g. zip or floppy drive) that are executed and saved to the hard drive of a desktop. Virus scanning at both the gateway and desktop provides a layered defence against viruses.

5.3.3.6. Backups

Data backup protects against loss due to unforeseen events such as: hardware/software failures, viruses, intruders, and environmental disasters. All valuable data should be regularly backed up. Daily backups can be required for critical applications, while other applications may only require a weekly backup. It is always preferred to store backups at an off-site secure facility. In the event of a catastrophe, data will still be available and intact. It is also important to test the backups regularly. This will ensure the data can be restored without error and in a timely fashion.

5.3.3.7. Periodic Network Security Audits

A good network security posture includes scheduling periodic security audits. These audits will verify that the security policy is being adequately enforced and that no security issues have occurred. Modifications to the network topology and the discovery of new threats may also necessitate impromptu non-scheduled security audits.

Security audits may consist of a variety of activities including:

- Network/system penetration testing
- Audit log reviews
- Security policy and procedures review
- System integrity checks

A well-run security audit will consider the latest threat and security vulnerability information. Information sources such as vendor websites, security news portals, and government coordination centres can all be used to gather the latest computer security information.

5.4. Maintain Mission Critical Service Delivery while a DDoS Victim

A DDoS attack is designed to saturate a network with spurious network data. Flooding involves sending inordinate amounts of network traffic to a system until: servicing the bogus requests overwhelms the system, the network infrastructure becomes overwhelmed, or all network bandwidth is consumed. This section describes the techniques used by the proposed network architectures to ensure uninterrupted service delivery to external customers even while under a DDoS attack.

5.4.1. Load Balancing Using Server Clusters

One method to deal with distributed attacks is with a distributed solution. If processing and communications requirements are spread evenly over multiple systems, then no single point of failure will exist. A load balancer is a hardware or software device that can scale the performance of many server-based programs such as a web server. Using a load-balancing algorithm, these devices distribute connection attempts and requests to the web servers among multiple servers. This process is transparent to the client as the load balancer takes care of maintaining sessions between the client and the servers. This configuration of multiple servers is called a cluster.

A cluster is a group of two or more systems working together to provide a service. Clustering is usually performed in one of two modes: active-to-active or active-to-passive [104]. In the active-to-active mode, all applications provide the service simultaneously and when one node fails the others compensate. In the active-to-passive mode, only one main node performs the application and when that node fails a standby node automatically takes over.

Load balancing optimizes the use of all available resources. If client requests vary drastically within a network infrastructure, load balancing can re-distribute service requests away from those nodes that are most stressed. It is also effective for dealing with DDoS floods. In addition to managing client server requests, load balancers can be used to spread network traffic among network infrastructure devices such as: routers and firewalls.

5.4.2. Redundant ISP Connections

The use of redundant ISP connections provides the network with two separate gateways to the Internet. Multiple Internet gateways ensure that connection to the Internet can be

maintained when one gateway fails. If a DDoS attack occurs on one network connection, that connection can be blocked. Then, network communications can be routed through another connection. In addition to providing twice the bandwidth of a single Internet connection, multiple gateways provide the opportunity for load balancing. Data can be transported on alternate routes to the Internet. In order to ensure that the connections are truly redundant, the selection process for ISPs should consider the common infrastructure that exists between them. Minimizing the amount of common infrastructure between the two selected ISPs will reduce the risk of a single DDoS attack effecting both network connections.

5.4.3. Good Working Relationship with ISP

It is important to maintain a good working relationship with ISP personnel. They will be able to provide insight into the quantity and type of traffic entering the network. ISP staff can also help trace spoofed IP datagrams back to the source. Filters placed on their infrastructure can reduce the amount of the attack flood. Therefore, filtering applied to the ISP infrastructure may provide some relief from a DDoS attack. Specific attack floods can even be blocked from entering the network.

In the event of a DDoS attack, email and other electronic communications methods may not be available. Even if the network is under attack, establishing out of band communication methods with the ISP allows information to be exchanged.

5.4.4. Firewall Load Balancing

Load balancers on a firewall can be implemented in either hardware or software; each solution has advantages and disadvantages [105]. Implemented correctly, load balanced firewalls eliminate the possibility of the firewall becoming a single point of failure. A software load balancing solution requires that software be installed on the firewalls. This

may affect the performance of the firewalls because some system resources may be required by the load balancing software. If a network upgrade or expansion occurs, potential scalability and licensing issues may arise.

Hardware implemented firewall solutions vary greatly among vendors. Typically, a hardware solution is a physical standalone component that directs the flow of datagrams between the firewalls. Each firewall processes network traffic and the load balancer acts as a network traffic director. In conjunction with firewalls, layer four switches can provide: load balancing, redundancy, and network health checking [106].

5.4.5. Fail-Over Routers

Routers provide the connectivity among different networks. In the event of system failure multiple routers in a network provide redundancy. If one router fails, due to system failure or malicious activity, all legitimate network traffic can be diverted to other routers. Where the routers connect the network to separate ISPs, the redundancy is further increased. If a DoS attack occurs on one ISP connection, the attack can be foiled by shutting down the corresponding router and diverting all network traffic to the other ISP connection.

5.4.6. Additional Bandwidth

An assessment of normal network activity will determine the bandwidth required to service normal network requirements. Bandwidth capacity should be periodically monitored to ensure it continues to meet requirements. Additional network bandwidth is useful when coping with a DDoS attack. The more bandwidth a network has the larger the attack must be in order to disrupt service delivery.

5.5. DDoS Resistant Network Architectures – A Case Study

The previous sections provided the necessary theory and methodologies to enable the development of secure and highly available network architectures. This section strives to move beyond the theoretical and propose three practical examples of DDoS resistant network architectures.

Due to varying organizational requirements and resource constraints, adopting exact instantiations of these network architectures might not be possible. No single network architecture solution will work for every organization. Each organization must develop its own network design in accordance with its operational environment specified in its TRA.

Therefore, these network architectures do not conform to a particular network security policy or TRA. Security policies and TRAs are site specific and as such are undertaken on a case by case basis. These proposed network architectures have been designed to protect an Internet connected network against a DDoS attack. Included with each network architecture diagram, is a description of the applied security technologies and specific configuration recommendations. These descriptions are meant to provide a framework in which the overall security posture of a network can be assessed. Informed decisions can then be made about the network design, cost, and the organizational needs of the users.

Any product-specific recommendations of the different technologies such as: load balancing, IDS, and networking infrastructure have been avoided. Products vary in capability and function depending on the operational environment in which they are deployed. The goal of this paper is not to recommend specific products or solutions, but to propose practical DDoS resistant network architectures.

5.5.1. Sample Network Requirements

In order to provide the necessary context for the proposed network architectures, the organizational requirements have to be defined. This includes the operational environment, user connectivity requirements, and a definition of the mission critical service that must be provided. The requirements for the three proposed networks are specified in Table 11: Sample Network Requirements. Three sample networks (A, B, and C) were chosen to show different DDoS mitigation strategies that can be employed. Network A services a small organization with limited resources. Network A has been adequately secured, but some availability options have not been adopted (due to the cost and its lower availability requirements). Network B services a medium-sized organization with moderate resources. Network B has been adequately secured and its higher availability requirements have been addressed.

Network A Small organization	Network B Medium organization	Network C Large organization
<ul style="list-style-type: none"> • small user population <100 • Internet connectivity • remote access required via a VPN • not geographically distributed • medium asset valuation • limited downtime of mission critical service is acceptable. < 20 hours per year of downtime is not acceptable • mission critical service – HTTP (web) access to corporate web server from the public 	<ul style="list-style-type: none"> • medium user population <1000 • Internet connectivity • remote access required via a VPN • not geographically distributed • medium asset valuation • very limited downtime of mission critical service is acceptable. < 15 hours per year of downtime is not acceptable • mission critical service – HTTP (web) access to corporate web server from the public 	<ul style="list-style-type: none"> • large user population >1000 • Internet connectivity • remote access required via a VPN • geographically distributed • high asset valuation • extremely limited downtime of mission critical service is acceptable. < 6 hours per year of downtime is not acceptable • mission critical service – HTTP (web) access to corporate web server from the public

Table 11: Sample Network Requirements

Network C is a large geographically distributed organization with vast resources.

Network C has high security and high availability requirements that have been addressed in its network design.

5.5.2. Network A

The organizational requirements for Network A, listed in Table 11: Sample Network Requirements, are typical for a small security conscious organization. The network connects to the Internet and has been divided into three logical security domains. Given its organizational requirements and the associated cost, the overall security posture and availability strategy employed is appropriate for the network. An overview of the security countermeasures and availability strategy implemented for Network A is found in Tables 12 to 16 inclusively.

Network A – Security Domains	
Security domain 1	The most untrusted security domain. It contains all external systems outside the organizations network boundary.
Security domain 2	The Demilitarized Zone (DMZ). It contains critical network servers that provide services to both external untrusted users and internal trusted users. It is not part of the untrusted network or the trusted network. It connects the untrusted network to the trusted network.
Security domain 3	This is the trusted internal network.

Table 12: Network A – Security Domains

Network A – Network Security Overview	
Router	<ul style="list-style-type: none"> • <u>Egress/Ingress filtering</u> – Stops incoming and outgoing spoofed datagrams. • <u>Limit or eliminate all ICMP traffic penetrating network boundary</u> – Stops some forms of DDoS communications. • <u>Suppress admin prohibited messages</u> • <u>Block ports and services not explicitly permitted</u> • <u>IP verify unicast</u> • <u>ACL</u> - allow Incoming/outgoing DNS port 53 Incoming/outgoing web 80 Incoming/outgoing mail 25 Incoming/outgoing SSH 22 Incoming/outgoing IDS console port Incoming/outgoing VPN port
Firewall – connects all three security domains	<ul style="list-style-type: none"> • <u>Type</u>: Proxy/application level. • <u>Rule set</u>: Restrict all access not expressly permitted. • <u>Use</u>: Network Address Translation (NAT).
IDS – network based	<ul style="list-style-type: none"> • <u>In front of firewall</u>: Detects all malicious network activity that passed through the network border router. • <u>In back of firewall</u>: Detects malicious network activity that penetrated the firewall and has entered the internal network. • <u>Inside the internal network</u>: Detects malicious activity from users that are VPN connected to the network. • <u>Signatures</u>: Implement an IDS signature policy that takes into consideration the type of systems you have deployed and the services offered by the network. Ensure that all DDoS signatures are part of the signature policy.
Virus Scanning (Gateway)	Perform virus scanning at the gateway (e.g. mail server).
Split DNS Server	Use two DNS servers, an internal and external, to keep internal IP addresses protected from outside disclosure.
VPN	Use a VPN to ensure both authentication and confidentiality of communication sessions.
VLAN	Separate the network into logical subnets to reduce the chance of data disclosure due to data sniffing.
Syslog Server	Use a Syslog server, as a central repository of log data, for more efficient log management.

Table 13: Network A – Network Security Overview

Network A – Host Security Overview	
Servers	<ul style="list-style-type: none"> • <u>Hardening</u> – Remove unnecessary programs and services. • <u>TCP/IP Stack hardening</u> – Perform configuration changes to the TCP/IP stack to make it more resistant to DoS attacks. • <u>Host-based IDS</u> – Install host-based IDS on critical servers. • <u>Cryptographic checksums</u> – Calculate cryptographic checksums for programs on critical workstations to detect unauthorized modifications.
Workstations	<ul style="list-style-type: none"> • <u>Hardening</u> – Remove unnecessary programs and services. • <u>Virus scanning at the desktop</u> – Scan for viruses on the local systems. • <u>Cryptographic checksums</u> – Calculate cryptographic checksums for programs on critical workstations to detect unauthorized modifications.
Limit Access	<ul style="list-style-type: none"> • <u>VLAN</u> – Create logical subnets within a network that can prevent illicit network monitoring. • <u>Ipchains</u> – Restrict access to a system based on the IP datagram header. • <u>TCPwrapper</u> – Restrict access to a system based on IP addresses. • <u>SSH</u> – Connect securely to systems on the network.
Limit Resource Sharing	<ul style="list-style-type: none"> • <u>File/printer</u> – Limit printer and file sharing to authorized users.

Table 14: Network A – Host Security Overview

Network A – Network Security Posture Maintenance Overview	
Security Updates	<ul style="list-style-type: none"> • <u>Patches/Service packs/Hot fixes</u> – Ensure all systems have the latest critical patches and updates installed. • <u>New IDS signatures</u> – Ensure that IDS have the latest IDS signatures. • <u>Virus signatures</u> – Ensure that virus scanners have the latest virus signatures.
Logging	Log and examine Router, IDS, Firewall, Server, and Critical host logs.
Incident Management Plan	Develop and endorse a detailed incident management plan.
Backups	Maintain functional data backups at a secure off-site location.
Periodic Security Posture Audits	Undertake regularly scheduled proactive security auditing. Monitor the latest security alerts and advisories from known sources.
Disaster Recovery Plan	Develop and endorse a detailed disaster recovery plan.

Table 15: Network A – Network Security Posture Maintenance Overview

Network A – Availability Overview	
Server Cluster	Use multiple HTTP servers to service incoming requests.
Out-of-Band Communications with your ISP	With your ISP document an alternate communications method that does not rely on normal network operation.
Router Configuration	Implement rate limiting to limit the amount of ICMP, UDP, and TCP SYN datagrams that enter the network.
High Bandwidth	Ensure that the bandwidth requirement of the network can accommodate additional network traffic bursts.

Table 16: Network A – Availability Overview

The network diagram that implements the security posture is Figure 18: Network A Architecture.

Figure 18: Network A Architecture

5.5.3. Network B

The organizational requirements for Network B, listed in Table 11: Sample Network Requirements, are typical of a medium sized security conscious organization. The network connects to the Internet and has been divided into four logical security domains. The overall security posture and availability strategy employed is appropriate for the network given its organizational requirements and the associated cost. An overview of the security countermeasures and availability strategy implemented for Network B is found in Tables 17 to 21 inclusively.

The increased availability requirements of Network B have been addressed through the addition of:

- Another security perimeter protected by an additional firewall
- A redundant ISP connection
- A fail-over router
- Load balanced firewalls

These differences between Network A and Network B are highlighted in Tables 17 to 21 inclusively.

Network B – Security Domains	
Security domain 1	The most untrusted security domain. It contains all external systems outside the organizations network boundary.
Security domain 2	The first DMZ. It contains critical network servers that provide services to both external untrusted users and internal trusted users. It is not part of the untrusted network or the trusted network. It connects the untrusted network to the trusted network.
Security domain 3	The second DMZ. It contains a second firewall that implements a more restrictive rule set than the firewalls in the first DMZ. Additionally, a network-based IDS sensor monitors network traffic entering and leaving the internal network.
Security domain 4	The trusted internal network.

Table 17: Network B – Security Domains

Network B – Network Security Overview	
Edge Routers	<ul style="list-style-type: none"> • <u>Egress/Ingress filtering</u> – Stops incoming and outgoing spoofed datagrams. • <u>Limit or eliminate all ICMP traffic penetrating network boundary</u> – Stops some forms of DDoS communications. • <u>Suppress admin prohibited messages</u> • <u>Block ports and services not explicitly permitted</u> • <u>IP verify unicast</u> • <u>ACL</u> – allow Incoming/outgoing DNS port 53 Incoming/outgoing web 80 Incoming/outgoing mail 25 Incoming/outgoing SSH 22 Incoming/outgoing IDS console port Incoming/outgoing VPN port
Firewall (security domain 2) – connects the 1 st , 2 nd , and 3 rd security domains	<ul style="list-style-type: none"> • <u>Type: Proxy/application level.</u> • <u>Rule set: Restrict all access not expressly permitted.</u> • <u>Use: Network Address Translation (NAT).</u>
Firewall (security domain 3) – connects the 2 nd , 3 rd , and 4 th security domains	<ul style="list-style-type: none"> • <u>Type: Proxy/application level.</u> • <u>Rule set: Restrict all access not expressly permitted.</u> • <u>Use: Network address translation (NAT).</u>
IDS – network based	<ul style="list-style-type: none"> • <u>In front of firewall:</u> Detects all malicious network activity that passed through the network border router. • <u>In back of firewall:</u> Detects malicious network activity that penetrated the firewall and has entered the internal network. • <u>Inside the internal network:</u> Detects malicious activity from users that are VPN connected to the network. • <u>Signatures:</u> Implement an IDS signature policy that takes into consideration the type of systems you have deployed and the services offered by the network. Ensure that all DDoS signatures are part of the signature policy.
Virus Scanning (Gateway)	Perform virus scanning at the gateway (e.g. mail server).
Split DNS Server	Use two DNS servers, an internal and external, to keep internal IP addresses protected from outside disclosure.
VPN	Use a VPN to ensure both authentication and confidentiality of communication sessions.
VLAN	Separate the network into logical subnets to reduce the chance of data disclosure due to data sniffing.
Syslog Server	Use a Syslog server, as a central repository of log data, for more efficient log management.

Table 18: Network B – Network Security Overview

Network B – Host Security Overview	
Servers	<ul style="list-style-type: none"> • <u>Service separation</u> – Separate internal mail and DNS servers. • <u>Hardening</u> – Remove unnecessary programs and services. • <u>TCP/IP Stack hardening</u> – Perform configuration changes to the TCP/IP stack to make it more resistant to DoS attacks. • <u>Host-based IDS</u> – Install host-based IDS on critical servers. • <u>Cryptographic checksums</u> – Calculate cryptographic checksums for programs on critical workstations to detect unauthorized modifications.
Workstations	<ul style="list-style-type: none"> • <u>Hardening</u> – Remove unnecessary programs and services. • <u>Virus scanning at the desktop</u> – Scan for viruses on the local systems. • <u>Cryptographic checksums</u> – Calculate cryptographic checksums for programs on critical workstations to detect unauthorized modifications.
Limit Access	<ul style="list-style-type: none"> • <u>VLAN</u> – Create logical subnets within a network that can prevent illicit network monitoring. • <u>Ipchains</u> – Restrict access to a system based on the IP datagram header. • <u>TCPwrapper</u> – Restrict access to a system based on IP addresses. • <u>SSH</u> – Connect securely to systems on the network.
Limit Resource Sharing	<ul style="list-style-type: none"> • <u>File/printer</u> – Limit printer and file sharing to authorized users.

Table 19: Network B – Host Security Overview

Network B – Network Security Posture Maintenance Overview	
Security Updates	<ul style="list-style-type: none"> • <u>Patches/Service packs/Hot fixes</u> – Ensure all systems have the latest critical patches and updates installed. • <u>New IDS signatures</u> – Ensure that IDS have the latest IDS signatures. • <u>Virus signatures</u> – Ensure that virus scanners have the latest virus signatures.
Logging	Log and examine Router, IDS, Firewall, Server, and Critical host logs.
Incident Management Plan	Develop and endorse a detailed incident management plan.
Backups	Maintain functional data backups at a secure off-site location.
Periodic Security Posture Audits	Undertake regularly scheduled proactive security auditing.
Disaster Recovery Plan	Monitor the latest security alerts and advisories from known sources.
	Develop and endorse a detailed disaster recovery plan.

Table 20: Network B – Network Security Posture Maintenance Overview

Network B – Availability Overview	
Server Cluster	Use multiple HTTP servers to service incoming requests.
Load Balancer	Install a hardware load balancer for HTTP server cluster.
Redundant ISP Connections	Use multiple ISP connections to allow a DDoS attack flood to be blocked and network communications routed through another connection.
Fail-Over Routers	Use fault tolerant router design.
Load Balanced Firewalls	Use fault tolerant firewall design.
Out-of-Band Communications with your ISP	With your ISP document an alternate communications method that does not rely on normal network operation.
Router Configuration	Implement rate limiting to limit the amount of ICMP, UDP, and TCP SYN datagrams that enter the network.
High Bandwidth	Ensure that the bandwidth requirement of the network can accommodate additional network traffic bursts.

Table 21: Network B – Availability Overview

The network diagram that implements the security posture in Tables 17 to 21 inclusive is Figure 19: Network B Architecture.

Figure 19: Network B Architecture

5.5.4. Network C

The organizational requirements for Network C, listed in Table 11: Sample Network Requirements, are typical for a large geographically separated security conscious organization. The network connects to the Internet and has been divided into seven logical security domains. The east coast network contains four logical security domains and the west coast network contains three logical security domains. The overall security posture and availability strategy employed is appropriate for the network given its organizational requirements and the associated cost. An overview of the security countermeasures and availability strategy implemented for Network C is found in Tables 22 to 26 inclusively.

The increased availability requirements of Network C have been addressed through the addition of:

- Internet connectivity redundancy through the use of a leased data line to a trusted network

The differences between Network B and Network C are highlighted in Tables 22 to 26 inclusively.

Network – Security Domains	
East Coast Network	
Security domain 1	The most untrusted security domain. It contains all external systems outside the organizations network boundary.
Security domain 2	The first DMZ. It contains critical network servers that provide services to both external untrusted users and internal trusted users. It is not part of the untrusted network or the trusted network. It connects the untrusted network to the trusted network.
Security domain 3	The second DMZ. It contains a second firewall that implements a more restrictive rule set than the firewalls in the first DMZ. Additionally, a network-based IDS sensor monitors network traffic entering and leaving the internal network.
Security domain 4	The trusted internal network.
West Coast Network	
Security domain 1	The most untrusted security domain. It contains all external systems outside the organizations network boundary.
Security domain 2	The Demilitarized Zone (DMZ). It contains critical network servers that provide services to both external untrusted users and internal trusted users. It is not part of the untrusted network or the trusted network. It connects the untrusted network to the trusted network.
Security domain 3	The trusted internal network.

Table 22: Network C – Security Domains

Network C – Network Security Overview	
Edge Routers (east and west coast)	<ul style="list-style-type: none"> • <u>Egress/Ingress filtering</u> – Stops incoming and outgoing spoofed datagrams. • <u>Limit or eliminate all ICMP traffic penetrating network boundary</u> – Stops some forms of DDoS communications. • <u>Suppress admin prohibited messages</u> • <u>Block ports and services not explicitly permitted</u> • <u>IP verify unicast</u> • <u>ACL</u> – allow Incoming/outgoing DNS port 53 Incoming/outgoing web 80 Incoming/outgoing mail 25 Incoming/outgoing SSH 22 Incoming/outgoing IDS console port Incoming/outgoing VPN port
Firewall (east coast: security domain 2) – connects the 1 st , 2 nd , and 3 rd security domains	<ul style="list-style-type: none"> • <u>Type: Proxy/application level.</u> • <u>Rule set: Restrict all access not expressly permitted.</u> • <u>Use: Network Address Translation (NAT).</u>
Firewall (east coast: security domain 3) – connects the 2 nd , 3 rd and 4 th security domains	<ul style="list-style-type: none"> • <u>Type: Proxy/application level.</u> • <u>Rule set: Restrict all access not expressly permitted.</u> • <u>Use: Network Address Translation (NAT).</u>
Firewall (west coast)	<ul style="list-style-type: none"> • <u>Type: Proxy/application level.</u> • <u>Rule set: Restrict all access not expressly permitted.</u> • <u>Use: Network Address Translation (NAT).</u>
IDS – network based	<ul style="list-style-type: none"> • <u>In front of firewall:</u> Detects all malicious network activity that passed through the network border router. • <u>In back of firewall:</u> Detects malicious network activity that penetrated the firewall and has entered the internal network. • <u>Inside the internal network:</u> Detects malicious activity from users that are VPN connected to the network. • <u>Signatures:</u> Implement an IDS signature policy that takes into consideration the type of systems you have deployed and the services offered by the network. Ensure that all DDoS signatures are part of the signature policy.
Virus Scanning (Gateway)	Perform virus scanning at the gateway (e.g. mail server).
Split DNS Server	Use two DNS servers, an internal and external, to keep internal IP addresses protected from outside disclosure.
VPN	Use a VPN to ensure both authentication and confidentiality of communication sessions.
VLAN	Separate the network into logical subnets to reduce the chance of data disclosure due to data sniffing.
Syslog Server	Use a Syslog server, as a central repository of log data, for more efficient log management.

Table 23: Network C – Network Security Overview

Network C – Host Security Overview	
Servers	<ul style="list-style-type: none"> • <u>Service separation</u> – Separate internal mail and DNS servers. • <u>Hardening</u> – Remove unnecessary programs and services. • <u>TCP/IP Stack hardening</u> – Perform configuration changes to the TCP/IP stack to make it more resistant to DoS attacks. • <u>Host-based IDS</u> – Install host-based IDS on critical servers. • <u>Cryptographic checksums</u> – Calculate cryptographic checksums for programs on critical workstations to detect unauthorized modifications.
Workstations	<ul style="list-style-type: none"> • <u>Hardening</u> – Remove unnecessary programs and services. • <u>Virus scanning at the desktop</u> – Scan for viruses on the local systems. • <u>Cryptographic checksums</u> – Calculate cryptographic checksums for programs on critical workstations to detect unauthorized modifications.
Limit Access	<ul style="list-style-type: none"> • <u>VLAN</u> – Create logical subnets within a network that can prevent illicit network monitoring. • <u>Ipchains</u> – Restrict access to a system based on the IP datagram header. • <u>TCPWrapper</u> – Restrict access to a system based on IP addresses. • <u>SSH</u> – Connect securely to systems on the network.
Limit Resource Sharing	<ul style="list-style-type: none"> • <u>File/printer</u> – Limit printer and file sharing to authorized users.

Table 24: Network C – Host Security Overview

Network C – Network Security Posture Maintenance Overview	
Security Updates	<ul style="list-style-type: none"> • <u>Patches/Service packs/Hot fixes</u> – Ensure all systems have the latest critical patches and updates installed. • <u>New IDS signatures</u> – Ensure that IDS have the latest IDS signatures. • <u>Virus signatures</u> – Ensure that virus scanners have the latest virus signatures.
Logging	Log and examine Router, IDS, Firewall, Server, and Critical host logs.
Incident Management Plan	Develop and endorse a detailed incident management plan.
Backups	Maintain functional data backups at a secure off-site location.
Periodic Security Posture Audits	Undertake regularly scheduled proactive security auditing. Monitor the latest security alerts and advisories from known sources.
Disaster Recovery Plan	Develop and endorse a detailed disaster recovery plan.

Table 25: Network C – Network Security Posture Maintenance Overview

Network C – Availability Overview	
Server Cluster	Use multiple HTTP servers to service incoming requests.
Load Balancer	Install a hardware load balancer for HTTP server cluster.
Leased Line	Allows for network connectivity to be diverted on demand to a dedicated network connection geographically separated from the network.
Redundant ISP Connections	Use multiple ISP connections to allow a DDoS attack flood to be blocked and network communications routed through another connection.
Fail-Over Routers	Use fault tolerant router design.
Load Balanced Firewalls	Use fault tolerant firewall design.
Out-of-Band Communications with your ISP	With your ISP document an alternate communications method that does not rely on normal network operation.
Router Configuration	Implement rate limiting to limit the amount of ICMP, UDP, and TCP SYN datagrams that enter the network.
High Bandwidth	Ensure that the bandwidth requirement of the network can accommodate additional network traffic bursts.

Table 26: Network C – Availability Overview

The network diagram that implements the security posture in Tables 22 to 26 inclusively is Figure 20: Network C Architecture.

Figure 20: Network C Architecture

Chapter 6 Conclusions and Future Work

6.1. Summary

Recent high profile Internet attacks have increased the awareness of DDoS attacks. Consortiums consisting of private industry, government, and academia have undertaken several initiatives to develop effective countermeasures to stop DDoS attacks. These initiatives have proposed short-term DDoS mitigation strategies and long-term research and development strategies. Despite the best efforts of the security community, these attacks continue to pose a major threat to the Internet.

The main objective of this study was to develop three DDoS resistant network architectures. These network architectures were designed to: (1) secure the network from participating in a DDoS attack and (2) maintain mission critical service during a DDoS attack.

Chapter one examined network security and DDoS attacks. Current statistics reiterated that DDoS attacks are one of the most serious threats facing the Internet. Chapter two provided the necessary background to understand the methodology behind general network and DDoS attacks. Chapter three presented a comparative analysis of two DDoS tools: TFN2K and Plague. In a controlled lab environment, each tool was tested to determine its behavior on a network. This led to the creation of IDS signatures to detect their activity on a network. Chapter four contains secure network architecture design recommendations and overall defensive strategies against DDoS attacks. These strategies include layered network security and survivability. The concepts of layered network security and survivability are central to our proposed DDoS resistant design strategy presented in chapter five. The case study, in chapter five, allowed us to implement this

strategy on three different network architectures. Finally, the thesis concludes in chapter six.

6.2. Conclusions

This study has resulted in three major contributions.

1. The development of three practical DDoS-resistant network architectures. These network architectures minimize the possibility of system compromise and thus DDoS attack participation. Additionally, these architectures have been designed to maintain mission critical services while under a DDoS attack. A case study was used to create realistic network designs based on differing network operational requirements.
2. The development of an extensible DDoS resistant network design strategy. Our network design strategy of adopting both layered network security and survivability can be applied to networks regardless of their size or function. Our case study presented significant guidance on the use of security technologies, security activities, and availability strategies. This allows the reader to understand and evaluate our proposed network designs and apply our network design strategy to their own networks.
3. The creation of intrusion detection signatures for two DDoS tools. The two DDoS tools examined, in this study, varied in capability and function. The development of Plague was incomplete and required an additional third party program to function. TFN2K was complete and contained advanced communications and security countermeasure avoidance techniques. We were able to observe patterns in network traffic generated by both of these tools. These observed patterns allowed us to create the corresponding intrusion detection signatures to detect their activity on a network.

6.3. Recommendations and Future Work

The network architectures proposed in this study were designed to protect a network against DDoS attacks. However, given a large enough attack every network will fail. Defending a network against a DDoS when it is occurring is a challenge. The network infrastructure becomes stressed, as a result of the attack, which makes response and/or recovery difficult. Therefore, we have two recommendations for future work that would aid in coping with a DDoS attack:

ISP – DDoS Response

A proactive DDoS defence strategy would be one in which an ISP could investigate and determine the causes of variances in network traffic. Any network traffic determined to be a DDoS attack flood could be blocked before it ever reached an individual network.

Therefore, we need to: (1) develop better DDoS detection techniques that rely on anomalous network behavior instead of signature comparisons and (2) develop filters that allow DDoS attacks to be blocked at core infrastructure points with minimal impact on the quality of service for individual networks.

Tracing Spoofed IP datagrams

IP address spoofing provides attackers with anonymity during a DDoS attack. We have discussed how tracing spoofed IP datagrams back to their true source is a formidable challenge. Currently, the best method to determine the true source of a spoofed datagram is to manually trace the path of the datagram from router to router until you reach the network of origin. There have been proposed techniques to provide a technical solution to tracing spoofed diagrams. However, these solutions are onerous because they typically require upgrades to the Internet infrastructure. It is neither practical nor feasible to have all network owners adopt this new technology. Unless this functionality can be part of a

default install of new network infrastructure, gaps will remain where the new technology will not be implemented. Therefore, we need to develop a technique that permits tracing of spoofed IP datagrams without significant upgrades to the existing infrastructure.

Acronyms

ACL	Access Control List
B2B	Business to Business
CAR	Committed Access Rate
CIAC	Computer Incident Advisory Capability
CEF	Cisco Express Forwarding
CERT/CC	Computer Emergency Response Team/Carnegie Mellon
CNN	Cable News Network
DDoS	Distributed Denial of Service
DMZ	Demilitarized Zone
DNS	Domain Name Service
DoS	Denial of Service
DoS	Denial of Service
DSIT	Distributed Systems Intruder Tools
FBI	Federal Bureau of Investigation
FTP	File Transfer Protocol
GUI	Graphical User Interface
HTTP	Hyper Text Transport Protocol
ICMP	Internet Message Control Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IRC	Internet Relay Chat
ISP	Internet Service Provider
IT	Information Technology
LAN	Local Area Network
MAC	Media Access Control
NAT	Network Address Translation
NIC	Network Interface Card
NIPC	National Infrastructure Protection Center
OSI	Open Systems Interconnection Reference Model
PC	Personal Computer
POP	Post Office Protocol
RFC	Request for Comments
RPC	Remote Procedure Call
SANS	System Administration, Networking, and Security
SNA	Survivable Network Analysis
SoS	Statement of Sensitivity
SSH	Secure Shell
TCP	Transmission Control Protocol

TCP/IP	Transmission Control Protocol/Internet Protocol
TFN	Tribe Flood Network
TFN2K	Tribe Flood Network 2000
TRA	Threat Risk Assessment
UDP	User Datagram Protocol
VLAN	Virtual Local Area Network
VPN	Virtual Private Network

Appendix A TFN2K

1. TFN2K Screen Shot

usage: ./tfn <options>

- [-P protocol]** Protocol for server communication. Can be ICMP, UDP or TCP.
Uses a random protocol as default
- [-D n]** Send out n bogus requests for each real one to decoy targets
- [-S host/ip]** Specify your source IP. Randomly spoofed by default, you need to use your real IP if you are behind spoof-filtering routers
- [-f hostlist]** Filename containing a list of hosts with TFN servers to contact
- [-h hostname]** To contact only a single host running a TFN server
- [-i target string]** Contains options/targets separated by '@', see below
- [-p port]** A TCP destination port can be specified for SYN floods
- <-c command ID>**
 - 0 - Halt all current floods on server(s) immediately
 - 1 - Change IP antispoof-level (evade rfc2267 filtering)
usage: -i 0 (fully spoofed) to -i 3 (/24 host bytes spoofed)
 - 2 - Change Packet size, usage: -i <packet size in bytes>
 - 3 - Bind root shell to a port, usage: -i <remote port>
 - 4 - UDP flood, usage: -i victim@victim2@victim3@...
 - 5 - TCP/SYN flood, usage: -i victim@... [-p destination port]
 - 6 - ICMP/PING flood, usage: -i victim@...
 - 7 - ICMP/SMURF flood, usage: -i victim@broadcast@broadcast2@...
 - 8 - MIX flood (UDP/TCP/ICMP interchanged), usage: -i victim@...
 - 9 - TARGA3 flood (IP stack penetration), usage: -i victim@...
 - 10 - Blindly execute remote shell command, usage -i command

2. TFN2K Network Data Captures

Snort was used to provide the network trace of the attack. Some snippets of the actual trace were deleted because of repetition; interesting traces were kept for explanation purposes. Some portions of the packet captures are also bolded to draw attention to them.

3. TFN2K Communications

TFN2K handler to agent communication is indicated by a sequence of trailing A's in the data payload portion of the datagram as discussed in chapter three. The following three datagram captures are examples of TFN2K communications using TCP, ICMP, and UDP

datagrams. The source IP address is spoofed because communication to the handler from the agent is not required to execute the attacks.

Handler to agent communication using TCP.

The TCP length field (TcpLen) is zero which should not normally occur. The source IP address is spoofed (197.229.68.0).

```
06/24-12:48:11.592688 197.229.68.0:23954 -> 192.168.0.2:16166
TCP TTL:234 TOS:0x0 ID:46497 IpLen:20 DgmLen:82
***A**S* Seq: 0xE94A51 Ack: 0x0 Win: 0x8FF4 TcpLen: 0
0x0000: 00 50 DA CA 55 87 00 50 DA CA 55 8E 08 00 45 00 .P..U..P..U...E.
0x0010: 00 52 B5 A1 00 00 EA 06 50 74 C5 E5 44 00 C0 A8 .R.....P...D...
0x0020: 00 02 5D 92 3F 3A 00 E9 4A 51 00 00 00 00 00 12 ...?...JQ.....
0x0030: 8F F4 C7 13 00 00 74 4D 48 50 49 76 35 45 44 32 .....tMHPivSED2
0x0040: 38 71 35 42 61 42 6A 35 2F 5A 4C 41 41 41 41 41 8q5BaBj5/ZLAAAA
0x0050: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAA
+++++
```

Handler to agent communication using ICMP.

The time to live field (TTL) is zero which makes it impossible for the datagram to leave the network boundary. Routers decrement the datagram's TTL field by one every time they forward the datagram. If a router encounters a datagram with a TTL of zero they simply drop the datagram. This ensures that datagrams eventually expire if they do not reach their intended destination. The source IP address is spoofed (197.229.68.0).

```
06/24-12:48:11.602688 197.229.68.0 -> 192.168.0.2
ICMP TTL:200 TOS:0x0 ID:61108 IpLen:20 DgmLen:70
Type:0 Code:0 ID:47997 Seq:9302 ECHO REPLY
0x0000: 00 50 DA CA 55 87 00 50 DA CA 55 8E 08 00 45 00 .P..U..P..U...E.
0x0010: 00 46 EE B4 00 00 C8 01 39 72 C5 E5 44 00 C0 A8 .F.....9r...D...
0x0020: 00 02 00 00 5F 4D BB 7D 24 56 74 4D 48 50 49 76 ....M.}SVtMHPiv
0x0030: 35 45 44 32 38 71 35 42 61 42 6A 35 2F 5A 4C 41 SED28q5BaBj5 ZLA
0x0040: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAA
0x0050: 41 41 41 41
AAAA
+++++
```

Handler to agent communication using UDP.

The source IP address is spoofed (197.229.68.0).

```
06/24-12:48:11.642688 197.229.68.0:63343 -> 192.168.0.2:61145
UDP TTL:246 TOS:0x0 ID:34455 IpLen:20 DgmLen:70
Len: 53
```

```
0x0000: 00 50 DA CA 55 87 00 50 DA CA 55 8E 08 00 45 00 .P..U..P..U...E.
0x0010: 00 46 86 97 00 00 F6 11 73 7F C5 E5 44 00 C0 A8 .F.....s...D...
0x0020: 00 02 F7 6F EE D9 00 35 58 D7 74 4D 48 50 49 76 ...o...5X.tMHPiv
0x0030: 35 45 44 32 38 71 35 42 61 42 6A 35 2F 5A 4C 41 5ED28q5BaBj5.2LA
0x0040: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
0x0050: 41 41 41 41
AAAAAAAAAAAA
AAAA
```

```
+++++
+++++
```

4. TFN2K Attacks

The following datagram captures are samples of datagrams produced by TFN2K attacks. It is interesting to note these attacks also contain some of the same characteristics found in the TFN2K communications traffic (i.e. TTL zero, TcpLen zero, and spoofed source IP addresses).

ICMP flood attack.

The TTL is zero and the source IP address is spoofed (134.126.182.0).

```
+++++
06/24-12:48:11.642688 134.126.182.0 -> 192.168.0.1
ICMP TTL:0 TOS:0x0 ID:11507 IpLen:20 DgmLen:92
Type:8 Code:0 ID:0 Seq:0 ECHO
0x0000: 00 50 DA CA 55 7B 00 50 DA CA 55 87 08 00 45 00 .P..U(P..U...E.
0x0010: 00 5C 2C F3 00 00 00 01 90 2F 86 7E B6 00 C0 A8 .\.....t....
0x0020: 00 01 08 00 F7 FF 00 00 00 00 00 00 00 00 00 00 .X.....
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x0040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x0060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
+++++
```

SYN flood attack.

The TcpLen is zero and the source IP address is spoofed (246.52.240.0).

```
+++++
06/24-12:43:47.202688 246.52.240.0:57438 -> 192.168.0.1:24303
TCP TTL:245 TOS:0x0 ID:6592 IpLen:20 DgmLen:40
**U**S* Seq: 0x9CF871 Ack: 0xDD3A0000 Win: 0x267F TcpLen: 0 UrgPtr: 0x2EAC
0x0000: 00 50 DA CA 55 7B 00 50 DA CA 55 87 08 00 45 00 .P..U(P..U...E.
0x0010: 00 28 19 C0 00 00 F5 06 04 DA F6 34 F0 00 C0 A8 .....4....
0x0020: 00 01 E0 5E 5E EF 00 9C F8 71 DD 3A 00 00 00 22 .X.....q.....
0x0030: 26 7F 92 1D 2E AC
&.....
+++++
```

UDP flood attack.

The source IP address is spoofed (103.197.130.0).

```

=====
06/24-12:46:53.702688 103.197.130.0:37406 -> 192.168.0.1:28130
UDP TTL:243 TOS:0x0 ID:45801 IpLen:20 DgmLen:29
Len: 9
0x0000: 00 50 DA CA 55 7B 00 50 DA CA 55 87 08 00 45 00 .P..U(.P..U...E.
0x0010: 00 1D B2 E9 00 00 F3 11 6A 20 67 C5 82 00 C0 A8 .....j g.....
0x0020: 00 01 92 1E 6D E2 00 09 FF F5 00 .X..m.....
=====

```

TARGA attack.

The Targa attack employs a number of DoS attacks as described in Table 2: Evolution of DoS Attacks. Samples of the associated Targa network attack traffic are listed below.

```

=====
01/25-15:28:09.186440 153.245.138.0:31570 -> 192.168.0.1:36895
TCP TTL:0 TOS:0x0 ID:17014 IpLen:20 DgmLen:266
1*U*PR** Seq: 0x66CF7580 Ack: 0x12198F7C Win: 0xE3B3 TcpLen: 32 UrgPtr: 0x914
TCP Options (1) => Opt 111 (40): EDDD A9ED 85B4 AA3E 2BFD BDA9 0F89 AA45 E68E
A699 B477 AC5F E4A2 661A E11A FC7F DB2B 044A 5390 6EAF
BD A9 0F 89 AA 45 E6 8E A6 99 B4 77 AC 5F E4 A2 .....E.....w...
66 1A E1 1A FC 7F DB 2B 04 4A 53 90 6E AF 6E 86 f.....+.JS.n.n.
6A E8 B9 16 AE 31 36 94 67 09 71 ED 75 CC 69 33 j....16.g.q.u.i3
39 AE D4 A4 16 D6 BC 0E 5C 1E 05 0C EB 20 66 18 9.....\.... f.
86 07 9A 7D 08 0F 2E 87 6D FC FD 0E 2D 0B 7F 1B ...j.....m.....
F7 24 2C 70 6C ED 7D AC 5E 69 61 0A EB B5 35 9A .S.pl.).ia...5.
77 6E AD D2 C5 35 E3 A6 0C 78 8B DB BA 16 9A 9D wn...5...x.....
B0 76 28 0E 6A 18 2E 91 9B BA F4 BD BB 58 05 6F .v(.j.....X.o
1E 56 92 42 FF 9A 5D E4 38 56 9E 95 86 7B B7 88 .V.B...8V...(..
D6 56 48 39 0F EF 51 94 E2 28 F5 A1 1A 75 0E 0C .VH9..Q...u...
E6 94 41 87 CF 2C 82 FA A1 7B C9 53 08 53 AA 6D ..A.....S.S.m
7A 95 9F 38 D3 14 EF 55 8F BF 2B FB 6D 8E 0C 40 z...8...U...m...@
AC 43 F0 9F 31 8C C4 2B A7 05 BE 3B 52 B7 AC E0 .C...l...+...R...
C8 1E 33 DC DF C7 ...3...
=====

```

```

=====
01/25-15:28:09.186440 158.203.233.0 -> 192.168.0.1
IPv6 TTL:0 TOS:0x0 ID:54675 IpLen:20 DgmLen:235
Frag Offset: 0x4 Frag Size: 0xd7
DC 22 34 B5 EC BE 9B 2B BC 51 39 AC 24 A2 65 BA .4....+.Q9.S.e.
A8 EB 64 FD C4 02 38 0B 27 B6 26 80 52 E6 E9 14 ..d...8...&.R...
F5 1D 45 6D EA 3C 06 AE A9 2F C8 47 9D 95 1C 38 ..Em.<...G...8
33 38 0F 3B AA 50 D4 40 52 88 B8 EA 16 06 3A 68 38...F.@R.....th
96 B8 AD E1 68 E1 02 7D B3 19 5A AA 5D E4 90 C7 ....h...Z....
A3 56 AF 78 85 28 9C B5 56 64 CA 3B 07 5D 4A 28 .V.x...Vd...J
3B D0 CF E4 08 F1 7F 74 7B 1B 76 57 0E 7D 4E 04 .....t(.vw.)N.
42 27 42 BD 91 F9 5D 2E DB 35 EA 5E 28 BB BA D8 B'B...].5..
20 0D EC F2 14 8F C0 FD F2 F2 3E 1F 38 3E 0F 06 .....>.8...
0C 08 7C 64 3B 73 9F DD E2 DA A6 3B BD BC DC BA ..|.ts.....
70 F4 1C 12 57 41 D6 87 1C EE 26 7E 48 74 03 1C p...WA....&-Hc...
6F 49 68 CF 64 FA 46 A8 75 95 CB 45 80 4C 60 07 cIn.d.F.u...E.L..
41 4E 9B D7 58 AD 81 E9 64 F0 E2 35 72 16 1F BA AN..X...d...Sr...
14 68 4B F5 1A 26 D2 . hK...&.
=====

```

```
=====
01/25-15:28:09.176440 139.231.71.0 -> 192.168.0.1
IGMP TTL:0 TOS:0x0 ID:28678 IpLen:20 DgmLen:215
Frag Offset: 0x6 Frag Size: 0xC3
D2 5B 80 44 59 9F 64 DB E5 1A 3B AF 58 D6 82 DB .{.DY.d....X...
E7 49 37 26 63 01 38 17 5B A8 6D 17 37 2C B3 64 .I7&c.8.[.m.7..d
EA 04 AD A4 31 6C F7 EC 40 32 D0 C1 DA 3F F6 93 ....1l..@2...?..
C5 AC DB 55 44 7A C9 E2 C8 5A B1 57 5B 3D C5 30 ...UDz...Z.W[=..0
18 D5 A0 24 66 6A 9B 07 E8 22 00 D8 A0 E5 0A EC ...Sfj....".....
8C 59 AA 40 59 C1 D3 0A 95 C2 D0 4D 8E 36 CF E9 .Y.@Y.....M.6..
0E 21 0B DB 84 B2 85 12 A9 5F 32 E5 A3 5B 85 18 .!.....2...{..
11 6A 85 71 B3 68 BF 02 8B 6E ED AE 0F B3 46 93 .j.q.h....n....F.
C4 C8 D8 FA DC 59 AE 0A 92 7D 25 7A 53 94 72 85 .....Y....)%zS.r.
0C 20 3E 21 B1 95 FD 9B 20 49 30 5D 41 9D 97 CB .>!.....IO)A...
A6 D0 3D 34 AF 9D 57 15 61 82 B2 3E 42 AE 11 8F ..=4..W.a..>B...
4C 5C 8E 34 AE 9F E2 1F A7 27 46 A3 AE EB F3 6C L\..4.....F.....
A2 C6 96 ...
=====q
```

```
=====
01/27-15:51:57.516440 235.95.147.0 -> 192.168.0.1
XNS-IDP TTL:0 TOS:0x0 ID:39331 IpLen:20 DgmLen:510
Frag Offset: 0x1586 Frag Size: 0x1EA
3A FA AE 13 31 98 E5 50 F9 AD 0D 06 53 AD C8 58 :...l...P....S...X
2E A0 EB 94 24 D0 F6 88 6A 0B E7 AD 56 FB AD 95 ....S...j....V...
1B 56 65 88 30 7F DD 95 FA 6B 7A D2 26 81 45 7B .Ve.0....kz.&.E{
61 7D 9B B0 BC D0 B1 C0 5B 7D AE 68 48 62 A1 54 a).....{).hHb.T
C8 96 41 FA BE BE 93 7B 5A CE 59 8D 94 9A 88 2C ..A....{Z.Y.....
78 AF 78 5B A7 8B 35 F3 71 DC EB 2F F3 AB B6 0C x.x[...5.q.....
4A B2 F5 28 75 72 EE BC 62 4D 05 A8 9A B4 2F F6 J..(ur...bM.....
CF 72 0E 7F 93 C6 60 AF 46 39 C5 35 00 19 19 E2 .r.....F9.5....
12 24 DE 53 40 53 17 63 29 FB 19 BB 9B B6 61 B5 .S.SeS.c).....a.
96 97 B4 D2 75 5E CE 79 BA 49 79 93 95 F3 38 B5 ....u<.y.Iy...8.
42 D4 F9 A4 B6 71 8E 11 E3 4D 44 98 8F 74 45 7B B....q...MD..tE{
D3 38 84 D0 95 65 EC B2 DB 74 95 B9 D1 2A B3 B4 .8....e...t.....
A8 17 0E 99 BD D8 AF B5 7B 3C B0 D4 D0 CF 62 EC .....(<...b.
AE 72 6A 5F 86 DD F7 AA 47 03 32 8C 39 68 70 ED .rj_....G.2.9hp.
DE A3 88 E6 9C B6 11 7C DE 11 14 6D A2 46 9A 25 .....|...m.F.3
B5 8C 03 63 C8 99 07 EA 0E 88 16 C7 1D C7 AE F5 ...c.....
4C 77 C8 CE 74 C0 96 B7 9E 0D E6 C3 4F 5E A2 96 Lw..t.....O...
00 DE B8 1C 89 91 0F 45 3B 36 A2 F9 F6 DB 15 50 .....E:6.....P
AC 78 CE 7B AC 8C CA B9 7E BB BE AE 50 BD F8 5A .x.(.....P...2
1B 4A 46 8F 06 69 BA 8C B8 4A A9 19 21 E1 A3 15 .JF..i...J...!...
65 81 75 3C 16 0C E9 66 2F A2 67 00 34 08 71 C9 e.u<...f.g.4.q.
A4 B3 E6 C9 32 13 4E D1 2A 9C A4 06 02 04 37 95 ....2.N.*.....7.
32 7F E4 63 F4 E7 59 43 89 0E AC CA 2D A3 A7 17 2..c..YC.....-...
81 39 CE F0 63 36 24 7F 3A 0B D7 43 33 58 AE FD .9...c6S...C3X..
06 DE 89 B4 7C A9 79 E7 E7 F4 4A 49 92 8E F9 35 ....|y...JI...5
D8 86 08 D2 63 4B 8D 80 B5 C0 77 66 83 A1 9B 43 ....cK....wf...C
34 92 34 7A AF 2B 8E 1E 16 A2 6A 10 AB 4F CB 87 4.4z.+....j...0..
6F F9 D0 2B 9C 3D 3E 7C 00 EA 78 EB 62 13 76 B6 c...+...=>|...x.b.v.
14 D0 65 91 EA 65 AB EF 15 E2 AE 5F E2 AB A1 CF ...e...e....._....
B5 AB F3 02 F1 D8 64 C0 31 EA 03 52 CB F6 D0 AF .....d.l..P....
CA 81 6A 96 1A A3 DD 59 AC F3 ..j.....Y...
=====
```

```

=====
01/27-15:51:57.586440 189.99.238.0 -> 192.168.0.1
PUP TTL:0 TOS:0x0 ID:19239 IpLen:20 DgmLen:165
BE 25 CE D1 39 87 31 56 97 B1 6E EE 83 60 A1 56 .8..9.1V..n..V
56 2F E3 C9 47 E0 CF 5E 39 A6 35 9D 25 0E 2B FA V/..G..^9.5..8..
BD 60 48 56 73 01 DD 68 30 7D 80 B8 A8 D5 F0 95 .HV..h0}.....
4D 0F 66 2C 1D A3 49 39 80 5A 06 DE 8E 4A A7 08 M.f...I9.Z...J..
30 63 A7 26 10 7D 9E 2C 3D 03 88 3F E1 F2 C8 3A 0c.8.)..=..?....
85 35 D7 CE DF CB 80 5F 43 03 7B D5 B9 AB E1 B0 .S.....C.(.....
7F E4 53 2E 53 29 CA BC 1F CC 40 12 C5 02 81 4C ..S.S)....@....L
45 3B CC 9C 43 A9 D8 CF F9 71 5B 8D 07 1C F3 16 E:..C....q[.....
2A 5D CB 23 AA 18 9D B4 F2 46 AE 2F D7 AD F1 E0 *|.#####F./....
FO .
=====

```

MIXflood attack.

The Mixflood attack sends a random combination of the SYN, UDP, and ICMP flood attacks.

Mixflood attack – UDP flood.

```

=====
01/27-15:55:31.596440 106.235.78.0:65409 -> 192.168.0.1:127
UDP TTL:248 TOS:0x0 ID:2071 IpLen:20 DgmLen:29
Len: 9
00 .
=====

```

Mixflood attack – ICMP flood.

```

=====
01/27-15:55:31.596440 6.65.63.0 -> 192.168.1.15
ICMP TTL:0 TOS:0x0 ID:11914 IpLen:20 DgmLen:92
Type:8 Code:0 ID:0 Seq:0 ECHO
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
=====

```

Mixflood attack – TCP flood.

```

=====
01/27-15:55:31.596440 118.120.233.0:16344 -> 192.168.0.1:54502
TCP TTL:201 TOS:0x0 ID:48952 IpLen:20 DgmLen:40
**U***S* Seq: 0x59546B Ack: 0x29C2000C Win: 0x44F2 TcpLen: 0 UrgPtr: 0x49C9
=====

```

Appendix B Plague

1. Plague Screen Shot

```
plague> help
quit          -      Closes your connection
help          -      Prints out this shit
stream        -      <port> <ip of target> <time>
syn           -      <port> <ip of target> <time>
bindshell     -      <ip of miniserver> <port>
pingall       -      Checks if ghosts are up
```

2. Plague Network Data Captures

Snort was used to provide the network trace of the attack. Some snippets of the actual trace were deleted because of repetition; interesting traces were kept for explanation purposes. Some portions of the packet captures are also bolded to draw attention to them.

3. Plague Communications

Pingall.

Pingall is a command that the handler uses to determine what agents are “alive” and awaiting attack commands. After the handler issues the Pingall command, all available agents respond to the handler. The handler sends the agent the login string (**ld.so.1**) and the agent listening on the default port (6969) sends the response PONG to the handler. No source IP address spoofing is used during handler to agent communications.

```
11/17-15:36:44.411408 192.168.0.7:1098 -> 192.168.0.2:6969
TCP TTL:64 TOS:0x0 ID:1032 DF
***AP*** Seq: 0xA602CD6D Ack: 0xA5494369 Win: 0x7D78
TCP Options => NOP NOP TS: 43332231 17092837

0x0000: 00 50 DA CA 44 F0 00 01 02 37 C8 2E 08 00 45 00 .P..D....7....E.
0x0010: 00 43 04 08 40 00 40 06 E3 34 C0 A8 00 07 C0 A8 .C..@.@..4.....
0x0020: 00 02 04 4A 1B 39 A6 02 CD 6D A5 49 43 69 80 18 ...C.9...m.ICi..
0x0030: 7D 78 E3 81 00 00 01 01 08 0A 02 95 3D 87 01 04 }x.....2....
0x0040: D0 E5 44 20 6C 64 2E 73 6F 2E 31 20 30 20 30 20 ..D ld.so.1 0 0
0x0050: 30
```

```

=====
11/17-15:36:44.411720 192.168.0.2:6969 -> 192.168.0.7:1098
TCP TTL:64 TOS:0x0 ID:101 DF
***AP*** Seq: 0xA5494369 Ack: 0xA602CD7C Win: 0x7D78
TCP Options => NOP NOP TS: 17092938 43332231

0x0000: 00 01 02 37 C8 2E 00 50 DA CA 44 F0 08 00 45 00 ...7...P..D...E.
0x0010: 00 39 00 65 40 00 40 06 B6 E1 C0 A8 00 02 C0 A8 ..9.e@.a.....
0x0020: 00 07 1B 39 04 4A A5 49 43 69 A6 02 CD 7C 80 18 ...9.J.iCi....
0x0030: 7D 78 54 08 00 00 01 01 08 0A 01 04 D1 4A 02 95 |xT.....J..
0x0040: 32 87 50 4F 4E 47 00                                2.PONG.

```

4. Plague Attacks

Stream flood attack.

The source IP address is spoofed (129.163.166.121).

```

=====
11/17-15:37:50.462688 129.163.166.121:23070 -> 192.168.0.1:80
TCP TTL:255 TOS:0x8 ID:41007 IpLen:20 DgmLen:40
***** Seq: 0x3606AF6B Ack: 0x0 Win: 0x4000 TcpLen: 20
0x0000: 00 50 DA CA 55 7B 00 50 DA CA 55 87 08 00 45 08 .P..U(.P..U...E.
0x0010: 00 28 A0 2F 00 00 FF 06 32 7B 81 A3 A6 79 C0 A8 .. ..2...Y..
0x0020: 00 01 5A 1E 00 50 38 08 AF 6B 00 00 00 50 00 .XZ..P8...k...P.
0x0030: 40 00 44 E5 00 00                                @.D...
=====

```

SYN flood attack.

The agent responds to the handler that the SYN flood attack is occurring by sending "flooding target" to the handler.

```

=====
11 17-15:37:50.911316 192.168.0.2:6969 -> 192.168.0.7:1103
TCP TTL:64 TOS:0x0 ID:119 DF
***AP*** Seq: 0xA8A3DCF6 Ack: 0xAA2A23CB Win: 0x7D78
TCP Options => NOP NOP TS: 17099588 43338881
0x0000: 00 01 02 37 C8 2E 00 50 DA CA 44 F0 08 00 45 00 ...7...P..D...E.
0x0010: 00 47 00 77 40 00 40 06 B6 C1 C0 A8 00 02 C0 A8 ..G.w@.a.....
0x0020: 00 07 1B 39 04 4F A8 A3 DC F6 AA 2A 23 CB 60 18 ...9.C.....*#...
0x0030: 7D 78 BC 16 00 00 01 01 08 0A 01 04 EB 44 02 95 |x.....D..
0x0040: 4C 81 07 0A 66 6C 6F 6F 64 69 6E 67 2C 74 61 72 L...flooding tar
0x0050: 67 65 74 2C 0A                                get .
=====

```


The datagram capture below is an example of the actual SYN flood traffic. The source IP address is spoofed (57.106.228.45).

```
=====
11/17-15:37:50.918298 57.106.228.45:0 -> 192.168.0.1:0
TCP TTL:40 TOS:0x0 ID:2450
0x0000: 00 50 DA CA 55 83 00 50 DA CA 44 F0 08 00 45 00 .P..U..P..D...E.
0x0010: 00 28 09 92 00 00 28 06 A9 FD 39 6A E4 2D C0 A8 .....9j...
0x0020: 00 01 B4 9D 00 01 B5 D1 C7 63 20 5C 06 6D 00 02 .....C..m..
0x0030: FF FF 12 11 00 00 00 00 00 00 00 00 00 00 .....

0x0000: 00 50 DA CA 55 83 00 50 DA CA 44 F0 08 00 45 00 .P..U..P..D...E.
0x0010: 00 28 09 92 00 00 28 06 A9 FD 39 6A E4 2D C0 A8 .....9j...
0x0020: 01 01 B4 9D 00 01 B5 D1 C7 63 20 5C 06 6D 00 02 .....C..m..
0x0030: FF FF 12 11 00 00 00 00 00 00 00 00 00 00 .....
=====
```

References

- [1] (October 23, 2001). Household Internet Use Survey. Electronic Commerce Release. Electronic Branch, Industry Canada. Retrieved December 12, 2001, from http://ecom.ic.gc.ca/english/documents/research/backgounder_oct23_01.pdf
- [2] (April 3, 2001). Electronic Commerce and Technology 2000. Industry Canada. Retrieved December 12, 2001, from http://e-com.ic.gc.ca/english/documents/ecom_ict_2000.pdf
- [3] Chrétien, J. (January 30th, 2001). Speech from the Throne to Open the First Session of the 37th Parliament of Canada. Retrieved July 23, 2001, from http://www.sft-ddt.gc.ca/sftddt_e.htm
- [4] McCall, T. (March 13, 2001). Worldwide Business-to-Business Internet Commerce to Reach \$8.5 Trillion in 2005. Gartner Group. Retrieved January 04, 2002, from http://www4.gartner.com/5_about/press_room/pr20010313a.html
- [5] Spitzner, L. (July 21, 2000). Know Your Enemy *The Tools and Methodologies of the Script Kiddie*. Retrieved January 04, 2002, from <http://project.honeynet.org/papers/enemy/>
- [6] CERT Coordination Center. CERT/CC. Retrieved January 08, 2002, from <http://www.cert.org/>
- [7] (January 10, 2002). CERT/CC Statistics 1988-2001. CERT/CC. Retrieved January 18, 2002, from www.cert.org/stats/cert_stats.html
- [8] The Honeynet project. Retrieved September 17, 2001, from <http://project.honeynet.org/>
- [9] Frequently Asked Questions. Retrieved December 20, 2001, from <http://project.honeynet.org/faq.html>
- [10] Spitzner, L. (July 22, 2001). Know Your Enemy: Statistics. *Analyzing the past... predicting the future*. Retrieved September 20, 2001, from <http://project.honeynet.org/papers/stats/>

- [11] (February 19, 2000) SDSC's Security Experiment. San Diego Supercomputer Center & NPACI. Retrieved December 2, 2001, from <http://security.sdsc.edu/incidents/worm.2000.01.18.shtml>
- [12] (August 17, 2000). CERT/CC Overview Incident and Vulnerability Trends. CERT/CC. Retrieved August 12, 2001, from <http://www.cert.org/present/cert-overview-trends/>
- [13] (August, 2001). How Many Online? Nua Internet surveys. Retrieved December 2, 2001, from <http://sbinfoCanada.about.com/gi/dynamic/offsite.htm?site=http%3A%2F%2Fwww.nua.ie%2Fsurveys%2F>
- [14] Harrison, A. (February 14, 2000). The denial-of-service aftermath. CNN.com. Retrieved January 2, 2002, from <http://www.cnn.com/2000/TECH/computing/02/14/dos.aftermath.idg/index.html>
- [15] (February 9, 2000). Cyber-attacks batter Web heavyweights. CNN.com. Retrieved December 6, 2001 from <http://www.cnn.com/2000/TECH/computing/02/09/cyber.attacks.01/index.html>
- [16] (February 10, 2000). The Yankee Group predicts over \$1.2 Billion Impact as a Result of Recent Attacks Launched by Internet Hackers. Yankee Group News Release. Retrieved January 18, 2002 from <http://www.yankeegroup.com/webfolder/vg21a.nsf/press/384D3C49772576EF85256881007DC0EE?OpenDocument>
- [17] (May 24, 2001). CERT hit by DDoS attack for a third day. ITworld.com. Retrieved January 18, 2002 from <http://www.itworld.com/Sec/3834/IDG010524CERT2/>
- [18] Fisher, D. (May 23, 2001). CERT Center hit by DDoS attack. ZDNet Tech Update. Retrieved October 24, 2001 from <http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2764475,00.html>
- [19] (September, 1981). RFC: 793. Transmission Control Protocol Darpa Internet Program Protocol Specification. Retrieved January 18, 2002 from <ftp://ftp.isi.edu/in-notes/rfc793.txt>
- [20] (September, 1981). RFC: 791. Internet Protocol Darpa Internet Program Protocol Specification. Retrieved January 18, 2002 from <ftp://ftp.isi.edu/in-notes/rfc791.txt>

- [21] Stevens, R. (October 1999). TCP/IP Illustrated, Volume 1 The Protocols. Addison-Wesley Professional Computing Series.
- [22] Bellovin, S. (April 1989). Security problems in the TCP/IP Protocol Suite, Computer Communications Review, Vol. 19, No. 2, pp. 32-48.
- [23] (January 23, 1995). CIAC advisory notice. F-08: Internet Address Spoofing and Hijacked Session Attacks. CIAC. Retrieved November 01, 2001 from <http://ciac.llnl.gov/ciac/bulletins/f-08.shtml>
- [24] (September 23, 1995). CERT[®] Advisory CA-1995-01. IP Spoofing Attacks and Hijacked Terminal Connections. CERT/CC. Retrieved August 23, 2001 from <http://www.cert.org/advisories/CA-1995-01.html>
- [25] (September 19, 1996). CERT[®] Advisory CA-1996-21. TCP SYN Flooding and IP Spoofing Attacks. CERT/CC. Retrieved August 23, 2001 from <http://www.cert.org/advisories/CA-1996-21.html>
- [26] (December 16, 1997). CERT[®] Advisory CA-1997-28. IP Denial-of-Service Attacks. CERT/CC. Retrieved August 23, 2001 from <http://www.cert.org/advisories/CA-1997-28.html>
- [27] (June, 1996). ISO/OSI Network Model. Retrieved January 12, 2002 from http://www.uwsg.indiana.edu/usail/network/nfs/network_layers.html
- [28] Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. (ISO/IEC 8802-3: 2000 (E)). IEEE Standard for Information technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements. Retrieved November 29, 2001 from <http://standards.ieee.org/getieee802/802.3.html>
- [29] IEEE Registration Authority Overview. IEEE.org. Retrieved January 3, 2002 from <http://standards.ieee.org/regauth/index.html>
- [30] Schuba, C. et al. (1997). Analysis of a Denial of Service Attack on TCP. Retrieved October 29, 2001 from <http://citeseer.nj.nec.com/cache/papers/cs/347/http:zSzzSzwww.cl.cam.ac.ukzSzuserszSzmgk25zSzsynkill.pdf/analysis-of-a-denial.pdf>
- [31] Characterizing and Tracing Packet Floods Using Cisco Routers. Cisco Systems, Inc. Retrieved January 28, 2002 from <http://www.cisco.com/warp/public/707/22.html>

- [32] Dunigan, T. (June 10, 2001). Backtracking spoofed packets. Retrieved January 28, 2002 from <http://www.csm.ornl.gov/~dunigan/>
- [33] Burch, H., Cheswick, B. (December 2000). Tracing Anonymous Packets to Their Approximate Source. Retrieved August 12, 2001 from http://www.silicondefense.com/research/itrex/archive/tracing-papers/burch00tracing_approximate_source.pdf
- [34] Flanagan, H. (April 30, 2001). Egress Filtering – Keeping the Internet Safe from Your Systems. SANS Institute. Retrieved January 28, 2002 from <http://rr.sans.org/sysadmin/egress.php>
- [35] (January 1998). RFC: 2267. Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. Retrieved September 27, 2001 from <ftp://ftp.isi.edu/in-notes/rfc2267.txt>
- [36] (October 2, 1997). Denial of Service Attacks. CERT/CC. Retrieved August 12, 2001 from http://www.cert.org/tech_tips/denial_of_service.html
- [37] Moore, D., Voelker, G., Savage, S. (2001). Inferring Internet Denial-of-Service Activity. CAIDA Web Site. Retrieved October 23, 2001 from <http://www.caida.org/outreach/papers/2001/BackScatter/usenixsecurity01.pdf>
- [38] Houle, J., Weaver, G. (October 2001). Trends in Denial of Service Attack Technology. CERT/CC. Retrieved November 3, 2001 from http://www.cert.org/archive/pdf/DoS_trends.pdf
- [39] (July 29, 2001). A Very Real and Present Threat to the Internet: July 31 Deadline for Action. CERT/CC. Retrieved October 15, 2001 from http://www.cert.org/congressional_testimony/CRannounce.html
- [40] (July 17, 2001). eEye Digital Security Advisory AL20010717. .ida "Code Red" Worm. eEye Digital Security. Retrieved January 1, 2002 from <http://www.eeye.com/html/Research/Advisories/AL20010717.html>
- [41] Moore, D. (November 28, 2001). The Spread of the Code-Red Worm (CRv2). CAIDA Web Site. Retrieved December 11, 2001 from http://www.caida.org/analysis/security/code-red/coderedv2_analysis.xml

- [42] Dittrich, D. (July 22, 2000). Distributed Denial of Service Is There Really a Threat? USENIX Security Symposium 2000. Retrieved December 14, 2001 from <http://staff.washington.edu/dittrich/talks/sec2000/>
- [43] Packet Storm.
<http://packetstormsecurity.nl/>
- [44] (November 18, 1999). CERT[®] Incident Note IN-99-07. Distributed Denial of Service Tools. CERT/CC. Retrieved December 12, 2001 from http://www.cert.org/incident_notes/IN-99-07.html
- [45] (July 20, 2001). CERT[®] Advisory CA-2001-20. Continuing Threats to Home Users Thursday. CERT/CC. Retrieved September 15, 2001 from http://www.cert.org/incident_notes/IN-99-07.html
- [46] (December 7, 1999). CERT[®] Coordination Center. Results of the Distributed-Systems Intruder Tools Workshop. CERT/CC. Retrieved May 12, 2001 from http://www.cert.org/reports/dsit_workshop-final.html
- [47] Dittrich, D. Dave Dittrich Home Page. Retrieved September 12, 2001 <http://www.washington.edu/People/dad/>
- [48] Information Security Reading Room. SANS Institute. Retrieved October 23, 2001 from http://www.sans.org/infosecFAQ/threats/threats_list.htm
- [49] (February 23, 2000). Consensus Roadmap for Defeating Distributed Denial of Service Attacks. A Project of the Partnership for Critical Infrastructure Security, Version 1.10. SANS Institute. Retrieved January 18, 2002 from http://www.sans.org/ddos_roadmap.htm
- [50] (March, 23, 2000). Help Defeat Denial of Service Attacks: Step-by-Step, Revision: 1.41. SANS Institute. Retrieved August 29, 2001 from <http://www.sans.org/dosstep/index.htm>
- [51] Dittrich, D. (May 1, 2000). The "mstream" distributed denial of service attack tool. Retrieved May 14, 2001 from <http://staff.washington.edu/dittrich/misc/mstream.analysis.txt>
- [52] Brooke, P. (January 8, 2001). DDoS: Internet Weapons of Mass Destruction. Network Computing. Retrieved July 5, 2001 from <http://www.networkcomputing.com/1201/1201f1c1.html>

- [53] Dittrich, D. (December 31, 1999). The "stacheldraht" distributed denial of service attack tool. Retrieved May 14, 2001 from <http://staff.washington.edu/dittrich/misc/stacheldraht.analysis>
- [54] (September 3, 1998). CERT[®] Advisory CA-98.11. Stack Overflow in ToolTalk RPC Service. CERT/CC. Retrieved June 11, 2001 from <http://www.cert.org/advisories/CA-98.11.tooltalk.html>
- [55] (September 15, 2000). CERT[®] Incident Note IN-2000-10. Widespread Exploitation of rpc.statd and wu-ftpd Vulnerabilities. CERT/CC. Retrieved June 11, 2001 from <http://www.cert.org/advisories/CA-99-05-statd-automountd.html>
- [56] (December 30, 1999). NIPC Alert. TRINOO/Tribal Flood Net/tfn2k. NIPC. Retrieved May 12, 2001 from <http://www.nipc.gov/warnings/alerts/1999/trinoo.htm>
- [57] (November 18, 1999). CERT[®] Incident Note IN-99-07. Distributed Denial of Service Tools. CERT/CC. Retrieved June 12, 2001 from http://www.cert.org/incident_notes/IN-99-07.html
- [58] (January 5, 1998). CERT[®] Advisory CA-1998-01. Smurf IP Denial-of-Service Attacks. CERT/CC. Retrieved July 17, 2001 from <http://www.cert.org/advisories/CA-1998-01.html>
- [59] (February 8, 1996). CERT[®] Advisory CA-1996-01. UDP Port Denial-of-Service Attack. CERT/CC. Retrieved July 17, 2001 from <http://www.cert.org/advisories/CA-1996-01.html>
- [60] Savage, S. (July 2001). Denial of Service Attacks: Why We Should Still Worry. Asta Networks. Retrieved August 12, 2001 from <http://www.astanetworks.com/resources/materials/TISC2001.pdf>
- [61] (September 01, 1997). LOKI ICMP tunneling backdoor. ISS X-Force Database. Retrieved June 15, 2001 from <http://xforce.iss.net/static/1452.php>
- [62] (April 7, 1997) Howard, J. An Analysis of Security Incidents on the Internet 1989-1995, pp.5. CERT/CC. Retrieved August 4, 2001 from <http://www.cert.org/research/JHThesis/Chapter1.html>
- [63] Internet Security: Network Attacks. University of Minnesota. Retrieved September 26, 2001 from http://www1.umn.edu/oit/newsletter/01/0801_itn/security.html

- [64] Dekker, M. (1997) Security of the Internet. The Froehlich/Kent Encyclopedia of Telecommunications vol. 15. New York. pp. 231-255. Retrieved November 16, 2001 from www.cert.org/encyc_article/tocencyc.html
- [65] (January 18, 2001). CERT® Incident Note IN-2001-01. Widespread Compromises via "ramen" Toolkit. CERT/CC. Retrieved July 12, 2001 from http://www.cert.org/incident_notes/IN-2001-01.html
- [66] Sam Spade is available at:
<http://www.samspade.org>
- [67] (October 1994). RFC: 1700 Assigned Numbers. Retrieved January 2, 2002 from <ftp://ftp.isi.edu/in-notes/rfc1700.txt>
- [68] Firewalk is available at:
<http://www.packetfactory.net/Projects/Firewalk>
- [69] Arkin, O. (June 2001). ICMP Usage in Scanning. Sys-Security Group. Retrieved July 2001. from http://www.sys-security.com/archive/papers/ICMP_Scanning_v3.0.pdf
- [70] Nmap is available at:
<http://www.nmap.org/nmap/index.html#intro>
- [71] Netcat is available at:
<http://www.atstake.com/research/tools/index.html>
- [72] Hawkins, S. (January 13, 2001). Understanding the Attackers Toolkit. SANS Institute. Retrieved November 19, 2001 from <http://rr.sans.org/linux/toolkit.php>
- [73] Miller, T. Analysis of the T0rn rootkit. SANS GIAC. Retrieved November 12, 2001 <http://www.sans.org/y2k/t0rn.htm>
- [74] Smith, C. (November 12, 2000). Covert Shells. SANS Institute. Retrieved December 12, 2001 from http://www.sans.org/infosecFAQ/covertchannels/covert_shells.htm
- [75] Tcpdump is available at:
<http://www.tcpdump.org>
- [76] TFN2K and Plague are available at:
<http://packetstorm.widexs.nl/distributed/>

- [77] Barlow, J. (August 3, 2000). Axent releases a full TFN2K Analysis. SecuriTeam.com. Retrieved November 11, 2001 from <http://www.securiteam.com/securitynews/5YP0G000FS.html>
- [78] (June 1999). RFC: 2612 The CAST-256 Encryption Algorithm. Retrieved January 2, 2002 from <ftp://ftp.isi.edu/in-notes/rfc2612.txt>
- [79] Northcutt, S. (1999). Network Intrusion Detection An Analyst's handbook. New Riders Professional Library. pp.17.
- [80] Snort is available at:
<http://www.snort.org>
- [81] Roesch, M. Snort - Lightweight Intrusion Detection for Networks. Securityfocus.com. Retrieved December 12, 2001 from <http://downloads.securityfocus.com/library/lisapaper.txt>
- [82] Roesch, M. Snort Users Manual Snort Release: 1.83. Retrieved January 30, 2002 from http://www.snort.org/docs/writing_rules/
- [83] (1999). Threat and Risk Assessment Working Guide. Communications Security Establishment. Retrieved June 12, 2001 from http://www.cse.dnd.ca/en/documents/knowledge_centre/publications/manuals/ITSG-04e.pdf
- [84] (November 1994). Security Information Publication 5. Guide to Threat and Risk Assessment for Information Technology. RCMP. Retrieved June 12, 2001 from http://www.rcmp-grc.gc.ca/tsh/pubs/reports/tra_guide.pdf
- [85] Ellison, R., Fisher, D., Linger, R., Lipson, H., Longstaff, T., Mead, N. (1998). CERT® Coordination Center. Survivability: Protecting Your Critical Systems. Retrieved January 1, 2002 from <http://www.cert.org/archive/html/protect-critical-systems.html>
- [86] Fisher, D., Lipson, H. (March 2000). Survivability – A new Technical and Business Perspective on Security. CERT/CC. Retrieved January 12, 2002 from http://interactive.sei.cmu.edu/Columns/Security_Matters/2000/March/Security.mar00.pdf
- [87] Ellison, R., Fisher, D., Linger, R., Lipson, H., Longstaff, T., Mead, N. (November 1997). Survivable Network Systems: An emerging Discipline. CERT/CC. Retrieved August 12, 2001 from <http://www.cert.org/research/97tr013.pdf>

- [88] MacLeod, D., Whyte, D. (July 2001). The role of network layered network security in achieving survivability. International Conference on Dependable Systems and Networks, IEEE Computer Society. Proceedings available at <http://www.ce.chalmers.se/conf2001/dsn/isswprogram.html>
- [89] Ellison, R., Linger, R., Longstaff, T., Mead, N. (August 1999). Survivable Network System Analysis: A Case Study. CERT/CC. Retrieved August 12, 2001 from <http://www.cert.org/archive/pdf/network-analysis.pdf>
- [90] MacLeod, D., Whyte, D. (September 2000). Towards System Survivability using the Single Virtual Enterprise Model and Layered Security through Information Protection Coordination Centres. CERT/CC. Retrieved September 12, 2001 from <http://www.cert.org/research/isw/isw2000/papers/55.pdf>
- [91] MacLeod, D., Whyte, D. (September 2001) Layered Network Security Design Considerations: When Layers Collide. CERT/CC. Retrieved January 3, 2002 from <http://www.cert.org/research/isw/isw2001/papers/Whyte-31-08.pdf>
- [92] (April 28, 2000). Help Defeat Denial of Service Attacks Step-by-Step. Mitre. Retrieved September 12, 2001 from <http://www.mitre.org/research/cyber/DDOS/>
- [93] (February 17, 2000). Strategies to Protect Against Distributed Denial of Service (DDoS) Attacks. Cisco Systems, Inc. Retrieved July 7, 2001 from <http://www.cisco.com/warp/public/707/newsflash.html>
- [94] (November 21, 2001). Router Security Configuration Guide. Router Security Guidance Activity of the System and Network Attack Center (SNAC). Version 1.0j. NSA. Retrieved December 3, 2001 from <http://nsa2.www.conxion.com/cisco/guides/cis-2.pdf>
- [95] Improving Security on Cisco Routers. Cisco Systems, Inc. Retrieved July 28, 2001 from <http://www.cisco.com/warp/public/707/21.html>
- [96] Zwicky, E., Cooper, S., Chapman, B. (2000). Building Internet Firewalls, 2nd edition. O'Reilly & Associates, Inc.
- [97] Ipchains is available at:
<http://www.gnumonks.org/ftp/pub/mirror/linux/ipchains/download/>
- [98] TCP wrapper is available at:
<ftp://ftp.porcupine.org/pub/security/>

- [99] SSH is available at:
<http://www.ssh.com/>
- [100] Lawson, C. (April 9, 2001). Hardening in the Enterprise: Always an After Thought? SANS Institute. Retrieved September 23, 2001 from
http://www.sans.org/infosecFAQ/sysadmin/hard_enterprise.htm
- [101] Bastille Linux is available at:
<http://www.bastille-linux.org/>
- [102] YASSP is available at:
<http://www.yassp.org/>
- [103] Logging Messages with the Syslog Service. Cisco Systems, Inc. Retrieved January 12, 2002 from
<http://www.cisco.com/univercd/cc/td/doc/product/iaabu/cddm/cddm111/adguide/syslog.htm>
- [104] Ching, A. (December 2000). Understanding the Network and System Configuration. Microsoft. Retrieved December 28, 2001 from
<http://msdn.microsoft.com/library/en-us/dnduwon/html/d5configover.asp?frame=true>
- [105] Restuccia, M. (November 21, 2000). Firewall Load Balancers. SANS Institute. Retrieved July 12, 2001 from http://rr.sans.org/firewall/load_balancers.php
- [106] Bjerke, J. (March 4, 2001). SANS Institute. The New Firewall Design Question. Retrieved June 12, 2001 from http://rr.sans.org/firewall/new_design.php