Tuomo Penttinen

Distributed Denial-of-Service Attacks in the Internet

Master's Thesis in
Computer Science and
Information Systems
20.12.2005

University of Jyväskylä
Department of Computer Science and Information Systems
Jyväskylä

ABSTRACT

Penttinen, Tuomo Sakari
Distributed Denial-of-Service Attacks in the Internet / Tuomo Penttinen
Jyväskylä: University of Jyväskylä, 2005.
148 pages
Master's thesis

The purpose of this research is to discuss extensively the various aspects of distributed denial-of-service (DDoS) attacks in the Internet by performing a comprehensive literature review and developing the current knowledge further mostly in a form of a novel classification of the DDoS attack mechanisms. DDoS attacks have caused major problems in the modern Internet since the late nineteen-nineties and yet no appropriate defenses have been developed. One explanation might be that the fundamentals of DDoS attacks are still not understood by many. This study attempts to make a comprehensive coverage of the realm of DDoS attacks and hence assist in defining a basis for understanding the DDoS attacks. The study begins by discussing the terminology in length and then advances into discussing the theory behind DDoS attacks. In this study additional emphasis was put into the discussion of the definitions of the key concepts, which were mostly either specifically created for the purposes of this study or redefined based on the current knowledge. The vast majority of the theoretical section is devoted into describing the DDoS attack mechanisms in detail. The main contribution of this study is the novel classification of the DDoS attack mechanisms, which will be presented in an attempt to depict the core features of the operations of the DDoS attacks. In addition, the study views a few prospects how the DDoS attacks may evolve in the future. One result of the study also indicates that DDoS attacks will continue to persist until the Internet infrastructure is significantly altered.

KEYWORDS: denial-of-service, distributed denial-of-service attacks, classification, concept definitions, automated intrusion agent, development, computer security, Internet.

TABLE OF CONTENTS

1INTRODUCTION 8
1.1 Research objectives 11
1.2 The organization of the thesis 14

2THE FIELD OF DISTRIBUTED DENIAL OF SERVICE 16
2.1 Key concept and term definitions 16
2.1.1 The Internet 17
2.1.2 Client, server / service provider, service 17
2.1.3 Denial-of-Service, Denial-of-Service attack 18
2.1.4 Distributed Denial-of-Service attack 21
2.1.5 Distributed Denial-of-Service network 22
2.1.6 Term definitions 23
2.2 A glance to the past of DoS and DDoS attacks 26
2.3 DoS attacks and the related problem magnitude 30
2.3.1 Information security and DoS 31
2.3.2 The problem magnitude and the threat of DoS attacks 31
2.4 Issues relating to law and liability 34
2.4.1 Legislation and cybercrime 35
2.4.2 Contributory negligence and downstream liability 36

3THE ROOTS OF THE CLASSIFICATION OF DDOS ATTACKS 39
3.1 DDoS as a phenomenon 39
3.1.1 The theoretical impossibility of distinguishing malicious traffic 39
3.1.2 The necessity of core network changes to deal DDoS properly 40
3.1.3 The main principles of the Internet 42
3.2 Classifying DDoS attack mechanisms 43
3.2.1 The root of the classification 45
3.2.2 The second level divisions of the classification 46

4DENIAL-OF-SERVICE ATTACK MECHANISMS 48
4.1 Attacks that target software 48
4.1.1 Local attacks targeting software 49
4.1.2 Remote attacks targeting software 50
4.1.3 Software attacks in contemporary DDoS attack tools 53
4.1.4 Defences against attacks that target software 53
4.2 Attacks that target protocols 56
4.2.1 Protocol attacks in contemporary DDoS attack tools 58
4.2.2 Defences against protocol attacks 58
4.3 Attacks that target bandwidth 59
4.3.1 Traffic validity and attack traffic route 61
4.3.2 Traffic generation in contemporary DDoS attack tools 63
4.3.3 Defenses against bandwidth consumption attacks 66
4.4 Possible evolutions 67
4.5 Summary 69

5DISTRIBUTED DENIAL-OF-SERVICE NETWORK MECHANISMS 71
5.1 Model selection 71
5.1.1 Agent-handler model 72
5.1.2 IRC-based model 75
5.1.3 Scattered model 80
5.1.4 Peer-to-peer model 84
5.1.5 Summary 88
5.2 Creation 89

5.2.1 Ownership of hosts comprising DDoS networks....89
5.2.2 Manual versus automated DDoS network creation....90
5.2.3 Gathering of hosts....92
5.2.4 Deciding host roles....93
5.2.5 Preparation of hosts....94
5.3 Coordination....94
5.3.1 Direct and instant communication....96
5.3.2 Direct, instant and stealth communication....98
5.3.4 Indirect, instant and stealth communication....100
5.3.3 Indirect, delayed and stored communication....102
5.3.5 Public mediums....103
5.3.6 Static instructions....105
5.3.7 Summary....105
5.4 Additional functionality....106
5.4.1 Update mechanisms....107
5.4.2 Stealth mechanisms....108
5.5 Possible evolutions....109
5.5.1 Derivatives of agent-handler model....109
5.5.2 Enhancements to IRC-based model....110
5.5.3 Advanced agents and agent networks....111

6OVERVIEW OF CURRENT COUNTERMEASURES AGAINST DDOS ATTACKS....115
6.1 Issues of attack traffic....116
6.2 Preventive countermeasures....118
6.3 Reactive countermeasures....120
6.4 Post-active countermeasures....123
6.5 Summary....124

7SUMMARY....126

REFERENCES....131

LIST OF FIGURES

FIGURE 1. DDoS attack mechanism classification.....44
FIGURE 2. Agent-handler model (taken from Spech and Lee 2003, 2).....72
FIGURE 3. IRC-based model (taken from Spech and Lee 2003, 3).....76
FIGURE 4. Scattered model.....81
FIGURE 5. Pure p2p model.....85
FIGURE 6. Hybrid p2p model.....86

LIST OF TABLES

TABLE 1. The Main Attributes of the DoS Attack Mechanisms..... 70

TABLE 2. The Requirements, Phases and Specialities of the DoS Attack Mechanisms..... 70

TABLE 3. The Advantages and Disadvantages of the DDoS Network Models..... 88

TABLE 4. The Advantages and Disadvantages of Different Communication Mechanisms..... 106

TABLE 5. The Stages of Countermeasures Against DDoS Attacks..... 125

APPENDIX

APPENDIX 1. CONCEPTS CREATED OR MODIFIED IN THIS STUDY..... 148

1 INTRODUCTION

Since the novel ideas of packet switching networks in the middle of 1960's (Davies 1982) and the first proof-of-concept packet switching network, the ARPANET in 1969 (Roberts 1986, 2-7), computer networks have become highly important components of contemporary societies. At present, computer networks are already used almost everywhere imaginable and the trend is not likely to change in the future.

The ARPANET, now commonly known as the Internet, is one such network and undoubtedly the most successful, the most used and the most known worldwide. The Internet's best-effort and end-to-end design principles (Blumenthal and Clark 2001, 1-2) along with the infamous TCP/IP protocol suite (Naugle 2001) are major factors in the Internet's triumphant success, but also in its inherent security problems. Albeit the Internet has been proven extremely robust in cases of random failure, it has also been proven extremely sensitive to specifically targeted malicious attacks (Albert et al. 2000). This is mostly because the Internet was not designed to be used in such a way it is being used today (Blumenthal and Clark 2001; Clark et al. 2002), which lead to the poor security design. For instance, already in the late eighties Bellovin (1989) pointed out several security problems with the TCP/IP protocol suite.

One of the Internet's largest security concerns is its intrinsic inability to deal with certain denial-of-service (DoS) type of attacks (Houle and Weaver 2001, 1-2). DoS is an established term referring to a situation, where a legitimate requestor of service, or in other words, a client, cannot receive the requested service for one reason or the other (Howard 1997). Instead, DoS attacks are

characterized by the attacker's *primary intent* to cause DoS to the requestors of the service in question (Howard 1997). This subtle difference between the meaning of DoS and a DoS attack is important to notice.

There is a rather large number of ways achieving DoS, as the primary intent to cause DoS is the only requirement for an attack to be classified as a DoS attack. DoS attacks can very well be launched both locally and remotely and they range from software exploits to bandwidth consumption attacks. Physical attacks are a concern as well and do belong to the domain of DoS attacks, but in this study they will not be discussed. A vast majority of DoS attacks can be countered relatively efficiently; for instance, attacks that target software can mostly be eliminated by fixing the faults in the software. However, attacks that target network resources are more of a problem. As Houle and Weaver (2001, 1-2) among many others have pointed out, bandwidth consumption attacks are built within the principles of the Internet and thus there is no comprehensive solution to be found. Based on that, it appears that any absolute solution would require a change in the principles themselves.

Distributed denial-of-service (DDoS) attacks belong to a subset of DoS attacks and along with computer viruses and worms, they can cause severe problems in today's computerized world. DDoS, or DDoS attack, is a commonly used term, which refers to a DoS attack using multiple attacking sources and is characterized by coordination (Mirkovic et al. 2002, 2-3), (Spech and Lee 2003, 1-2). Although not a requisite, DDoS attack is usually aimed to exhaust network resources, which means that DDoS attacks most often are bandwidth consumption attacks.

The severity of DDoS attacks that target network resources mainly results from a few key points. First, *the power of many is usually greater than the power of a few* (Mirkovic et al. 2002, 2). This point refers to the Internet's intrinsic inability to manage bandwidth consumption attacks, as according to the Internet's best effort and end-to-end design principles, any host can send any amount of traffic to any other host as fast as possible. Subsequently, when a DDoS attack is commenced using a large number of attacking hosts, often the target's bandwidth resources are quite easily exceeded. Second, *attack and normal traffic can be by their very nature indistinguishable*, which leads to difficulties in mounting and designing efficient countermeasures. The point is based on the fact that there is no reason why the attack traffic should look any different from the normal traffic. Third, as Savage et al. (2000, 1) hinted, *the indirect nature of the attack induces significant difficulties in tracing the one or more original sources of the attack*, which makes the capture of the culprits and the shutdown of the attack problematic. Last, Mirkovic et al. pointed out that the *security of the Internet is interdependent*. Essentially, this means that every host that can be compromised can be used against every other host connected into the Internet (Mirkovic et al. 2002, 2). In other words, poor security of an arbitrary host connected to the Internet is a problem shared by everyone else connected into the Internet.

Even though the DDoS attack technology has existed at least from six to seven years, DDoS attacks were not much of mainstream interest in neither public nor research circles before the economically major hits to such giants as Yahoo!, Amazon.com, CNN and eBay during the year 2000. After those events, media started noticing the phenomena and increased amount of public research got dedicated to it, which has been the trend ever since. Today, quite a few research articles have been published and a large amount of informal material discussing

the topic from various viewpoints can be found in the Internet. It is only reasonable to expect that military has its own research concerning the area as well.

Recently, DDoS attack capability has also been planted to computer "worms", which is another term for automated intrusion agents. The problem is real, as with high-speed fully automated propagation techniques computer worms can infiltrate millions of hosts in a matter of minutes (Staniford, Paxson and Weaver 2002; also see Voyiatzis and Serpanos 2003). In case these worms were armed with DDoS functionality and were fully controllable, the consequences could be severe. Nevertheless, the required technology has existed for a few years already. Unfortunately it is another open problem looking for an answer.

Finally, it has to be taken into account that DDoS attacks are not performed only by young cyber vandals often referred as "script kiddies" (The Jargon File 2003) anymore, but also by people with more fine-tuned objectives in mind. The motives are numerous, such as terrorism, and the possible damages can be severe.

1.1 Research objectives

The main purpose of this study is to provide a clear and thorough coverage of the area of denial-of-service attacks in the Internet. In principle, this study attempts to aid the DDoS research to evolve by providing general consistency into the field and insight into issues yet not thoroughly considered or brought together, especially into the field of attack mechanisms. The study does not consider attacks that occur outside the Internet, although there are a few exceptions where a broader view will be momentarily adopted. The focus is in

DDoS attack mechanisms of which a new classification was formed. The study is based on a comprehensive literature review, which spans an area of source codes and analyses of DoS and DDoS attack tools, news reports, academic articles and technical reports.

Generally, the research regarding the area of DDoS has concentrated in defence mechanisms and tools and it has left the attack territory practically untouched. To this day, it seems only few research papers, such as the studies by Mirkovic et al. (2002) and Spech and Lee (2003) have attempted to classify or explain the different attacking mechanisms, methods and other issues regarding attack technology, and even these studies seem to reside on overly generic levels. The lack of research dedicated to DDoS attack technology in turn undermines the basis on which the defence mechanisms against DDoS attacks have been developed. Comprehensive understanding of the attack technology is a necessity for designing proper defence. Without research to the attack technology the required level of knowledge of the attack technology cannot be established.

In this study the DDoS attack mechanisms and their unique aspects will be analysed and explained in detail in an attempt to clarify the previously described rather vague area of research. The discussion is based on analysing the core principles of the DDoS attacks and on a novel classification of DDoS attack mechanisms that was created for this study. The analysis of the core principles of DDoS attacks lays the foundation for the classification and it shows that any network similar to the modern Internet cannot provide an absolute defence against DDoS attacks. The classification in turn was formed to clearly separate and represent the aspects of DDoS attacks and to be the first comprehensive depiction of the core of DDoS attack, which makes the

classification a novel addition to the research field regarding DDoS attacks. Furthermore, the classification enables specific and realistic discussion of the evolution of DDoS attacks, which is another relatively undiscussed topic. The classification was built by analyzing the properties of DDoS attacks in which analyzing the logic and the functionality of live contemporary DDoS attack tools was in an important role. This study also depicts some ways in which the DDoS attacks may be evolving in the near future.

In addition, there appears to be both confusion and ignorance regarding appropriate terms and their exact meanings. Several studies appear to have used either inaccurately or erroneously some of the basic terms, such as the terms DoS and DDoS. For instance, the concepts of DoS and DoS attack may have been regarded equal, as was in the study by Kargl, Maier and Weber (2001, 2) or the concept of DoS may have been used instead of the concept of DoS attack, as was in the study by Gresty, Shi and Merabti (2001, 1). Similarly, several studies have not defined, characterized nor referenced the terms DoS, DoS attack and DDoS at all, which is not much better either, as the concepts and terms lay the foundation on which the study will be built. These particular terms and concepts still seem to lack proper definitions, which emphasizes the point that the definitions should not be disregarded. For example, during the year 2003 Shalunov and Teitelbaum (2003, 2) noted that they were unable to find a broad enough definition of DoS in the literature, which caused problems to their research. In this study, the key concepts will be extensively discussed in an attempt of bringing uniformity and consistency to the terminology.

The focus of this study is in DDoS attack mechanisms and in the DDoS attack field. Issues related to defence against DDoS attacks will be discussed only at the level that is required for understanding the rest of the study. The approach

is to start from generic issues concerning the DDoS attack field and then advance into details of the DDoS attack mechanisms. The prime objectives of this paper can be summarized to

- clarify and uniform terms, concepts and definitions,
- analyse the details of DDoS attack mechanisms and the principles DDoS attacks rely,
- present the novel classification of DDoS attack mechanisms,
- discuss a few of the possible evolutions of the DDoS attack mechanisms and
- provide a clear, consistent and thorough explanation of the subject in question and thus enable extensive comprehension of DDoS attacks.

1.2 The organization of the thesis

The second chapter introduces and defines the key concepts and terms used in this study. Either most of the definitions are modified versions of what the previous studies have proposed or entirely new when the concepts they refer to appear to lack proper definitions. These definitions are explicitly mentioned. The second chapter also provides an overview of the field of denial-of-service attacks, which for instance includes a glance to the history of DoS attacks and an introduction to the subject of DDoS and cybercrime. The chapter is largely based on literature review.

The chapter three introduces the principles on which the DDoS attacks rely, reasons why the new classification of DDoS attack mechanisms was created and the basis for the new classification. The chapter three also provides a figure and an overview of the classification as well as explains the functions of the main classes of the classification. The chapters four and five discuss the classification

of DDoS attack mechanisms in length. These two chapters reflect the two main and the most important classes of the new classification, which in turn represent the two most fundamental characteristics of DDoS attack mechanisms. The chapter four details the DoS attack mechanisms in theory and in practice. The chapter five similarly details the DDoS network mechanisms. The chapters four and five both conclude by depicting a few ways how the mechanisms they present can evolve. In addition, the chapters four and five include summary tables of their respective mechanisms due to the relatively significant amount of information presented and due to the emphasized importance of these chapters. The chapter six briefly discusses the contemporary countermeasures against DDoS attacks and presents a classification of the countermeasure types. These chapters are mostly constructive, although they are based on comprehensive literature review.

The chapter seven concludes the study. A review and discussion of the most important results achieved will be presented. In addition, topics for further research will be proposed.

2 THE FIELD OF DISTRIBUTED DENIAL OF SERVICE

The aim of this chapter is to introduce the field of denial-of-service (DoS)-, and distributed denial-of-service (DDoS) attacks. In the first paragraph the most important terms and the key concepts of the field are defined. The second paragraph outlines some of the most important events in the past that relate to DoS attacks. In the third paragraph a view of information security in regard to DoS attacks is presented and an indication of the threat of DDoS attacks is outlined. The fourth paragraph introduces the most basic issues of DDoS as a cybercrime activity.

2.1 Key concept and term definitions

The purpose of this paragraph is to define the key concepts and terms used in this research. At the time of writing some of these concepts were either inaccurately defined, or not defined at all, which is why special emphasis was put into the definitions of the key concepts as an attempt to uniform the terminology of this research field. In addition, a few common terms were defined in order to avoid confusion of intentions between this study and other sources of information.

The definitions and terms discussed in this chapter were chosen, created or modified to suit this research field and the purposes of this thesis best. In this study these definitions were used in the context of denial-of-service attacks in the Internet.

2.1.1 The Internet

Since this study focuses into the DDoS attacks in the Internet and the word Internet could be used at least in two different ways, it is reasonable to define how the word is used in this study. According to the Federal Networking Council (FNC) Resolution (1995),

> "The Federal Networking Council (FNC) agrees that the following language reflects our definition of the term "Internet".
>
> "Internet" refers to the global information system that --
>
> (i) is logically linked together by a globally unique address space based on the Internet Protocol (IP) or its subsequent extensions/follow-ons;
>
> (ii) is able to support communications using the Transmission Control Protocol/Internet Protocol (TCP/IP) suite or its subsequent extensions/follow-ons, and/or other IP-compatible protocols; and
>
> (iii) provides, uses or makes accessible, either publicly or privately, high level services layered on the communications and related infrastructure described herein."

The word “internet”, with a lowercase “i”, refers to a collection of networks connected with each other. In this sense the Internet could be pictured as the most comprehensive collection of networks interconnected.

2.1.2 Client, server / service provider, service

Denial-of-Service is essentially an availability problem in a client/server environment, as denial-of-service is about inability of the client to obtain *service* from *a service provider*. Hence, the definitions for *client* and *server* were taken from the literature regarding client/server architectures.

As Lewandowski (1998, 1) states, “Clients serve as consumers in a client/server system. That is, they make requests to servers for services or information and then use the response to carry out their own purpose.” This characterization of a *client* was used in this study.

In addition, Lewandowski (1998, 1) characterizes the meaning of a *server* in a manner that goes well within the requirements of this study. He states, “The server plays the role of the producer, filling data or service requests made by clients.”

Based on these characterizations of client and server a *service* could be defined in this context as *any kind of information the server provides to all of its legitimate clients requesting it.*

2.1.3 Denial-of-Service, Denial-of-Service attack

As it was pointed out in the introduction, the concepts of denial-of-service (DoS) and denial-of-service attacks are often used wrong, such as considering them as the same. Nonetheless, denial-of-service and denial-of-service attack are two completely different concepts where the former refers to an event or a situation and the latter refers to an intent driven illicit act.

Howard (1997) states, “The most comprehensive perspective would be that regardless of the cause, if a service is supposed to be available and it is not, then service has been denied.” The definition of *denial-of-service* used in this study was created on this basis.

Denial-of-Service (DoS) is an event or a situation, in which a legitimate client cannot access the requested service to which the client is entitled to and which should be available.

According to CERT (2001), "A "denial-of-service" attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service." Howard (1997) stated the same in a slightly more verbose manner, "A *denial-of-service attack*, therefore, is considered to take place only when access to a computer or network resource is *intentionally* blocked or degraded as a result of malicious action taken by another user." With a slight modification, the definition provided by CERT (2001) is the definition for *denial-of-service attack* used in this thesis.

A denial-of-service attack is characterized by an exclusive function of the attack and an explicit attempt by one or more attackers to prevent one or more legitimate users of a service from using that service.

With these modifications, the stress is on two important points. First, the number of targets or attackers is irrelevant. Second, the single purpose of the attack must be to cause a denial-of-service, which means that if the attack has any other functions besides causing DoS the attack cannot be categorized as a DoS attack.

As some other attack types may cause DoS situations as a side effect that nevertheless could be argued to be intentional, the second point is extremely important in constraining what can and cannot be classified as a DoS attack. Otherwise the definition would cover excessively wide area of seemingly different attack types, which would make the definition rather useless.

For instance, it is common for viruses and worms to consume much of both host and network resources while propagating and executing their primary functions. This has often leaded to severe DoS situations, such as what was witnessed with the Morris Worm in 1989 (Orman 2003, 1-2). The problem is that in cases like this the intent of causing DoS is practically impossible to prove. It is entirely plausible to assume the attacker expected DoS situations to occur; for instance, due to the heavy propagation traffic. This in turn would qualify the requirement of intent and consequently it could be stated that the launching of worms and viruses is a DoS attack. The problem is that this applies to many other attack types as well, which would make the concept of "denial-of-service attack" more of a generic description rather than a specific type of an attack. Based on that, it is emphasized that the intent alone is not a good enough requirement, but the primary function of the attack has to be stated as well.

When it is required that the attack's exclusive function is to cause DoS a clear conceptual separation between true DoS attacks and side effects of other attacks can be made. This goes well with the example of viruses and worms, as it is commonly acknowledged that the two primary functions of viruses and worms are to infect and propagate, no matter what their final objectives may be. Based on this, worms and viruses can launch DoS attacks; however, the launch of worms and viruses cannot be considered as a DoS attack. Therefore, in case DoS occurs due to a side effect of some other attack, such as the traffic generated by worm propagation the attack does not qualify the requirements of a DoS attack.

As a last example, Gresty et al. (2001, 1) pointed out a service can be denied due to malicious service or data modification. Their point was that as the service the client expects to receive is unavailable (the client receives maliciously modified data, which is not what the client requested) the client is effectively denied from

the requested service. Such a scenario is a matter of denial-of-service; however, according to the new definition, it is not a denial-of-service attack. The data modification is in itself an attack, which has a side effect that is a denial-of-service.

2.1.4 Distributed Denial-of-Service attack

Probably the most common definition of a DDoS attack follows the idea of having multiple machines each deploying a DoS attack towards one or more targets (Mirkovic et al. 2002, 1; Stein and Stewart 2002). Such a definition is almost correct, however, it fails to include the aspect of coordination between the attacking hosts, which is the most fundamental characteristic of a DDoS attack. For that reason a new definition was formulated.

Distributed Denial-of-Service (DDoS) attack is a DoS attack, in which a multitude of hosts performs DoS attacks in a coordinated manner to one or more targets.

This definition emphasizes three important aspects. First, DDoS is essentially a DoS attack. More accurately, DDoS attacks are a subset of DoS attacks. Second, there must be more than one source attacking. Third, there must be coordination between the attacking hosts. In case either one of these conditions is not met the attack cannot be called as a distributed denial-of-service attack.

In this study the abbreviation *DDoS* is often used to refer to distributed denial-of-service attacks. This is an important point to notice, as whereas there is such a thing as denial-of-service, there is no such a thing as distributed denial-of-service in the same sense; the service can be denied, but the service cannot be denied distributed unless the service itself is distributed. Only distributed

denial-of-service attack can exist. The word “attack” is appended only when it is considered there is a chance of misunderstanding; For instance, DDoS tool could refer to both DDoS attack- and defence tools. Therefore, the clarifying word would be appended in that particular case.

The possibility of arbitrary attackers randomly selecting the same host as a target should also be noted, as this event is not a DDoS attack, although it may seem to be. The view adopted in this research considers this possibility as what it in most simplistic level appears to be; separate attackers engaging in separate DoS attacks against the target, as contingency is hardly a form of coordination. Furthermore, an *attack* as a concept is singular in regard to its objectives, which means that even in case an attack is an aggregate of other attacks, all of the “sub attacks” strive for the shared objectives. Random target selection is not a shared objective. Therefore, in case multiple attackers each randomly select the same target the resulting attack cannot be reasonably stated as a shared attempt in achieving a shared objective amongst the attackers.

2.1.5 Distributed Denial-of-Service network

DDoS is always about multiple DoS attacks targeted to one or more specific destinations. As it was argued in the previous subparagraph, coordination is a crucial part of DDoS. Coordination of multiple hosts in turn implies the existence of some sort of a network structure, which could be titled as *DDoS network*. However, for the purposes of this study, no suitable definition for such network was found. Therefore, a tentative definition for *DDoS network* was created.

DDoS network is a network of hosts that are being controlled by a same static entity using the same control interface to administrate DDoS attacks.

The "control interface" refers to the specific methods the entity can use to control the hosts. Essentially, the control interface is sort of a protocol, which the network apprehends. The notion of the static entity, whether or not it consists of several subjects refers to the controller of the network. The combination of the control interface and the network controller serve as a network identifier. In case these qualifiers were not determined, it would be practically impossible to identify a network and state which hosts belong to which network. Furthermore, the purposes of the network must include the administration of DDoS attacks. Otherwise the definition would come overly broad, as it would cover many seemingly different computer networks as well.

2.1.6 Term definitions

Host is a computer connected to a network (FOLDOC 1993). This particular term is preferred and commonly used when referring to a networked computer. In this study this practise was followed.

DDoS attack software is a program or a set of programs that contributes in some way to the functionality of a DDoS network.

End-host is the last node or the node with the least amount of authority in a DDoS network responsible of attacking. In other words, end-host is the node that performs the actual attack as ordered by some other node with higher authority in the same network.

DDoS networks most often consist of nodes with different tasks amongst each other. Certain nodes may only be coordinating other nodes, which ultimately inject the attack traffic. End-host is always the host responsible of injecting the attack traffic. DDoS networks usually consist of large number of end-hosts and may additionally consist of some other nodes responsible of various other tasks, such as the coordination of the end-host activity.

DDoS zombie / *DDoS slave* / *DDoS daemon* / *DDoS host* / *DDoS agent* is an end-host computer program. Depending of the author, most commonly one of these terms is used to refer to the end-host program. For instance, Kargl et al. (2001, 4) used the term zombie, whereas Mirkovic et al. (2002, 2) used the term agent. Different terms could be used as well. In this study, *DDoS agent* was used, as it appeared to be quite commonly used.

DDoS server / *DDoS handler* / *DDoS master* is an intermediate computer program in the DDoS network, which controls a set of DDoS agents and is usually being directly controlled by a cracker or another intermediate computer program. As in the case of end-host programs, all these terms are commonly used in the literature. For instance, Kargl et al. (2001, 4) uses the term DDoS master, whereas Mirkovic et al. (2002, 2) uses the term DDoS handler. Again, other terms could be used as well. In this study, *DDoS handler* was used.

DDoS client is a computer program, which a cracker uses to control DDoS handlers and DDoS agents. These programs provide the user interface to the DDoS network.

Cracker is an individual, who by his / her action seeks either to cause damage or to gain an unauthorized access to a computer system or systems. Some define a

cracker as an individual who attempts to gain an unauthorized access to a computer system (FOLDOC 1993; Wordnet 2.0). Some others define a cracker as an individual who seeks to cause damage in computer systems. In this study the term cracker is used to refer to both.

Hacker is an individual who enjoys exploring the details of programmable systems and how to stretch their capabilities (FOLDOC 1993). The term hacker is widely used and often many groups of people perceive it differently. Many mistakenly confuse the terms hacker and cracker with each other.

IP spoofing is a technique of masquerading a source address of the Internet protocol (IP) to appear as something else than it really is.

Packet is "the unit of data sent across a network. "Packet" is a generic term used to describe a unit of data at any layer of the OSI protocol stack, but it is most correctly used to describe application layer data units ("application protocol data unit", APDU)" (FOLDOC 1993).

Datagram is a "self-contained, independent entity of data carrying sufficient information to be routed from the source to the destination computer without reliance on earlier exchanges between this source and destination computer and the transporting network" (FOLDOC 1993).

Worm is a program that propagates itself over a network and simultaneously reproducing itself (FOLDOC 1993). The term is often used instead of *automated intrusion agent*, which is another term used occasionally to refer to a similar program.

2.2 A glance to the past of DoS and DDoS attacks

In this paragraph, a few of the most notable events of the past regarding DoS attacks are briefly discussed. The information presented here is based on public sources only.

In February 1996, CERT published an advisory regarding a User Datagram Protocol (UDP) port DoS attack (CERT CA-1996-01 1996). In the history of CERT this was the first advisory issuing a clear DoS attack warning. In September 1996, CERT published another advisory alerting of a Transmission Control Protocol (TCP) SYN flooding DoS attack (CERT CA-1996-21 1996). This particular attack was originally discussed in a well-known hacker magazine Phrack with the source code of the implementation of the attack included (Phrack Magazine 1996). Since then this attack type has received a rather notorious reputation amongst the attacks in the Internet and it is probably the most used DoS attack of all time. Most DDoS attack tools have implemented this attack type and even specific countermeasures have been developed against it. One example of such a countermeasure is SYN cookies (Bernstein 1996). In December 1996, CERT published the third and the last DoS attack advisory of the year (CERT CA-1996-26 1996). This time the advisory dealt oversized Internet Control Message Protocol (ICMP) echo request packets that when encountered could have halted certain operating systems. This particular attack is often referred as the “Ping of Death”.

The year 1996 is a landmark in the history of DoS attacks. Certainly, there must have been flaws in software that might have resulted in DoS before and it is quite reasonable to assume that some underground circles might have known these attacks as well as many others long before public did. However, these three attack types are the first widely noted DoS attacks.

The year 1998 started with an advisory from CERT detailing a so-called “smurf” DoS attack (CERT CA-1998-01 1998). Even as this particular attack is launched from a single node, it might result in multiple machines sending traffic to the target. Attacks like these are often referred as amplified DoS attacks. These types of attacks are not equivalent to DDoS attacks.

The potency of smurf attack proved to be real. Wired reported several Internet Service Providers (ISP) and Internet Relay Chat (IRC) administrators having serious problems getting their networks online shortly after the year change in January 1998 (Glave 1999). A few months later CNET reported that the University of Minnesota was suffering also of a smurf DoS attack (Festa 1998).

In July 1999 CERT published reports noting widespread exploitation of RPC vulnerabilities (CERT IN-99-04 1999; CERT IN-99-05 1999). Later on, it was found that in many cases this exploitation phase was followed by DDoS attack tool installation. It was this advisory that first time described two DDoS attack tools publicly (CERT IN-99-07 1999). These tools were Trin00 and Tribe Flood Network (TFN). Between these two TFN was more evolved regarding attacking capabilities. Trin00 could mount only a UDP flood, whereas TFN had UDP, ICMP and TCP SYN flood attacks as well as a smurf attack capabilities built-in. In addition, the advisory had a statement, which in a way gave an insight what to expect in the near future, “These tools appear to be undergoing active development, testing and deployment in the Internet.” (CERT IN-99-07 1999) Only a month later CERT published another advisory describing a new DDoS attack tool named Tribe FloodNet – 2k Edition (TFN2K) (CERT CA-1999-17 1999). TFN2K was the improved follower of TFN coming from the same author. Even as it contained several new ways to flood the target that were not present at the predecessor, the other new functionality proved to be more disturbing.

Most dramatic improvements were aimed at rendering communication between nodes in DDoS network more obfuscated (TFN2k 1999).

The years 1998 and 1999 are significant in the history of DDoS, since during those years the first publicly available DDoS attack tools, such as the previously mentioned Trin00, TFN and TFN2k were developed. During 1998 before these tools were available in the Internet there were conceivably other DDoS attack tools already built, such as fabi and BlitzNet. The exact chronological evolution of DDoS attack tools could not be verified, but in any case, tools that might have existed before Trin00 and TFN did not spread around the Internet as they did. The early DDoS attack tools were probably being developed and known only in small underground cracker and hacker circles.

The year 2000 started with another CERT advisory describing a new DDoS attack tool named Stacheldraht and discussing the developments in the field of DDoS (CERT CA-2000-01 2000). Stacheldraht, in a similar fashion to TFN was another improved DDoS attack tool based on Trin00 and original TFN. The main improvement was the addition of encryption to communication. This feature was present in the TFN2K as well. In addition, Stacheldraht provided automated update of the agents. A feature yet unseen in the DDoS attack tools.

In February 2000, something previously unseen in the history of the Internet occurred. The wave of massive DDoS attacks began. Among many other news sites, BBC News reported that Yahoo! was brought down for three hours (BBC News 2000). A day later eBay, Buy.com, CNN.com, Amazon.com were all under heavy DDoS attacks as reported by Seattle Post-Intelligencer (2000). The duration of these attacks was similar to the Yahoo! attack and thus let experts to believe they were connected. The magnitude of these attacks was something

completely unexpected. For instance, it was noted that in some cases the overall amount of incoming data was over a Gigabit per second (Garber 2000, 1). The frenzy the Yahoo! attack bred was certainly justified; if the company who had probably the greatest web resources could be taken down then anything can be taken down. The events escalated into the media and DDoS attacks received the attention they should have received a long time ago.

In 2002, a new DDoS attack tool named PUD appeared in the Internet (PUD 2002). This particular tool uses a custom coordination method, which is based on peer-to-peer ideology to control its instances. The coordination method is novel to DDoS attack tools and it is much more robust and scalable compared to the traditional derivatives of the agent-handler model. The same coordination method with identical source code was adopted into the Slapper worm (Slapper 2002), which targeted Apache web servers operating on Linux (CERT CA-2002-27 2002).

After the events of February 2000, many devastating DDoS attacks have kept occurring in the Internet, although the DDoS attack technology has not evolved much. For instance, in January 2002 one particular attack resulted in a drastic aftermath as a British Internet service provider was forced to close its doors (The Register 2002). During the year 2002 also occurred probably the most severe DDoS attack in the history of the Internet, as the thirteen root DNS servers were simultaneously targeted and seven of them were knocked out of the Internet (washingtonpost.com 2002).

In March 2003, the Arab satellite television network Al-Jazeera was forced out of the Internet and some speculation was expressed that the attack might have had something to do with the U.S led war in Iraq (Infoworld 2003). The year

2003 also hinted of the vast power the worm and the DDoS attack technology together hold, as the Blaster worm managed to infect more than 1.4 million computers worldwide (Berghel 2003 1). The purpose of the original Blaster worm was to initiate a DDoS attack against the Microsoft, but the attack traffic never reached the original target as was initially designed by the attacker. During the year 2003 it also became apparent that spammers were starting to use DDoS attacks to disable anti-spam blacklists as a means of spreading spam more efficiently (Tynan 2003). That way the ones who relied on those particular sites on filtering incoming spam were more open to spammers again.

Overall, a lot has happened during the past few years in the DDoS field as a whole. DDoS technology has evolved and attacker motivation and identity has faced a shift as well; it is not reasonable anymore to expect that only young misfits, often referred as script-kiddies use DDoS technology in their personal vendettas and quests for glory. The considerably large power of DDoS attacks have been noted everywhere, which is verified by the events discussed previously. The rather recent event of businessman hiring a cracker group to perform DDoS attacks against three of his competitors emphasized the notion that today DDoS attacks are being used as tools of achieving more fine-tuned criminal objectives as well (Poulsen 2004).

2.3 DoS attacks and the related problem magnitude

In this paragraph it is first described how DoS attacks relate to information security and then some indication of the problem magnitude and the threat of DoS attacks is provided.

2.3.1 Information security and DoS

Computer and information security is traditionally defined by three distinct concepts, which are *confidentiality*, *integrity* and *availability*. *Confidentiality* aims to ensure information can be accessed only by those who are authorized for it. *Integrity* aims to ensure information is always accurate, consistent and complete. *Availability* aims to ensure information is always available to those authorized for it without any service or access degradation (well described by Jonsson 1998, 3-4).

As pointed out by Howard (1997), DoS attacks aim to assault against *availability*. Based on the definitions provided by Howard (1997) and Houle and Weaver (2001), it can be further stated that only an attack that targets availability as its sole function is a DoS attack. Other attacks that cause availability problems by primarily attacking integrity or confidentiality, for instance in a manner Gresty, Shi and Merabti described (2001, 1) cannot be considered as DoS attacks. In other words, in a theory third party cannot get an access to protected data using DoS attacks nor a third party can corrupt data by performing DoS attacks. However, DoS attacks may be some way involved in the overall attacking strategy against confidentiality and integrity breaches. It also should be noted that in practise DoS attacks might indirectly result in data corruption through a system flaw of some sort. For example, in case a DoS attack crashes a system while a disk write operation is being performed data could result corrupted. Still, physical damage should never occur only because of successful DoS attacks occurring in computer networks.

2.3.2 The problem magnitude and the threat of DoS attacks

As Moore, Voelker and Savage (2001, 1) noted a few years back, there is not much public data available regarding the frequency and characteristics of DoS attacks, and it seems the situation has not changed much since. The study Moore, Voelker and Savage performed was probably the first of its kind and it showed that approximately 4000 DoS attacks occur weekly in the Internet (Moore, Voelker and Savage 2001, 1, 6). Due to the limitations of the method they used, the authors remarked that the estimate was probably an underestimate of the total number of attacks (Moore, Voelker and Savage 2001, 4).

In addition to the study by Moore et al. (2001), only the computer crime and security surveys carried out by the FBI and the CSI were found to give some indication of the prevalence of DoS attacks. Their most recent survey reported that 17 percent out of the 481 respondents experienced one or more DoS attacks during the last 12 months (Gordon et al. 2004, 10), which quite surprisingly is significantly less compared to their findings a year before (Richardson 2003, 10). According to the same survey, denial-of-service attacks were the second most costly malicious activity performed, causing approximately $26 million worth of damages to the 269 respondents in total (Gordon et al. 2004, 11), which was also notably less compared to the year before. However, it should be noted that these surveys are based on the responses of relatively small number of U.S. computer security practitioners, such as government agencies, financial institutions and universities, which is why these results are indicative at best. Nevertheless, these findings still show the potency and topicality of DoS attacks.

Although DoS attacks are quite likely the most prevalent in the Internet, it does not imply DoS attacks are not or could not be a problem elsewhere. According to the definitions of a DoS attack provided by Houle and Weaver (2001) as well as Howard (1997), DoS attacks can occur anywhere, anytime and by any method. The definitions cover everything from physical assaults to software exploits requiring only a malicious intent to cause DoS to coexist.

The prevalence of the problem is emphasized, as the motives to perform DoS attacks are numerous even from the technical standpoint alone. For instance, DoS attacks can be very efficient decoys for hiding penetration to other systems besides the target of the DoS attacks. Likewise, certain penetration and spoofing attacks require specific machines to be momentarily unable to access the network (Bellovin 1989, 1-4). Furthermore, certain software might misbehave unexpectedly under an extreme DoS attack and enable the attacker to gain administrator level privileges to the system.

Besides being a problem to networked computers, DoS attacks are a problem to offline computers as well. Otherwise stated, DoS attacks belong both to *local* and *remote* threat classes. The *local threat class* refers to attacks, which require only a local access to the computer. Without proper system administration, these attacks most commonly require only a normal user account to the system to be attacked. The *remote threat class* refers to attacks executed through some type of a network medium, such as the Internet.

The local DoS attacks lack severity, as they usually can be well countered with appropriate system administration. These attacks most commonly are about exploiting flawed programs and poor system configuration. Due to this and because local DoS attacks are always singular in nature, that is, they come from

only one source and target only one computer, they will not be further discussed in this study. The focus is in remote attacks, which can be problematic due to the very design of the Internet, as will be shown in chapter three. As will also be discussed in chapter three, any computer connected to the Internet is a possible target of DoS attacks and added to that, even a minimal defence against certain remote DoS attacks may be difficult to mount.

Bandwidth consumption attacks have also a unique property of involving at least three types of victims. Assuming the attacker constructed the DDoS network by compromising hosts the owners of the compromised hosts could be seen as secondary or initial victims, as Kabay (2000) states. According to Kabay, "final" victims are those who receive the attack traffic. In addition, DDoS attacks based on bandwidth consumption affect the general performance of any link relaying the attack traffic and thus those links and any client or another link dependant of them could be seen as secondary victims of these attacks.

2.4 Issues relating to law and liability

The aim of this paragraph is to take a brief glance to the most basic issues regarding cybercrime and DDoS. The motivations for this type of discussion are the facts that DDoS is a severe act of cybercrime, and as the actual attackers behind DDoS are rarely caught and the damages suffered can be significant the questions of law and liability become important issues. The victims may seek compensation from some other party regardless were the actual attackers caught or not. Moreover, the victims may be subsidized by a jurisprudentially valid case.

First, a short overview of the problems regarding cybercrime is presented and then the most important arguments of liability are briefly discussed. The purpose of this chapter is not to provide legal advices. For such matter, a qualified attorney should be consulted.

2.4.1 Legislation and cybercrime

The issue of cybercrime has been troubling the legal environment for quite some time. One of the most profound problems lies in the controversial natures of technology and legislation. As Chen et al. eloquently put it, "It usually takes laws months, if not years, to be developed, approved and implemented. Technology, on the other hand, seems to change in a matter of days, or sometimes even hours." (Chen et al. 2002). In addition, cybercrime by its very nature is highly global. The culprit of an attack occurred in the United States could very well be somewhere in the Asia, which emphasizes that legislative cooperation between countries is a definite requirement to catch and prosecute cyber criminals accordingly. Unfortunately, global legislative cooperation appears to be still in its infancy with rough times ahead. Legislation regarding cybercrime may be inadequate in western countries; for all that, some developing countries may not have any legislation concerning the issue at all. Furthermore, some countries might even be harbouring criminals, which it is not reasonable to expect that even the basis for global cooperation concerning this issue be in near prospects.

Besides the legal problems, acquiring adequate technology and resources for monitoring and upholding the law can be costly and problematic. Consequently, even if cybercrime legislation were in order the high economic costs to which many countries simply cannot afford may prove to be yet

another voluminous obstacle to cross. Nonetheless, international cooperation is probably the only way to approach the problem efficiently.

2.4.2 Contributory negligence and downstream liability

In addition to issues regarding legislation, specifically in the case of DDoS, the questions of liability are essential. Generally, it is difficult to trace the attackers behind DDoS attacks (see chapter six), which is why attackers behind DDoS attacks are rarely caught regardless of the effort. Certainly, in the case of a successful trace and capture, attackers could be prosecuted accordingly in a crown court. However, as DDoS attacks may easily cause enormous economical damages, catching the actual culprit may not do much to the victim to compensate the damages endured. This in turn could lead to a search of some other party capable of appropriate compensation of damages and who could be held at least partly liable of the attack.

There has been lots of discussion concerning the liability of network administrators, company managers and hardware manufacturers. Kabay (2000) points out the important concepts of *downstream liability* and *contributory negligence* concerning the issue. He states that in case it can be proven that owners or administrators of "first-line" infected hosts have not reasonably secured their network, they could be prosecuted for contributory negligence. By first-line infected hosts he referred to hosts that were first compromised and later on used in attacks. He elaborates the issue well,

> What arguments would the plaintiffs' attorneys use in laying blame on the first rank victims? A strong case could be made using expert witnesses who would show that the vast majority of security breaches on sites linked to the Internet derive from out-of-date software and from inadequately configured defences. The witnesses would testify that fixes for well-

> known vulnerabilities have been available for years at no cost from software manufacturers, security firms, and from volunteers freely exchanging solutions. If subpoenaed, some of the network administrators from the slave-infested sites would testify that they knew that their sites were vulnerable, they knew where to get the fixes, but they just didn't have time to get the fixes installed. At that point, a clever attorney would ask, "Why not?"

It could be argued that the same should apply to software- and hardware vendors as well. However, along with other issues, the term “reasonably secure” becomes more difficult to define and with that proving the possible occurrence of contributory negligence. The line of what is seen as “reasonably secure” and what is not clear. There are, however, already motions of class-action lawsuits against certain vendors for contributory negligence (DDoS-ca.org).

At present, there seems to be no precise laws that would force service providers to enable special countermeasures to secure their networks. As the initial set-up and appropriate maintenance of “reasonable security” can be quite expensive, and as it does not directly return the service provider any benefit either, there are no direct economic incentives for engaging into such an effort. Moreover, as many service providers invoice based on the bandwidth usage, DoS attacks may prove rather beneficial for the service provider regardless of the questionable morale behind. Regardless, laws that would force service providers to engage in appropriate steps to secure their networks would probably mitigate the problem of security breaches notably. Similar would happen if individuals would better secure their home computers. Even if it would be seen as too harsh to hold individuals liable of their personal computers being used in attacks, individuals could be charged dynamically based on their bandwidth usage. High invoices of net usage would most likely

encourage customers to pay better attention to their personal computer security. The issues of economic incentives and pricing scheme designs were briefly presented by Lejeune (2002, 13-16).

3 THE ROOTS OF THE CLASSIFICATION OF DDOS ATTACKS

The key for understanding an arbitrary problem is to understand the domain in which the problem occurs and the factors that cause the occurrence. In this chapter firstly the DDoS attacks as a phenomenon is discussed and then the new classification of DDoS attack mechanisms is introduced. The theory behind DDoS attacks is discussed first and in length, since it provides the basis and the borders for the actual attack mechanisms and thus to the classification.

3.1 DDoS as a phenomenon

The problem with the current DDoS countermeasures is that none of them is able to guarantee any percentage of success in defence. Some of the countermeasures may be very effective in specific situations, but none can provide a firm, situation independent guarantee of efficiency.

The threat of DDoS is not only built into the contemporary network structures, but is inevitably present, as long as the points discussed here hold true. The aim of this paragraph is to discuss these fundamental issues regarding the nature of DDoS.

3.1.1 The theoretical impossibility of distinguishing malicious traffic

The first conclusion is based on the definition of DoS attack formulated in chapter two and it states that in essence *DoS attack traffic may appear as identical to normal traffic* (see chapters two and four).

The definition of DoS attack states that explicit intent and the sole purpose of the attack are the only aspects that separate DoS attacks and therefore DDoS from legitimate network activity. This is especially true in the case of bandwidth consumption attacks and thus well present in DDoS. Therefore, the statement translates as, no matter how high the traffic volumes may be or how abnormal the traffic streams may appear there does not have to be any malicious activity present and vice versa. The concept of flash-crowds (Jung et al. 2002, 1) is a fine example of sudden, high network usage without a malicious intent. On the other hand, an event that appears as a flash-crowd could very well be a DDoS attack, and the other way around. Basically, there is no way to tell (see chapter four).

The point is, "intent" and definitely "primary intent" accompanied with "purpose" do not belong into the domain of computable problems. Consequently, in case the attack and normal traffic are assumed as identical, the malicious traffic, and therefore malicious hosts cannot be separated from their legitimate counterparts. This leads to the conclusion that eliminating DDoS traffic without any side effects to any other traffic streams is impossible. Hence, defence mechanisms based on analysing traffic can never provide an absolute solution, as attack and valid traffic can be by their very nature indistinguishable.

3.1.2 The necessity of core network changes to deal DDoS properly

Based on the first conclusion, it seems that either the threat of DDoS would have to be (1) dealt prior any traffic is released into an arbitrary network system or (2) every possible target would have to possess superior amount of

bandwidth available compared to the added amount of bandwidth of all of its possible clients. However, neither one of these options is readily feasible.

In case resources were infinite, the second option would be valid and malicious bandwidth consumption would not be a problem. However, even without formal demonstration it can be stated that resources are and will always be finite. For that reason, it can be concluded that *arbitrary amount of bandwidth capacity can always be breached when hosts can contact limitlessly each other in the same network*. (It is here presumed that the network is not a static entity and it allows addition of new hosts and changes in bandwidth capacities of all hosts. This is the case in the modern Internet.) The notion of limiting how hosts can contact each other is emphasized, as without these limitations DDoS would have to be countered before any traffic is released (conclusion two). In the modern Internet, the limitations are built of traffic filters above the actual core of the Internet. The problem is that the Internet's best effort and end-to-end design principles leave the user complete control of traffic creation (see subparagraph 3.1.3). The Internet does not control in any way whether the user provides correct and appropriate information in packet headers. Hence, traffic filters cannot operate based on user identity, which forces the filters to rely on the properties of encountered packets and traffic flows. However, according to the first conclusion, normal and attack traffic can be indistinguishable, which inevitably results in either no filtering at all or filtering portions of every traffic stream equally. In any case the outcome is denial-of-service to some caused either directly or indirectly by a denial-of-service attack.

To sum up, without limitations resource boundaries can be reached as long as the resources remain finite. On the other hand, traffic can be filtered, but the

filtering will necessarily affect both normal and attack traffic. Because of this, it seems that the threat of DDoS has to be dealt prior any traffic is released.

Dealing the threat of DDoS prior any traffic is injected to the network is difficult. Reclined to the first conclusion, there is no computable way of knowing whether or not the intentions of users are malicious. Laws and regulations could certainly make difference in people's willingness to engage or participate into malicious activity. However, the problem is still likely to persist, as it has so far, regardless of the new laws imposed. Therefore, the only viable solution would appear to be a network infrastructure where the operations of each client could be controlled. Such an infrastructure would rely on configurable control mechanisms built to the main core of the network. For instance, such an infrastructure could guarantee that each host would be able to specify which other hosts are allowed to initiate a transaction with it and with what capacity. In that case "the server" would always have more bandwidth compared to the added amount of all of its possible clients together (by server it is referred to a host that is the target of connection / service requests. Thus, any host can be both client and server depending of the particular situation). However, at bare minimum this kind of scenario requires that every host would have to be uniquely and with absolute confidence identifiable. From this follows that this type of functionality cannot be placed on end-hosts, which would mean drastic changes to the infrastructure of the contemporary Internet.

3.1.3 The main principles of the Internet

The Internet was not built with security in mind. The Internet is based on so-called *best-effort* and *end-to-end* design principles and although they are the reasons for the Internet's high efficiency and popularity, they are the sources of

many inherent security problems as well. As discussed previously, DDoS attacks exist due to this design.

Coarsely speaking, the best-effort principle accompanied with the end-to-end principle means that the Internet is only concerned of *routing* packets injected to it *as fast as possible* to the specified destinations, leaving everything else for the end-hosts to handle (Blumenthal and Clark 2001, 1-2). This means that at the core level the Internet is concerned only of what the Internet Protocol portion of a packet embodies (see the definition of the Internet in subparagraph 2.1.1). The Internet Protocol specifies the network level header according to which the users are ought to construct their packets in order to transfer data through the Internet (RFC 791 1981). The Internet extracts information from the packets it encounters, specifically from the IP portion and operates based on the extracted information. The Internet is not concerned of whom or what created the packets it encounters or where they came and where they are heading. This means that everything above physical layer is left for the user to construct without restrictions. The stateless routing scenario cannot provide the means to authenticate the source of the traffic. Hence, the Internet provides complete user anonymity.

3.2 Classifying DDoS attack mechanisms

As it was mentioned previously, there are multiple ways of performing DoS attacks in the Internet alone. Although the methods are numerous, common characteristics of the attacks can be observed, which can be used as a basis for a classification.

It was stated earlier that the aspects regarding actual DDoS attack mechanisms have received a minimal amount of attention in the academic world. Mirkovic et al. (2002) proposed a taxonomy of DDoS attacks, which outlined the various aspects of DDoS attacks on a highly abstract level. Spech and Lee (2003) proposed another taxonomy of DDoS attacks adding some detail to some of the issues while leaving several other important aspects out. Both of these taxonomies lacked of detail and comprehensiveness. Moreover, these taxonomies did not ultimately describe what DDoS attack mechanisms are, how they operate and how they merge with each other. No other academic papers discussing DDoS attack mechanisms were found.

As a major part of this study a classification labeled as *DDoS attack mechanisms* was made in an attempt to explore, unify and clarify the various aspects of DDoS attacks. The classification is shown in FIGURE 1.

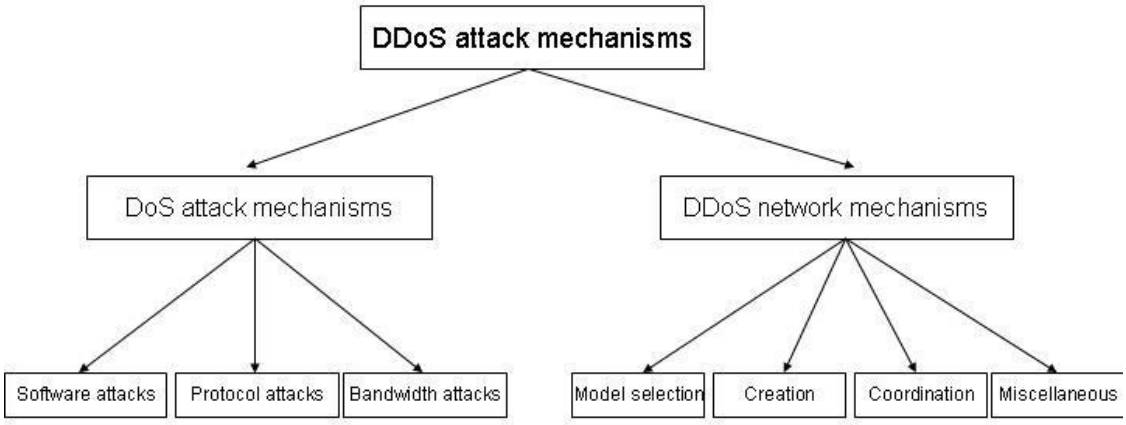

FIGURE 1. DDoS attack mechanism classification.

In this paragraph the classification is outlined. The chapter four discusses the first main branch of the classification named d*enial-of-service attack mechanisms*

in detail. The chapter five concentrates into the other main branch of the classification named *DDoS network mechanisms*.

The information on which the classification was built was gathered from live and publicly available DDoS attack tools. The source code of the tools used as references are: Blitznet (1999), DOSnet (2002), Distributed DNS Flooder (2001), Flitz (2001), Kaiten (2001), Knigth (2001), Mstream (2000), Omega v3 (2000), Peer-to-peer UDP Distributed Denial of Service (2002), Skydance (2001), Stacheldraht (2000), Tribe Flood Network (1999), Tribe FloodNet-2k edition (1999), and Trin00 (1999). Analyses of DDoS attack tools used as references are Trinity (Marchesseau 2000), Shaft (Dietrich, Long and Dittrich 2000), Power bot (Dittrich 2001) and GT bot (GT Bot 2003). The source codes of these tools could not be found for further inspection.

It is noted that the underground circles of the Internet are likely to possess technology not known to the public. Therefore, some ideas that the underground circles are likely to know, however, that are not present in above-mentioned tools are discussed as well.

3.2.1 The root of the classification

The root of the classification was named as *DDoS attack mechanisms.* In this context the term mechanism is used to refer to an abstract description of a procedure and the term method is used to refer to an exact procedure. In that sense a mechanism can be accomplished with multiple methods, whereas a method is atomic and unambiguous. The first division between the *DoS attack mechanisms* and the *DDoS network mechanisms* was chosen to describe the clearly distinct aspects of attack traffic creation and administration of nodes of DDoS

networks. Rather interestingly, this particular division between traffic creation and network administration was not noticed by Mirkovic et al. (2002) nor Spech and Lee (2003), even though it is probably one of the most fundamental characteristics of DDoS attacks.

At the fundamental level DDoS is multiple DoS attacks coordinated at the same set of targets. From this comes the naming *DoS attack mechanisms* to describe the appearance and creation of traffic aimed at causing DoS. The label of *DDoS network mechanisms* was chosen to refer to all other operations nodes in the DDoS network commence, as clearly there is always a network structure of some sort responsible of mutual coordination between the nodes. In turn, coordination as a term could be used to describe any kind of information exchange. Since these categories cover the aspects of actual attacking processes as well as any type of internal data exchange the nodes of DDoS networks might perform, this first level division is considered as comprehensive.

3.2.2 The second level divisions of the classification

In principle, DoS attack mechanisms reflect *the type of traffic* being injected to the network and destined to the chosen set of targets. The generated traffic may take various forms ranging from a single packet with specific characteristics to a stream of millions with random characteristics. However, without mechanisms of administrating a multitude of hosts that eventually create the attack traffic DDoS cannot be executed properly.

The "DoS attack mechanisms" -branch constitutes of three subcategories labeled, *attacks targeting software*, *attacks targeting protocols* and *attacks targeting bandwidth.* When physical attacks are excluded these three categories are clearly

comprehensive, although the labelling is arguable. In chapter four this branch along with these categories will be discussed in detail.

The "DDoS network mechanisms" -branch divides into the subcategories labeled *model selection*, *creation*, *coordination,* and *additional functionality*. These categories may appear slightly more opaque compared to the subclasses of the DoS attack mechanism -branch, even though they could similarly be counted as comprehensive in regard to the possible objectives of the DDoS network mechanisms. The categorization underlines the notion that each DDoS attack is built upon a network structure, which is created following a network model. The categorization also emphasizes that there must exist some type of coordination amongst the participants of DDoS networks in order to commence DDoS attacks. Instances of all of these first three mechanism classes are always present in DDoS attacks. The final subcategory named *additional functionality* was included to cover all other mechanisms that are not generic in nature and do not belong to the other three subcategories. This branch along with these categories will be thoroughly discussed in the chapter five.

4 DENIAL-OF-SERVICE ATTACK MECHANISMS

In this chapter the left branch of the classification named as *DoS attack mechanisms* is discussed. The branch is further divided into three mechanism classes that are *attacks that target software*, *attacks that target protocols* and *attacks that target bandwidth*.

It should be noted that even though these categories are comprehensive as well as distinct the actual attack methods often are not. A DoS attack may be an arbitrary blend of methods that belong to any of the three mechanism classes. For instance, the characteristics of the traffic created in bandwidth consumption attack may simultaneously fulfill the criteria of the two other mechanism classes. Due to this and because the exact attack attributes can be varied extensively the number of possible DoS attack methods is large, which is why it is impractical, if not impossible, to list every possible DoS attack method.

4.1 Attacks that target software

DoS attacks that *target software* rely on the attacker's ability to perform a function or an operation against the target software, which either immediately or eventually causes DoS situation. In other words, the aim of DoS attacks targeting software is either *system or software crash* or *system resource consumption* (often the slang term “nuking” is used to refer to these attacks). The targeted software can be anything ranging from operating systems to lightweight applications.

Attacks that target software are the eldest way of performing DoS attacks. As software will never be perfect, exploitable flaws and other deficiencies will continue to exist. Some of these deficiencies might prove to be efficient DoS attacks.

There are two requirements for this type of an attack to be successful. First, the attacker is required to *be able to contact* the target software. Second, the target software or the configuration of the target software must have *a fault or deficiency*, which can be exploited in a way that results in DoS.

For attackers software-targeted DoS attacks are suitable because they are by far the easiest to mount, possibly excluding the creation of an exploit. Usually, only one or a few packets at most are required for causing the desired outcome. This renders these attacks easy and fast to deploy from any kind of a host in a silent manner. Moreover, without appropriate defences applied these attacks are usually highly effective in causing DoS when the aforementioned requisites are met.

Usually DoS attacks are performed remotely; however, attacks that target software are a significant local threat as well.

4.1.1 Local attacks targeting software

As local DoS attacks are not a direct threat to the Internet and they are never components of DDoS attacks they are out of the scope of this study and will not be discussed beyond what is noted here. Husman wrote a decent introductory paper regarding the topic (Husman 1996).

It is important to notice that local DoS attacks can be highly effective and they can cause notable damages. Moreover, local DoS attacks are usually much easier to perform compared to remote DoS attacks, which is one of the reasons local DoS attacks should be well considered as a possible threat.

To perform a local DoS attack targeting software any method that efficiently consumes important system resources or crashes the system is adequate. In an insufficiently administered system the methods are usually numerous. Any user might be able to use as much of the system's resources as available. Even data loss, file system corruption or physical damage could occur in an extreme case.

A case in point of this is an intense recursive process creation coupled with intensive disk usage. Such a method in a poorly administered system is a definite way to consume all of the system's CPU and memory resources rapidly. Furthermore, the method could lead to more severe consequences. Essentially, the process creation could make the system overly busy to handle any user input and thus leave the administrator no choice but to hard boot. This in turn may result in a file system corruption as the system is suddenly interrupted while a disk is being actively used.

4.1.2 Remote attacks targeting software

Generally, remote DoS attacks adhere to the following two phases that are generic to remote exploitation, which means that the steps are the same regardless of the objective of the remote exploitation. For instance, the remote exploitation can be aimed to degrade system availability, gain unauthorized

system access or compromise the integrity of data. The phases are *environment setup* and *exploitation*.

First, in case the target software can only be exploited in a certain set of states, a successful attack must provide the required state transitions to successfully exploit the target. The volume of state transitions may be zero. For example, the target may be exploitable with a single packet regardless of the state of the target making the exploitation easy. Then again, successful exploitation may require several interrupt-free state transitions, which is likely to complicate the exploitation procedure. Anyhow, this first phase could be referred as *environment setup*.

To illustrate the idea, consider a situation where a web-server software has a vulnerability in its Secure Sockets Layer (SSL) connection initialization function. As SSL is an application level protocol built upon a TCP connection, a successful attack would require an established TCP connection with the server in prior to the actual exploitation. Furthermore, as the vulnerability would most likely be in some specific function of the SSL library, a successful attack would be required to lead the server process to a state in which the vulnerable function would be called next.

Second, a successful attack requires construction of the exploit, which will be sent to the target once the previous step is completed. As mentioned, the exploit might be a single packet with specific characteristics, such as erroneous protocol header values that the target is unable to handle properly. The second phase that consists of the construction of the exploit and transmitting the exploit to the target could be referred as *exploitation.*

A fine example of a classic DoS attack aimed to cause software to crash is an attack named “Jolt” (Roberson J., 1997). The exploit program is ludicrously trivial and contains only about 130 lines of code. The idea was to create erroneously fragmented and over-sized ICMP ECHO packets, which the target systems were unable to deal properly. At the time this method was effective against certain operating systems and their flawed TCP/IP stacks, but today this method is a remnant at most. Still, it clearly demonstrates the very essence of remote DoS attacks targeting software.

Although DoS attacks targeting software are made often to cause the target software to crash instantly, the alternative of consuming system resources is a commonly used attack as well. Intense and continuous stream of packets with software exploitation characteristics may be a very effective method in bringing a system to a crawl. This is why many contemporary DDoS attack tools use malformed packets that are normally used as individual attacks against software, even though the primary purpose of the attack could be to overwhelm the networking capacity of the target. To put it otherwise, such an attack has two *attack vectors*, where the primary attack vector is to consume the bandwidth resources of the target and the secondary attack vector is to consume the system resources of the target.

For instance, although any contemporary system should be able to handle the previously discussed “Jolt” attack without halting, the exploit may still abnormally increase the usage of system resources due to the overhead of attempting to assemble erroneously fragmented packets. Thus, there is a “flaw” or deficiency in the system's TCP/IP stack even if the system is able to restore normal state of operation once the attack stops.

4.1.3 Software attacks in contemporary DDoS attack tools

Even though the most of the studied DDoS attack tools manipulate the packet headers more or less randomly, none of the tools specifically create packets that are known to be software targeted DoS attacks and only a few of the tools create packets that might be problematic for some TCP/IP stacks to handle. The tools that manipulate packet headers have various packet flooding methods, but mostly the variance of methods appears to exist only as an attempt of providing better throughput of attack traffic to the target in different circumstances. For instance, normally a TCP SYN packet initializes a new connection and a TCP ACK packet acknowledges a received packet. In some situations a negligent firewall or a router on the border of a network could deny packets that appear to be connection initialization attempts, but it could allow packets that at first appear to be acknowledgements of established connections. Similarly, in some situations ICMP could be completely blocked as an unrequired protocol, but UDP could be allowed due to its high usability in video and audio streaming.

4.1.4 Defences against attacks that target software

The DoS attacks targeting software are the easiest to defend once the vulnerability becomes publicly known. A patch to fix the flaw in the software can be created, and thus *patching the software* appropriately is a definite solution. However, this is not always immediately possible; the software vendor might not be reachable or the patch for the vulnerability is not released yet. Alternatively, *a signature* of the attack can be created. A signature in this sense is either a raw byte sequence against which passing packets would be matched (see a brief description in Sommer and Paxson 2003, 1) or a stream of events, which for instance may be represented as network packets (see a brief

description in Vigna et al. 2003, 1). With this signature, or *pattern*, many modern network intrusion detection systems (NIDSs) (see for instance Sundaram 1996), such as SNORT (Roesch, 1999) can be configured to detect and filter the invasive packet or packet stream in case of an encounter. Both of these defence methods are relatively straightforward to deploy once the aforementioned pre-conditions are met. Furthermore, these particular solutions should work always once properly implemented and thus, attacks that target software are a part of the rare class of DoS attacks that can be countered efficiently.

In case the vulnerability is not publicly known, defences become more difficult to mount. Solutions can still be searched from the field of intrusion detection systems that are able to detect anomalies in network traffic and enact accordingly (briefly described in Sekar et al. 2002, 1; also see Kaleton Internet 2002, 9-10 and Sundaram 1996). An *anomaly* in this sense can be anything classified as abnormal network behaviour and is thus situation specific. These types of detection systems do not attempt to scan for static signatures, but instead attempt to statistically recognize patterns that can indicate to attack activity (Kaleton Internet 2002, 9-10; Stolfo et al. 2001, 6; Sundaram 1996). The negative side is that anomaly detection, or *statistical pattern recognition* is not as reliable as the signature-based approach, assuming the signature in the latter is properly constructed. For instance, as Sekar et al. briefly noted (2002, 1), "systems often exhibit legitimate but previously unseen behaviour, which leads anomaly detection systems to produce a high degree of false alarms". However, anomaly detection has the capability of reacting to attacks not in public knowledge, which is why these systems may be worth investing.

Lastly, a general solution to software targeted DoS attacks is to *firewall* the points of the network considered crucial for defence. According to Oppliger

(1997, 3), ”a firewall builds a blockade between an internal network that is assumed to be secure and trusted, and another network, usually an external (inter)network, such as the Internet, that is not assumed to be secure and trusted.” This depiction of a *conventional firewall* is relatively accurate when augmented with the notion that these types of firewalls can be used to secure access to individual hosts as well.

As a measure against software targeted DoS attacks the efficiency of firewalls depends of multiple factors, such as of the type of the protected target and of the security policy the firewall is configured to uphold. In the most simplistic case where a firewall is used to protect a single machine, the conventional firewall technology may guarantee solid defence against any type of software targeted DoS attack with relatively minor effort. By configuring the firewall to drop incoming packets that do not belong to established connections from the inside or connections that are related to them guarantees that illicit packets from the outside must belong to the aforementioned connection types or otherwise they will reach only the firewall. Still, even in this case the efficiency of defence relies on three assumptions.

First, the firewall software itself is required to be secure and immune to software targeted DoS attacks. Second, any unchecked communication port in the external network interfaces is a possible point for security breach. From this follows that maximum efficiency following this method even in the most simplistic case requires that there are no open points of entry to the outside, which effectively denies the possibility of hosting services to the outside. Third, the host inside the firewall is expected not to engage into activity that could compromise the firewall nor establish connections with parties that could exploit the connections to send malicious data through the firewall. While these

conditions hold true the defence based on conventional firewalls is likely to be very effective even in small networks, assuming inbound traffic to the network traverses only through the firewall. However, the efficiency of conventional firewalls in more complex situations, such as in large networks with complex topologies and diverse user activity to protect the network and the machines in it becomes increasingly more difficult to establish (Bellovin 1999; Ioannidis et al. 2000). Because of this Bellovin (1999) presented the concept of *distributed firewall.* However, the analysis of such system as a defence method against software targeted DoS attacks is out of the scope of this study.

It would be ideal if Internet service providers and other owners of intermediate routers would implement such methods to control malicious traffic passing their borders. Unfortunately, even today these types of validity checks are rarely performed due to the associated overhead and increased costs.

4.2 Attacks that target protocols

DoS attacks that *target protocols* rely to the attacker's ability to exploit specifications of the protocols in a way that will result in DoS. Comparing to the other types of DoS attacks, protocol attacks do not have certain static actions to be performed, such as what was already observed with the attacks that target software. On the other hand, there exists remarkably straightforward attacks like TCP SYN (Vivo et al. 1999, 3), but then there are also attacks like “Shrew” (Kuzmanovic and Knightly 2003), which are more sophisticated and difficult to implement. Differentiating protocol attack traffic from any normal and valid traffic is also more difficult than in the case of attacks that target software. The individual packets of the attack traffic stream may not contain any kind of a signature diverging from normal packets. The traffic streams, however, may

contain distinguishable patterns, such as abnormally high percentage of TCP SYN packets, which could be a sign of an ongoing TCP SYN attack.

There are two possible aims of DoS attacks that target protocols. The first is to exploit the specifications of the attacked protocols in a way that causes *performance degradation to the operation of the protocols*. The second is to exploit the specifications of the protocols in a way that causes *arbitrary problems to the systems using those protocols*. It is important to note that whereas the first is an attack directly *against the protocols themselves* the second is an attack *against the systems dependant of the protocols*. In any case, DoS attacks that target protocols are always only a remote threat.

A good example of a direct attack against a protocol is an attack named "Shrew", which attempts to exploit the TCP's retransmission timeout mechanism by relying on controlled traffic bursts that aim to force the TCP flow to enter timeout state repeatedly (Kuzmanovic and Knightly 2003). An example of a protocol attack against a system dependant of the protocol is the already mentioned TCP SYN attack (Vivo et al. 1999, 3), where the attacker exploits the TCP connection creation procedure specified in RFC 793 (RFC 793 1981, 29-34). Another example of an attack of the second type is the so-called reflection attack (Paxson 2001), in which hosts are tricked to send responses to packets with false source addresses to the targets. In brief, the procedure is to create packets with the IP source address set to be the IP addresses of the targets. These packets are then sent to arbitrary hosts or servers in the Internet, which are likely to send a response one or more times to the source address that was extracted from the IP portion of the packet.

4.2.1 Protocol attacks in contemporary DDoS attack tools

There are only a few protocol attack methods implemented in the contemporary DDoS attack tools. Nonetheless, almost every DDoS attack tool has an ability to perform a TCP SYN attack. There are DDoS attack tools that perform reflection attacks, such as the Drdos (Drdos v2.0 2002) and there are tools that attempts to exploit the DNS protocol (RFC 1034 1987, RFC 1035 1987), such as the ddsnf (Distributed DNS Flooder v0.1b 2001) and pud (Pud 2002). Besides these, contemporary DDoS attack tools are not able to produce many other protocol attacks. Furthermore, all of these attack methods are of the type that attempt to cause arbitrary problems to the systems using those protocols. Attacks that attempt to cause performance degradation to the operations of the protocols are yet to be witnessed.

4.2.2 Defences against protocol attacks

Generally, these types of attacks are somewhat more complex subject compared to the attacks that target software, as there are no common factors necessarily present. Due to the same reason, appropriate countermeasures may be considerably more difficult to mount, since every attack may require unique defensive methods to be applied. Attacks of this type are unified only by the attempt of exploiting the protocol specifications.

For instance, TCP SYN attack can be effectively countered by the use of TCP SYN cookies (Bernstein 1996). Then again, reflection attacks are highly problematic to eliminate and defend against without changing the specification of TCP. Similarly, Kuzmanovic and Knightly (2003) state that it is very difficult

to defend against “shrew” attacks without significant sacrifices of system performance.

4.3 Attacks that target bandwidth

Attacks that target bandwidth may appear as the easiest in nature, but in fact they are the most flexible and configurable of DoS attacks. These attacks aim to overwhelm the target or the links of which the target's networking capability depends with such an amount of traffic that it causes either partial or complete DoS to the target. It is emphasized that it is not essential for the attack traffic to reach the target; however, the attack traffic must be able to reach and congest the links of which the target's networking capability depends. For instance, such links could be the routers of the target's Internet service provider (ISP) that relay the target's traffic. Unlike the other two classes of DoS attacks, attacks that target bandwidth succeed always, given that sufficient amount of attack traffic is able to reach the target. However, at present the amount of traffic required to cause DoS may be very large. For instance, some high profile web portals may easily have gigabytes of bandwidth available. Because of that, bandwidth consumption attacks are often ineffective without the combined power of multiple attacking hosts.

As an example, consider a small organization connected to the Internet via two 10 Mb/s links. An attacker has been successful in compromising sixty home computers all connected to the Internet via 1024 Kb/s DSL lines. An attacker sets each machine to transmit packets with random header values to random IP addresses belonging to the domain of the target organization. The attacker also instructs the hosts to use approximately one fourth of their maximum connection speeds to avoid easy detection, making the overall amount of attack

traffic approximately 15 Mb/s. Even though the organization's links are able to handle the traffic, the overall network performance is seriously impaired assuming considerable portion of the attack traffic is able to reach the destination. However, in case the attacker instructs the hosts to attack using their full bandwidth capabilities, the resulting traffic of 60 Mb/s would completely saturate the target's networking capabilities, assuming the attack traffic would be adequately divided to both links. Moreover, such an amount of traffic could cause congestion to one or more of the links the target network is directly connected as well. In that case, the attack traffic would not necessarily have to reach the target. Note that this attack is in fact a DDoS attack.

The drastic effect of completely saturating a link is an absolute denial-of-service to everyone dependent of the saturated link. Moreover, bandwidth consumption attacks could last a while in case of a determined attacker. Even attacks lasting several days have been witnessed (The Register, Tue 22nd Jan. 2002). In addition, bandwidth consumption attacks affect the general performance of any link relaying the attack traffic and thus those links and any client or another link dependant of them could be seen as secondary victims of these attacks.

In essence, packet flooding, which is another term for bandwidth consumption attacks, is straightforward to perform but the most difficult to defend against. No additional software to perform these attacks is required, as tools, such as *ping*, provided with modern operating systems enough. The actual attack methods are also simple in logic, as the core function is only a loop creating and transmitting a set of packets. However, when considering the appearance of actual attack traffic and its compatibility with the normal traffic dynamics, the part of creating attack traffic becomes much more difficult. The main aim is to

get a maximum throughput to the attack traffic. Consequently, more valid the traffic appears, more likely it will reach its destination regardless the defences applied and thus more efficient the attack will be. Therefore, the concept of *traffic validity* is crucial.

4.3.1 Traffic validity and attack traffic route

The notion of *traffic validity* can be examined from the views of a single packet and a traffic stream. When viewing individual packets, it appears logical to state that every correctly constructed (error-free) packet is valid. That is, in an error-free packet the header values are correct and they do not contradict with the possible data content. On the other hand, a properly constructed packet could still contain a malicious payload, such as a character string that is either a known or an unknown exploit. This questions the adequacy of the statement that error-free packet is valid. Still, due to the possibility of unknown exploits, malicious packet content cannot be flawlessly detected in real-time. Requiring that a valid packet is both correctly constructed and that it does not have a malicious payload leads ultimately to the realization that in that case the validity of packets is impossible to compute in real-time. Hence, the validity of individual packets would have to be defined by the requirement of containing no construction errors to be of any use, even if the definition would not be complete.

When viewing traffic streams the exact semantics of the validity of individual packets based on correctness lose some significance, albeit it is plausible to argue that every stream made of valid packets is also valid based on what was said previously. For all that, it should be noted that each and every network has its own traffic characteristics, which are the result of all the factors that

influence the traffic flows, such as the user base and their habits, network topology, hardware and so on. Carefully measuring traffic patterns for a predefined period enables various statistics to be calculated from which the characteristics of the network can be observed. In that given context a traffic stream or even an individual packet that does not meet the calculated statistical value boundaries could be treated as invalid. From this follows that the correctness or validity of traffic cannot be statically defined, which in turn means that the concept of traffic validity has to be dynamic and thus situation dependent.

In respect to the argument of situation dependence concerning the traffic characteristics, the concept of *attack traffic route* has to be noted as well. The concept refers to the main route the attack traffic traverses on its way to the destination. It is noted that portions of the attack traffic may traverse different routes; however, the vast majority of each traffic stream is highly likely to take the same route. In any case, it may seem that if the concept of traffic validity were extremely interpreted, an attacker would be required to generate traffic that appears valid throughout the attack traffic route to ensure maximum throughput, which is very difficult. However, as most of the links relaying traffic cannot and do not filter the traffic based on the context, this is not required. In most cases there are only two essential networks to consider for optimizing throughput of the attack traffic route: the source and the destination networks.

The source network of which the traffic is first transmitted to the Internet is relevant, as the owner of the network may have implemented various types of network devices capable of detecting and filtering invalid traffic. For instance, most modern routers provide the functionality of ingress / egress filtering

(Ferguson and Sanie 2000). Thus, carelessly generated attack traffic may fall filtered before it even reaches the Internet.

The destination network is relevant, because it most often is the only source of attempts of filtering the invasive traffic in upstream links in regard to the position of the destination network. In case the attack traffic matches closely the normal traffic patterns in the destination end the attack traffic cannot be clearly separated from other traffic flows. This raises difficulties in applying filters into upstream links, as interfaces relaying the attack traffic are hard to distinguish from others and proper filters cannot be mounted without sufficiently accurate traffic characteristics. Regardless, any intermediate link is highly unlikely to perform drastic traffic filtering without incentives from its downstream links.

4.3.2 Traffic generation in contemporary DDoS attack tools

In contemporary publicly available DDoS attack tools the methods of creating traffic are relatively primitive. Many of the tools fail even the first test of traffic validity, as they fail to create proper packet headers. Slightly more advanced portion of the attack tools create packets correctly, however, some of them using static header values. This type of traffic can be easily encountered with filter rules once the attack is detected. The most advanced of these tools appear to include methods for *heavy header value randomization* and *traffic stream fluctuation*.

The header value randomization can be either complete or controlled. Complete header randomization means randomizing the values of most or all of the header fields without advanced limitations besides what the protocol specification enforces. Controlled header randomization adds limitations to the

randomizing process, such as the restrictions that force the random values to be generated within specified ranges. Generally, h*eader value randomisation* is common in the contemporary DDoS attack tools, but it appears not to follow any advanced mechanism. Some tools randomize every header field whereas some randomize only a few of the fields, such as the protocol type and port numbers, which seem to be the most common randomized fields. Regardless, in all of these tools the header value randomization is done using a pseudo-random number generator seeded by either a constant or some unique value, such as the result of "time" C-library function. Some of these also do restrict the values to a certain range as in controlled header randomization. Those tools that do not randomize the field values may still alternate the values according to some simple logic, such as changing the desired value in every third packet to one of the predefined alternatives.

Based on the observation that the header value randomisation does not follow any advanced mechanism, it can be deduced that the aim was in altering the properties of individual packets instead of traffic flows. While effective against static traffic filters, more sophisticated detection mechanisms could be able to detect data streams with highly randomised packet header values to some extent.

Fluctuating the traffic streams is a method of alternating the properties of traffic streams, such as the transmission rate and burst length dynamically. Alternation of header values belongs to the category as well, but usually not in the form of uncontrolled value randomisation, since completely randomized values may be distinguished in traffic analyses as spikes. Fluctuation of traffic streams is a more advanced approach to masquerade the attack traffic as valid, which is why it is not peculiar that only one of the contemporary DDoS attack

tools provides a method to fluctuate traffic streams. This particular method is based on controlling the burst length. Controlling the traffic bursts or *pulsing* the attack in other words can be an effective way to deceive attack detection mechanisms when coupled with appropriate header value randomization. However, the method will also require more hosts to participate into the attack in order to produce enough of attack traffic, since the delays between the bursts cannot be excessively small to achieve the desired outcome.

Additionally, most of the contemporary DDoS attack tools are capable of various types of source IP spoofing. The source IP can be completely randomised, but such traffic is likely to be identified as malicious rather easily, which is why the source IP would most likely be obscured in some other way. The easy detection is due to the probable result that the randomised IP values are likely to hit address ranges that should not exist, be used or the IP ranges are otherwise reserved for something else. What this means is that there exist unused and reserved IP addresses and as such any packet that has either IP address value set to any of these addresses can be dropped immediately. In addition, service providers can implement ingress filters to reduce source address forging without much of an effort (Ferguson and Sanie 2000). The idea is that any valid packet departing to the Internet must have its source address prefix belong to the network of its service provider. Any other prefix is a positive sign of a forged source address. Nonetheless, these restrictions can be passed by hard coding rules not to use the specified address ranges and forging only the desired prefix of the 4-byte IP source address to bypass ingress filters. These functionalities do exist in some of those aforementioned tools and they can be considered as the most basic capabilities implemented to avoid early filtering.

4.3.3 Defenses against bandwidth consumption attacks

The emphasis is in the fact that bandwidth is the lowest possible level DoS attacks can target when physical attacks are excluded, which is one of the reasons these attacks are difficult to defend. Usually, contacting and making arrangements with the owners of upstream links responsible of relaying the attack traffic and investing to larger bandwidth resources are the only feasible defensive options available. Still, arranging a proper defence with the owners of upstream links may be a difficult task.

First, a signature of the attack traffic is required in order to identify the upstream links that relay the attack traffic (the signature can characterize a single packet or a sequence of packets, as discussed in paragraph 4.1.4). Then, the observed signature will be provided to the owners of the upstream links. Based on the signature, the owners of the upstream links can configure the links to start filtering the traffic matching the signature. The difficulties arise, as the aim is to perform the filtering without overly affecting normal traffic. However, creating an accurate attack traffic signature may not be easy at all, given the possibility of a sophisticated attacking tool and attacker. Excessively drastic traffic filtering is hardly a solution when a reasonably accurate attack traffic signature cannot be established, as it may very well result in DoS due to the filtering of valid traffic. In addition, it should be noted that traffic filtering is an adequate first aid at best with no capabilities of ultimately solving the problem (see paragraph 3.1). Furthermore, it may very well be that the owners of the upstream links are reluctant to start installing appropriate filters for one reason or the other. Lastly, if the bandwidth consumption attack is in fact highly distributed, extra level of difficulty is added, as the same set of procedures would then have to be carried out with multiple upstream link owners. It

should be noted that there are systems such as the ACC that automatically attempt to control the traffic aggregates the DoS attacks against bandwidth are likely to produce (Mahajan et al. 2002). However, in addition to the previously mentioned difficulties concerning the attack traffic, these types of systems require sufficiently widespread installation to be truly usable.

When regarding the issue of tracing the attackers, the task is not any easier. As Savage et al. (2000, 1) pointed out, in today's Internet it is very difficult to trace the source of the attack traffic without cooperation with the owners of upstream links, since the Internet itself does not record any state information. Several tracing mechanisms, often referred as *IP-tracing* or *IP-traceback* have been developed (Savage et al. 2000; Bellovin 2000; Song and Perrig 2000; Snoeren et al. 2002; Goodrich 2002; Dean et al. 2002), but all of these techniques require sufficiently widespread deployment to be usable, which is a problem.

4.4 Possible evolutions

When considering *attacks that target software* not much is likely to change. Software flaws that could be exploited in a way that would result in DoS will be found in the future as well. Due to the nature of this type of DoS attacks the exact attack methods will remain similar.

In a similar manner, it is likely that new protocols will contain deficiencies and flaws that can be exploited as many protocols can be exploited today. Likewise, there will be found more flaws amenable to DoS attacks in contemporary protocols. *Attacks that target protocols* are likely to stay until protocol design reaches extremely high level of maturity. When considering the exact attack methods, fundamentally there is not much to change, as the exploitation always

requires an exact set of procedures unique to the particular attack to be successful.

Attacks that target bandwidth will remain fundamentally the same; however, the attack methods are likely to become increasingly more difficult to be defended. Considering the nature of a bandwidth consumption attack, the evolution in that field can only regard the previously discussed concept of *traffic validity* at key nodes within the *attack traffic route*. Noting that contemporary DDoS attack tools are still relatively primitive in their traffic creation methods, there is still much to develop.

Generally, there are two approaches according to which the attackers are likely to develop their tools. Namely, these approaches are *static estimation* and *dynamic determination*.

Static estimation is about creating estimates of relatively generic traffic characteristics that would be likely to appear valid almost regardless the location. Once such estimates are created the static code to produce this type of traffic generation is easy and fast to implement and does not require much of maintenance effort afterwards. Another question is how difficult it is to create such estimates that are generic enough. In any case, there can be situations where rough estimates of the traffic shape validity are not enough and where accurate detection mechanisms may be able to detect attack traffic generated this way.

Dynamic determination is a more subtle process as it takes the dynamic nature of the concept of traffic validity into account. Dynamic determination requires continuous traffic observation at the key locations in the attack traffic route,

which provides the attacker an exact image of the desired traffic characteristics at any given time. Based on the observations a skilled attacker can create traffic that is extremely difficult to filter. Even though acquiring traffic characteristics of a single network is not necessarily a formidable task, doing so to hundreds or thousands of networks can be very difficult. In addition, the code required to generate traffic based on the real-time traffic observations is much more difficult to implement than implementing the code required in the case of static estimation. The traffic observation and corresponding attack traffic creation would have to be automated to be feasible, which also adds a level of difficulty. The complete automation in turn suggests creating much more powerful DDoS node programs compared what can be observed today. Such programs could be referred as *advanced agents.* The agents would be capable of dynamically observing and analyzing traffic patterns and operating accordingly.

4.5 Summary

In this chapter the DoS attack mechanisms branch was presented. The main points of the different DoS attack mechanisms are summarized into TABLE 1 and TABLE 2.

TABLE 1 presents the main characteristics of every DoS attack mechanism and the main differences between the DoS attack mechanisms. The attack vector column refers to the targets of the attacks that the attacks exploit. The defence mechanisms column lists the main mechanisms that can be used to defend against the attacks. The main effects column portrays the possible damages the attacks may induce. The attack-efficiency column evaluates the ability of the attack type to succeed in causing DoS. The defence-efficiency column in turn

evaluates the applicability of the previously mentioned defence mechanisms in regard to the attack type they are used to defend against.

TABLE 1. The Main Attributes of the DoS Attack Mechanisms.

Attack type	*Attack vector*	*Defence mechanisms*	*Main effects*	*Attack efficiency*	*Defence efficiency*
Attacks that target software	Flaws in software	Intrusion Detection Systems and software patches	System halt or system crash	Excellent without defence mechanisms	Excellent when applied
Attacks that target protocols	Flaws in protocol specifications	Situation specific, no general methods	Service degradation and poor operability	Mediocre; these attacks rarely halt the operability, but they may last long	Situation specific
Attacks that target bandwidth	Network resources	Early traffic filtering in the upstream routers and IP-traceback	Either partial or complete DoS to all dependant of the target	Excellent always when then attack traffic is properly created	From extremely poor to poor depending of the characteristics of the attack traffic

TABLE 2 presents the requirements, phases and specialities of the DoS attack mechanisms. In TABLE 2 the requirements column reflects the attributes that are mandatory for the attack type to be successfully conducted. The phases column in turn displays the necessary procedures that are required to launch an attack. Finally, the speciality column denotes a unique feature of the attack type that is especially important to notice.

TABLE 2. The Requirements, Phases and Specialities of the DoS Attack Mechanisms.

Attack type	*Requirements*	*Phases*	*Speciality*
Remote attacks that target software	Ability to contact the software and a fault or deficiency in the software that can be exploited	Environment setup and exploitation	The possibility of remote and local exploitation
Attacks that target protocols	Flaws in protocol specifications and ability to exploit them	No generic phases	The attack type can be used to achieve three different objectives
Attacks that target bandwidth	The required amount of bandwidth to exceed the target's bandwidth resources	Possibly determination of proper attack traffic shape and continuous traffic generation	The importance of the validity of the attack traffic

5 DISTRIBUTED DENIAL-OF-SERVICE NETWORK MECHANISMS

In this chapter the right branch of the classification named as *DDoS network mechanisms* is discussed. The branch is further divided into four mechanism classes that are *model selection*, *creation*, *coordination* and *additional functionality*.

5.1 Model selection

The term DDoS network was quite loosely defined in chapter two to be a set of hosts that are controlled by the same entity, share the same control interface and which purpose is the administration of DDoS attacks. Ultimately DDoS network is as any other network of hosts, which communicate amongst each other and work for a common goal. The unique aspects of DDoS networks are that some members have to be capable of performing DDoS attacks and every member is controlled by the same static entity. Nevertheless, when considering suitable network models for DDoS the common network theories apply well for the most parts, since the purpose of the DDoS network model is to determine the locations and the relationships of its nodes. The malicious usage, however, does favour traits, which only some network models are able to provide. The dimensions between different network models are numerous, ranging from fast creation to efficient, comprehensive and secure communication.

Since it is not practical to attempt to cover the suitability of every network model to DDoS network administration the focus in this study is only in the four models contemporary publicly available DDoS attack tools use. These models are the *agent-handler model*, *IRC-based model*, *scattered model* and *peer-to-*

peer model. In this paragraph these network models and some of their variations are presented. In the next paragraph the issues regarding the creation of these same models will be discussed.

5.1.1 Agent-handler model

Most commonly DDoS networks are formed using the *agent-handler* model or some slight variation of it. The model is illustrated in FIGURE 2. In essence, the attacker communicates directly with one or more *handlers*, which communicate directly with the *agents*. The agents in turn perform the attacks.

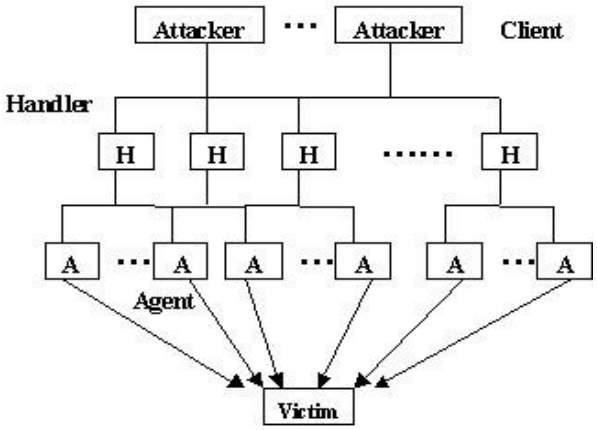

FIGURE 2. Agent-handler model (taken from Spech and Lee 2003, 2)

This hierarchical model is the most commonly used, probably due to its simple design. The ease of implementing this scheme could be counted as one of its greatest advantages. In addition, the hierarchical structure has a few other good properties.

- The message propagation within the network is efficient and rapid.
- The structure is constantly perfectly defined.
- The structure allows easy and fast mobilization of portions of the DDoS network.

With the proper division of roles and distribution of agents amongst handlers, the time variation of message reception between agents ought to be relatively small. Because of that *the message propagation within the network is efficient and rapid*. In principle, the bandwidths of handlers and the number of agents they control are the dominant factors in defining how large is the time variation between the first and the last agent receiving the same message, as each handler has to contact each agent it controls individually. It is because of that the poor selection of handlers as well as division of agents amongst handlers can severely increase the time variation, and vice versa. The time that goes to the communication between handlers and attacker should be insignificant, as the number of handlers is usually substantially smaller compared to the number of agents. Similarly, other factors, such as routing and bandwidths of agents are not supposed to alter the message reception times notably in normal conditions.

Due to the hierarchical design *the structure of the network is constantly perfectly defined*, which in turn gives the attacker complete knowledge of the state of the network at any given time. Among many other benefits, the complete knowledge of the structure and state of the network eases locating the nodes, which might prove to be essential in sudden need of contacting the nodes individually, such as in case of arbitrary errors.

As the hierarchical structure is a recursive concept, DDoS network based on the agent-handler model could be depicted as a collection of smaller DDoS networks. Each of to these organized portions of the DDoS network can be used simultaneously in distinct tasks. In other words, the structure allows easy mobilization of portions of the DDoS network. There is no reason in mobilizing a network of 10,000 hosts against a personal computer.

However, the agent-handler model has a few significant fallacies as well, which undermine its preference for usage.

- Maintenance overhead and increased possibility of identity exposure.
- High handler dependence.

As the attacker is required to contact handlers individually when issuing commands to nodes, some *maintenance overhead* is introduced along with *increased possibility of attacker's identity exposure*. The increased possibility of identity exposure is a direct consequence of having to contact possibly a multitude of handlers (or agents) directly, which increases the number of locations where these messages could be intercepted. There are, however, communication techniques that lessen these risks and some of them will be briefly discussed later in this chapter. Furthermore, these problems could be diminished by adding another handler layer to communicate directly with all the other handlers (*multi-handler level* design) or adding links between handlers so that handlers can relay messages to each other. However, these alterations indicate clearly a fundamental weakness with the hierarchical design; there is a *high handler dependence*.

In other words, the hosts that link different levels together are crucial to the network. The "higher" the link is in the hierarchy the more important it is to the DDoS network. As an example, consider a network that is constructed using the multi-handler level design; the handler of all the other handlers is essential without which the network is unable to operate. The loss of an "intermediate" handler in such network can be tolerated, but the loss may significantly lessen the attack strength of the network. Moreover, the loss of a handler with which the attacker communicates directly may expose the location of the attacker. In any case, intercepting messages between two levels in this design could reveal

important information concerning the DDoS network and actions of the attacker. For instance, inferring a handler location from a control message and consequently inspecting the handler could reveal the identities and locations of all the agents the handler controls. Similarly, intercepting a message from handler of all the other handlers could lead to disabling or seizing the whole network. Lastly, the closer to the attacker the messages go the greater the risk of attacker exposure. From this follows that it is essential to retain the confidentiality and availability of handlers

5.1.2 IRC-based model

Topologically IRC-based model is a derivative of agent-handler model. However, as it has many notable properties due to IRC being the coordination medium that are not present in the basic agent-handler model it is discussed separately. The IRC-based model is illustrated in FIGURE 2.

According to Oikarinen (Oikarinen 1993), “The server forms the backbone of IRC, providing a point to which clients may connect to talk to each other, and a point for other servers to connect to, forming an IRC network.” Hence, IRC network is a collection of IRC servers around the Internet that together form a teleconferencing system. Oikarinen defines “client” as anything connecting to a server that is not another server. A client in IRC network is identified by a unique “nickname” that is at most nine letters long. Oikarinen defines “channel” as a named group of one or more clients, which will all receive messages addressed to that channel. Therefore, in principle, a client connects to a server that relays client's messages to other servers, which host the other clients the message was addressed. These servers then forward the message to the clients they are hosting.

If thought in terms of agent-handler model, the handlers in IRC-based model are the IRC servers that the attacker commands through an IRC server. The IRC servers relay the commands to the agents, which are hosts connected to the IRC servers. The IRC-based model could be seen as a derivative of the basic agent-handler model with the handler depth being two and the first handler level containing a single handler, being the "handler of handlers" or *master handler* in other words. With this handler the attacker communicates. However, it is important to note that it is not restricted in anyway what IRC servers the agents can use. The agents could be connected to the same IRC server (master handler) as the attacker.

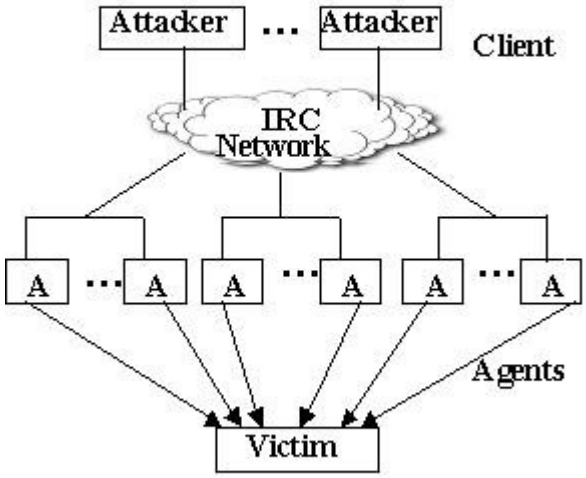

FIGURE 3. IRC-based model (taken from Spech and Lee 2003, 3)

DDoS attack tools use IRC and its channels as a meeting point for agents and attackers. In essence, the attacker implements the minimal features required to communicate appropriately with an IRC server along with the packet flooding functionality into the agents. The DDoS agent in this scenario is a simple IRC client that connects to an IRC server, joins a specified channel and begins waiting for commands. The attacker sends messages to the channel following a

format the agents understand. All agents in the channel receive all the messages sent to the channel and act only in case the message meets the criteria required. For instance, the criteria could be a regular expression matching the nickname of the agent.

DDoS attack tools that use IRC as a communication backbone emerged publicly in 2001. The model has several major advantages over the basic agent-handler model.

- Traffic observation is difficult.
- IRC provides working communication backbone with a lot of functionality.
- Dynamic handler change is an easy and fast operation.

Traffic observation is difficult mostly due to the *usage of common IRC ports* and the *popularity of the IRC*. IRC is definitely one of the most used virtual chatting environments today and therefore traffic destined to established IRC ports, such as TCP 6667-6669 are usually allowed to pass firewalls, routers and other devices capable of traffic filtering. Moreover, due to the global popularity of the IRC the amount of IRC traffic is likely to be quite large almost regardless of the observation point. Pinpointing the few messages attackers send to the channels is overly difficult without some additional information of what exactly to search. Furthermore, intercepting traffic streams known to belong to a popular chatting environment, especially without evidence of malice, brings privacy issues forth. Based on that, it seems that the IRC as a communication backbone is a fine way to masquerade the control traffic of DDoS networks.

IRC provides working communication backbone with a lot of functionality. This lessens the implementation overhead of the agent code to the attacker and

provides plenty of usable functionality. A good example is the channels of the IRC, which make controlling the agents extremely easy, as the attacker is only required to send messages to the channel.

Dynamic handler change can be accomplished by connecting to a different IRC server, which makes it a very easy and fast operation. Changing handlers dynamically makes detecting the presence of DDoS networks more difficult and hence aggravates the network seizure, as the server operators have less time to notice anything possibly suspicious.

However, the IRC-based model has few negative aspects as well.

- Large message propagation area.
- Non-existent protection against network seizure and sensitive information disclosure in case of presence exposure.
- Limited scalability.

As stated in the specification of the IRC protocol (Oikarinen 1993), messages are broadcast and processed at all IRC servers that host any of the clients to whom the messages were addressed. Added to that, clients receive every message destined to the channels they have joined. This type of *message propagation* could compromise information confidentiality and increase the possibility of DDoS network seizure, since there are numerous possibilities where a third party could intercept the DDoS control messages.

As an illustration, consider that a third party has obtained a copy of the agent binary. Executing the binary and consequently capturing the network traffic the agent sends and receives will give the third party information of the identity of the attacker (the nickname and the IP address from which the attacker is

connected to the IRC) and of all operations the attacker commences using that particular channel or channels the agent is connected. Once the third party has learned the details of the IRC servers, channels, channel keys and so on, the third party can seize the network by obtaining control of the channels the attacker uses. Anyone who is able to send messages to the channels is practically in control of all the agents in those channels, assuming the commands the agents responds to are known.

Setting the channel modes “secret”, “key” and “moderated” on (explained in Oikarinen 1993, section 4.2.3.1) is in essence all that can be done to protect the channel from unauthorized viewing, access and usage. In addition, IRC implements a channel mode “invite-only”, which can be used to grant channel access to only those that have been specifically invited to join the channel by their nickname. The downside is that this option forces explicit invitation of each agent to the channel, which is not completely problem-free either. The moderated flag will prevent unprivileged users sending messages into the channel even if they were able to join the channel. Hence, the flag could prevent the loss of control of the agents. However, even in presence of moderated channel, a successful channel intruder receives channel messages normally and thus the intruder might be aware of actions the attacker performs if the attacker commands the participants of the channel publicly. As a minimum measure, it is thus essential to retain the channel key as confidential to maintain some kind of information confidentiality.

The problem with the channel key is that it either must be inserted into the agent code or dynamically transmitted to the agents and furthermore, it must be transmitted in clear text to the server when joining the channel. From this follows that there are multiple possibilities of learning the channel key and

subsequently gaining access to the channel. Statically coded string is easily obtained from the object code and network traffic capture would reveal unencrypted channel key. Thus, it would be preferable that the agent executable would never be inspected by a third party, which suggests that choosing agents for the network is a careful operation to assure the owners of the compromised hosts are unlikely to notice the misuse.

A significant problem the IRC-based model faces is its *limited scalability*. Even though the IRC specification does not state an upper bound for the number of clients a channel can hold, large channels should not be used in controlling DDoS networks. A channel with several thousand participants is likely to raise the interest of IRC operators who are responsible of administrating IRC servers. Dividing agents amongst multiple channels as well as multiple servers is a solution, but raises control overhead. Consequently, very large DDoS networks are hard to administrate using this model.

5.1.3 Scattered model

Scattered model differs vastly from the other contemporary DDoS network models due to its complete lack of intercommunication, which means that the nodes of the scattered network are completely unaware of each other. Even the attacker most often has no knowledge concerning the location of the nodes. Topologically the nodes are arbitrarily located and they are linked only to the target via the attack traffic, otherwise there is usually no other traffic originated. In other words, a DDoS network following the scattered model is build of independent agents. From this follows that coordination of such DDoS network has to be handled through other means, such as using *static instructions* placed

in the agent code or via *"drop-zone" -based* approaches that do not require intercommunication. Scattered model is illustrated in FIGURE 3.

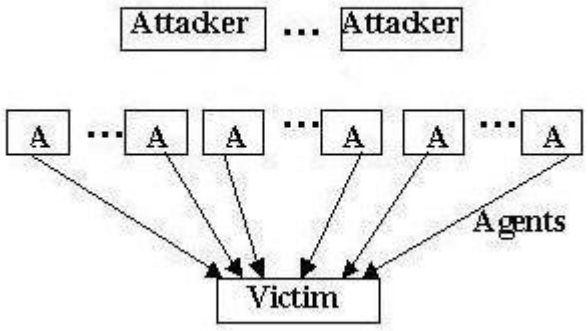

FIGURE 4. Scattered model.

An example of a *static instruction* is a time value against which the system clock is occasionally compared. Other similar methods might rely on specific user actions and other system events to occur. *Drop-zone -based* approaches are active mechanisms, where the agents independently retrieve the commands the attacker has placed into a server (often referred as a drop-zone due to the analogy of other party "dropping" the cargo into a fixed destination for the other to retrieve it). It should be noted that the use of these types of mechanisms has dramatic effects on the usability of scattered model. The term scattered model in this study is primarily used to refer to the "plain" model without any of these additional communication mechanisms. Exceptions will be explicitly stated.

The scattered model has emerged through constantly increasing use of automated intrusion agents (AIA), which are often referred as computer worms. There are very few reasons why attackers would choose this model without automating the intrusions, as will be demonstrated ahead.

To simple purposes the plain scattered model using static coordination appears suitable. Some of the attributes the model has include

- low implementation overhead,
- low risk of disclosing the identity of the attacker,
- significant difficulty of network seizure,
- high scalability and
- rapid network creation with AIA technology.

As the agent of scattered network requires only the attack and intrusion procedures to be implemented, the scattered model has relatively *low implementation overhead. Because of that* the agent executable can be very small, which might be transferable even in a single datagram, as a worm called Slammer was (Moore et al. 2003). In addition, the small executable size is a notable factor in making automatic propagation efficient and difficult to detect.

Since there is no communication between agents and attacker once the agent is operational, there is a *low risk of attacker's identity exposure*. If the agent is capable of self-propagation, the location from which the first agent is launched will most likely be the only link leading to the attacker afterwards. Since there is no communication between the agents either, it is very difficult to decipher the locations of other agents by capturing and inspecting an agent. These agents act independently based on their statically coded objectives. There is no intercommunication present and there are no state information records held. Because of these reasons there is *significant difficulty in seizing or halting the network*, which can make networks following this model difficult to eradicate.
Because of the small executable size and zero need for intercommunication the scattered model has *high scalability* to arbitrarily large networks. During fall 2003 a worm named Blaster demonstrated this by propagating to over 1.4

million hosts (Bailey et al. 2005). From these reasons follows that even notably large networks following this model are *rapidly constructed* using a proper propagation algorithm.

However, to more sophisticated purposes the plain scattered model does not scale due to the complete lack of control of the network. The operations of agents cannot be dynamically defined nor can objective parameters be altered, as there are no communication mechanisms present. Due to the same reason the model has *low usability* and *short life span*, which make it usable to the simplest tasks only.

In case the scattered model is enhanced with a drop-zone based coordination mechanism the attacker can issue orders after the initial launch remedying the greatest deficiency in the plain model. Then again, the use of such coordination mechanisms also undermines almost all of the advantages the plain model has.

- Scalability decreases dramatically, as the drop-zone mechanisms rely on autonomous servers that agents use.
- The risk of attacker's identity exposure increases, as the drop-zones elicit information about the locations where the attacker's actions could be observed or trails to the attacker could be found.
- The possibility of network seizure increases, as the operation of the network relies on these drop-zones.

Based on these observations, it appears the usability of scattered model is limited by the balance of functionality and usability versus scalability. Either the attacker is forced to refrain from dynamic control or the attacker has to implement some kind of drop-zone -based coordination mechanism, which in turn presents a new set of difficulties.

5.1.4 Peer-to-peer model

Much more advanced model compared to the three previously discussed models follows the principles of peer-to-peer (p2p) networks. According to Schollmeier (2001, 1),

> "A distributed network architecture may be called a Peer-to-Peer (P-to-P, P2P,.) network, if the participants share a part of their own hardware resources (processing power, storage capacity, network link capacity, printers,.). These shared resources are necessary to provide the Service and content offered by the network (e.g. file sharing or shared workspaces for collaboration). They are accessible by other peers directly, without passing intermediary entities. The participants of such a network are thus resource (Service and content) providers as well as resource (Service and content) requestors (Servent-concept)."

In short, pure peer-to-peer networks are built of equal nodes that function as both client and server. The equality of nodes or the lack of rigid structure in other words is the trademark of p2p networks and the reason of its numerous benefits. Topologically p2p networks resemble the *scattered* DDoS network model. Unlike the scattered model, however, the p2p model is build upon the requirement of intercommunication. As the definition of p2p implies, the exact semantics of how the nodes intercommunicate are not defined. For instance, in pure p2p networks (Schollmeier 2001) where the network consists only of equal nodes the nodes can maintain adaptive lists of arbitrary sizes of the other nodes to which they directly communicate. In hybrid p2p networks (Schollmeier 2001) there are special nodes that for instance specifically maintain lists of the nodes for the nodes to request them. These issues have high importance to the functionality, usability, efficiency and stealth of the network. The p2p model has much potential as is, but poor design choices can make custom p2p implementations as vulnerable as the basic agent-handler networks. Generally, networks that introduce much responsibility to some of their nodes also

introduce high dependence to these nodes. Coupling specific functionality to only some node equals to the notion that the specific functionality can be removed by disabling the node. From this follows that hybrid p2p networks introduce risks relevant to DDoS networks that the pure model does not. The FIGURE 5 illustrates the DDoS network model following the pure p2p design and the FIGURE 6 illustrates the DDoS network model following the hybrid p2p design.

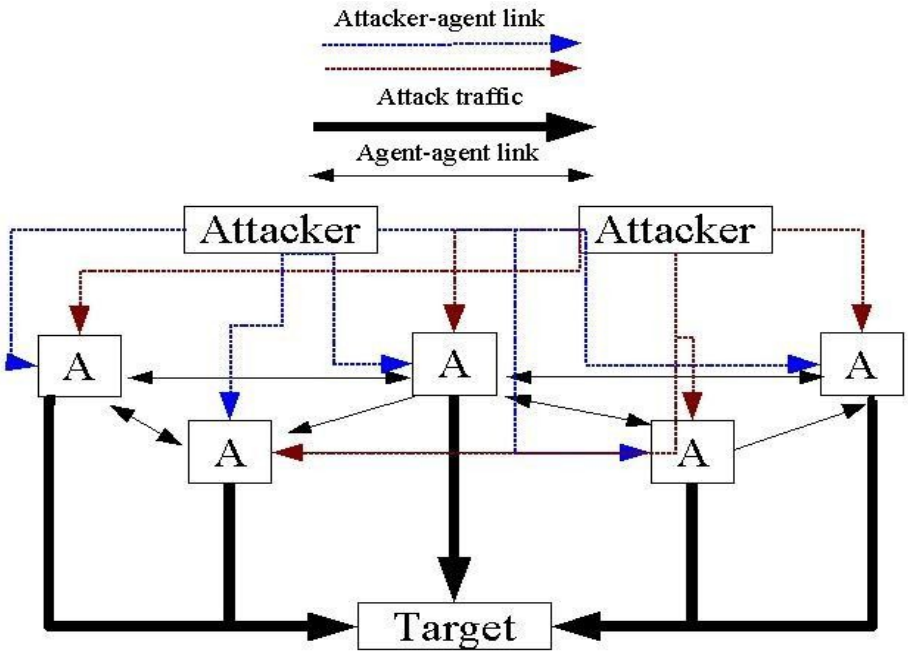

FIGURE 5. Pure p2p model.

Pure p2p networks are based on the links within the nodes that are maintained by the nodes. Every node is required to have at least one incoming and one outgoing link to another node, but besides that everything else is determined by p2p implementations. It is possible that in some pure p2p network a node may have outgoing links to every other node of the network and simultaneously the node may have only the required incoming link, which usually points to the *parent* of the node, if such exists. The parent refers to the node that initialized the particular node, which is the procedure in automated intrusions.

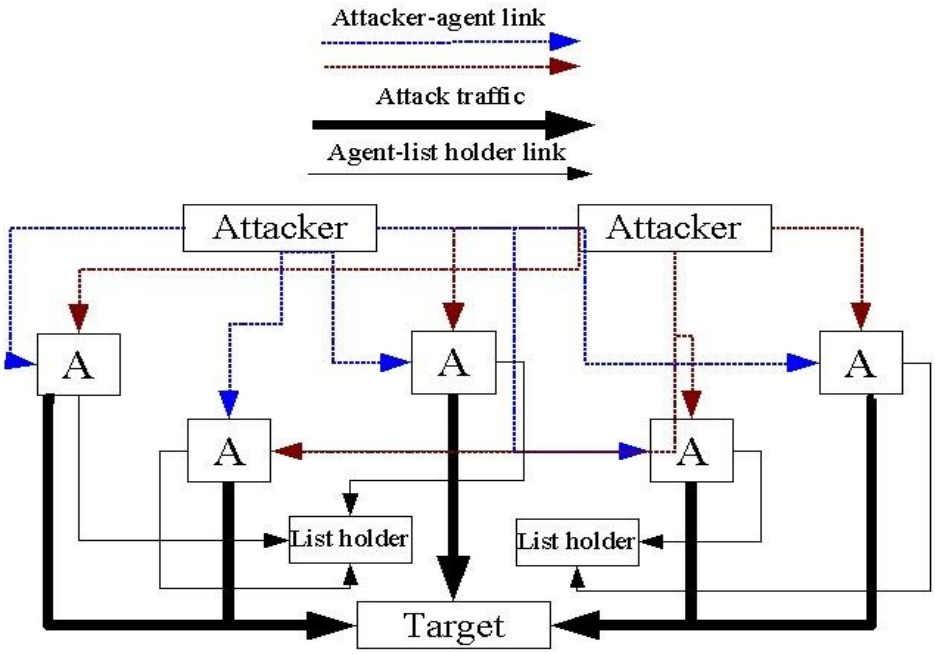

FIGURE 6. Hybrid p2p model.

In hybrid p2p design the links between the agents do not exist, but instead one or more agents are emphasized as the *list holders* the normal agents contact to retrieve location information of the other agents. Because of this at least every normal agent has a link to the list holders.

Of the studied DDoS attack tools only Pud (2002) used the p2p network design as the coordination backbone. This particular implementation followed a derivate of the pure model. The main reasons why p2p model has not been used more often in DDoS networks are probably the novelty of p2p networks and the relatively significant implementation overhead custom p2p networks induce. In addition, it is more difficult to assure message reception amongst all of the nodes of the p2p network compared for instance to the hierarchical agent-handler model. Because of the undefined structure the p2p networks are also difficult to organize, which in turn is one of the main advantages of the agent-handler model. However, p2p model presents a competent DDoS

network alternative due to its many advantages not present in other DDoS network models.

- Commands are easy to inject into p2p networks.
- Peer-to-peer networks present a low risk of exposing the controllers of the network.
- Peer-to-peer networks are scalable, usable, robust, adaptable and difficult to seize or shut down.
- With the AIA technology peer-to-peer networks are fast to create.

The ease of *injecting commands* to the network follows from the equality of the nodes, as each node is capable of forwarding messages to the other nodes, which in turn means that any node of the network can be used as a launch pad for message propagation. From the same reason follows that the messages the attackers exchange with the network are a few in number and the exchange can occur anywhere in the network, which implies that there is a low risk of *exposing the controllers of the network*. Considerate p2p implementations also maintain adaptive lists of the locations of some of their peers, which enable arbitrary addition and deletion of nodes from the network without manual intervention and arbitrary network problems. Because of this the p2p networks are vastly *scalable*, *robust* and *difficult to seize*. The structural freedom in turn makes the *adaptability* and diverse *usability* of the p2p networks possible.

For instance, the hybrid p2p model basically enables the simulation of almost any other network design. Similarly to the scattered model the p2p model is best suited to be used with the AIA technology due to its anarchistic *network structure that is easy and fast to create automatically*. While propagating the automated agents can store information of their parents, children and siblings as well as dynamically transmit coordination data of the other nodes to them,

which in essence creates the p2p network. This is also the main difference with the scattered model where the agents propagate in a similar fashion, but without the active coordination and storing the information of the other nodes.

5.1.5 Summary

In this paragraph the different contemporary DDoS network models were thoroughly discussed. The models define the foundation of DDoS networks and thus the models play an important role in the future use of the DDoS networks. The models have their unique characteristics and they all vary considerably with each other. These differences and the characteristics of these models were discussed in this paragraph and the main advantages and disadvantages of each model are summarized into TABLE 3.

TABLE 3. The Advantages and Disadvantages of the DDoS Network Models.

Model	*Advantages*	*Disadvantages*
Agent-handler	• Rapid and efficient message propagation, • well defined structure and • enables partial mobilization	• Maintenance overhead, • increased possibility of identity exposure and • high handler dependence
IRC-based	• Traffic observation is difficult, • provides a working backbone and much functionality as well as • dynamic handler change is easy and fast	• Futile message propagation, • poor protection against network seizure and disclosure of sensitive information as well as • limited scalability
Scattered	• Implementation is easy, • low risk of identity exposure, • significant difficulty of network seizure, • rapid network creation using AIA technology and high scalability	• Does not scale to sophisticated purposes, • short life span and • generally low usability
Peer to peer	• Low risk of identity exposure, • significant difficulty of network seizure, • high usability, adaptability and robustness, • easy to command dynamically, • rapid network creation using AIA technology and high scalability	• Proper implementation is challenging, • message propagation may be incomplete, • possible difficulty of achieving complete network control and • difficult to organize

5.2 Creation

As discussed in the previous paragraph, the selected network model has a wide impact on the functioning and operability of DDoS networks, including the creation process. The selected network model closely relates to the processes the network creation must and must not and can and cannot involve. In this paragraph the creation of the four previously introduced DDoS network models are discussed. The creation process consists of multiple phases regardless of the DDoS network model. These phases are discussed in this paragraph. It should be noted that these phases are not present in all DDoS network models and they do not need to follow any rigorous order. First, however, a few words about the ownership of hosts comprising DDoS networks and then a few notes about the main differences concerning manual and automated DDoS network creation.

5.2.1 Ownership of hosts comprising DDoS networks

It was mentioned in the introduction that the security of the Internet is interdependent. With DDoS attacks this notion is especially confirmed, since DDoS attacks often are performed using compromised hosts, although it is possible that a group of people either voluntarily agrees to join into an attack or otherwise allows the attacker to enter and use their machines. It cannot be excluded either that the attacker is the legitimate owner of large number of hosts or has the resources to acquire the required hardware, which could very well be true in case of large organizations, well-funded terrorist groups or even independent countries. The requirement to compromise additional hosts is evident when the attackers do not legitimately possess the required resources, which most often is the case with individuals or small groups.

Even though no public research regarding the usage of compromised hosts in illegal activity was found, it can be assumed that a significant number of the DDoS attacks in the Internet today are executed by using compromised hosts. This assumption is based on the observation that there are numerous reports of DDoS attacks being performed by crackers and script kiddies (gathered by Dittrich, shown on his web-page), whereas no mentionable public evidence of DDoS attacks that could have been performed by some other previously mentioned perpetrators was found. Still, the possibility of someone employing a cracker group to perform DDoS attacks is real (for instance, see Poulsen, SecurityFocus Aug. 26 2004). In any case, it is unlikely that these groups possess enough of resources on their own, which suggests that most of the DDoS attacks in the Internet are initiated using compromised hosts. This in turn emphasizes the notion that the security of the today's Internet depends of factors of which no single entity has any control.

5.2.2 Manual versus automated DDoS network creation

The actual host compromising process can be manual, semi-automatic or automatic. As compromising hosts manually is tedious process, several automation methods have been witnessed since the first publicly available DDoS attack tools. These automation methods range from simple scripts that scan vulnerable hosts to programs that automate everything leaving either no control or only the attack management to the attacker. The programs that automate everything are examples of using automated intrusion agent technology to form large DDoS networks rapidly.

The manual DDoS network creation has its advantages in comprehensive control of what is being done at any given time. The chosen network model can

be followed to the letter and even specific hosts with desirable properties can be included to the network instead of others less suitable. The advantages of choosing which hosts are included into the DDoS network and specifically into which roles could be significant. For instance, it is notably easier to prefer rarely inspected hosts or hosts with high-speed connections to the most important roles of the DDoS network. Hence, manually created DDoS networks can be accurately customized to fit into desired parameters, which again may be situation dependant. The difficulty is that the work overhead may be overly high, which implies that manual DDoS network creation does not scale. Furthermore, there are larger risks of identity exposure in case of being noticed by a careful system administrator or only being careless. Automated intrusions are usually difficult to trace back to the attacker.

The automated DDoS network creation is a step towards much larger DDoS networks, significantly faster creation process and diminished possibility of identity disclosure compared to the manual creation process. A notable disadvantage is the previously mentioned lack of adequate control of host and possible host role selection. As these procedures normally involve human reasoning to some extent, they may be incomputable by nature and thus the implementation of proper automation algorithms may be excessively difficult. It is also more probable that the automated intrusions and intrusion attempts will be detected, which might present an unacceptable risk to situations that have a zero tolerance for exposing the creation of DDoS networks.

Still, it is reasonable to combine the best aspects of both automated and manual creation mechanisms. For example, the host gathering process could be automated once the IP ranges of the networks where the hosts will be compromised has been determined. In essence, this type of hybrid creation

processes require manual intervention only in issues that require special precision or human reasoning and leave every other processes to be automated.

5.2.3 Gathering of hosts

The phase of gathering of hosts is common to all four previously discussed DDoS network models, assuming the attacker has to compromise hosts in the first place. The exact technical details of compromising hosts are irrelevant for the purposes of this study. For the purposes of this study it is enough to state that the attacker's aim is to gain preferably an administrator level access into each host in order to gain full control over the entire network of hosts. In addition, there is significance whether this phase will be performed either automatically or manually in respect to the DDoS network model.

Even though this phase can be either manual or automated in all four models, there is minimal sense in compromising hosts manually when the chosen network model follows the scattered or peer-to-peer design choices. This is especially true in plain scattered networks, since the usability of such networks depends on large size and fast creation (see subparagraph 5.1.3).

On the other hand, the gathering of hosts to form agent-handler and IRC-based networks may require prudence due to the reasons stated previously (see subparagraphs 5.1.1 and 5.1.2). This in turn is likely to involve human reasoning in some form, which would make proper automation difficult to implement. Hence, DDoS networks following agent-handler or IRC-based models may be difficult to create automatically while retaining the benefits of these models.

5.2.4 Deciding host roles

Normally the gathering of hosts is followed by a phase of *deciding host roles*, assuming the chosen network model supports multiple roles. Generally, the set of different roles in DDoS networks reflects the intent of organizing the network and yielding different tasks to different hosts. For instance, in the agent-handler model the attacker chooses which hosts will be labeled as handlers and which as agents. In the IRC-based model the selection of IRC servers could be thought to belong to this phase as well.

The difficulties of automating aspects of DDoS network creation that involve human reasoning have been briefly mentioned before. The phase of host role selection is a fine example of a procedure facing such difficulties. Considering that the handlers are likely to communicate directly with multiple agents and that the identities of agents are known to the handler it would be preferable that the hosts would belong to people that do not know or care much about their computers where the risk of exposure is the smallest. Perhaps this could be deduced from the configuration, the ease of exploitation or the location of the host. Similarly, it could be desirable to place handlers to positions that meet desired properties, such as fast links to some of the agents or the probability of attack traffic taking a desired route. Although these properties are programmable to a certain extent the work overhead and resulting code size may be too large to ultimately guarantee nothing.

In scattered and peer-to-peer models all nodes are equal and thus conventionally this phase is not present in them. Breaking the convention would break the design and even though the roles could provide some additional value to the operations of these network models, the broken design

would most likely lose some of the most important benefits these models have as well.

5.2.5 Preparation of hosts

A common phase to every DDoS network model is the *preparation of hosts*, which in essence is a matter of transferring and installing the required software, such as the agent executable to the compromised hosts. In essence, this phase includes every operation the attacker might perform as a part of setting the host up for later usage. These operations may include installation of other malicious software, alteration of log files and concealment of the presence of the attacker as well as the installed software.

In the conventional agent-handler model handlers will be provided the handler software and lists of agents or other handlers they control and agents will be provided the agent software accordingly. In the rest of the models the procedure is slightly simpler, as it is only required that copies of the agent are transmitted and successively initiated in all hosts.

5.3 Coordination

When viewing the properties of DDoS attacks more closely the aspects of *coordination* and *multi-home behaviour* get involved. Multi-home behaviour means that there is more than one participant in an attack. Coordination means that the participants either can "self-organize" or that they can be organized to initiate an aggregated assault to the chosen set of targets. Self-organising means that the participants can organize their shared objectives without a controlling entity.

Coordination is an absolute part of DDoS and it is about different nodes in the DDoS network exchanging relevant information with each other. For instance, an attacker might issue orders to some of the nodes in the DDoS network, which in turn might command each other and finally return information such as how the attack is progressing to the attacker. In essence, coordination of hosts is a requirement for a purposeful function of DDoS networks. However, proper coordination may not be an easy task to accomplish, since there is multitude of attributes that should be considered. For instance, proper coordination mechanisms should maintain the secrecy of the identities of the attackers. Coordination that is built upon a communication mechanism that transfers data in plain text to statically defined unreserved communication ports is much easier to detect compared to a mechanism that transfers the data encrypted in images using steganography. Moreover, inadequate coordination mechanisms may lead to inefficiency, possible inability to meet the set objectives in case the mechanism is not robust enough and even seizure of the DDoS network, which may occur in case a third party is able to detect and decipher critical information sent from one node to another.

The exact attributes of coordination are defined by a combination of low-level technical *communication mechanisms* and *DDoS network model*. The DDoS network models were already discussed and the communication mechanisms determine the technical details of transferring coordination data amongst the nodes through the routes the DDoS network models define at an abstract level.

Overall, there are numerous possibilities for appropriate coordination; however, many of the network models and communication mechanisms have their deficiencies that could possibly compromise the confidentiality of DDoS networks. The best coordination method is a combination of a network model

and communication mechanisms that together form the most secure way of coordination considering the objectives of the network and the particular environment of operation. Good technical methods for coordination do not assure much as long as the design of the network is inadequate and vice versa.

In this paragraph several mechanisms and methods used in communication are discussed. Some of the similar communication methods discussed here have been briefly mentioned by Nazario et al. (2001, 16-17)

5.3.1 Direct and instant communication

Direct and instant communication mechanisms are based on normal socket programming and they form the set of the most basic communication methods. This is also the most common communication mechanism used in contemporary DDoS attack tools. As a method belonging to this mechanism class, many DDoS attack tools use *predefined and unreserved communication ports* and *predefined protocol types* as a communication method. In essence, the DDoS software uses one or more unreserved communication ports for incoming connections, which are used for desired purposes, such as passing commands or situation reports to the node or from the node to some other node.

For example, Trin00 used UDP 27444 for handler-to-agent-, UDP 31335 agent-to-handler- and TCP 27665 for attacker-to-handler communication. To control handlers attacker connects to the specified port and receives a minimal interactive command prompt to which attacker can issue a few different commands, such as one that instructs the handler to order the agents it controls to attack a specified target. In turn, communication between handlers and

agents are operated through connectionless channels using predefined messages expressed as strings and predefined port numbers. (trin00 1999)

This type of communication method is straightforward to implement and due to the use of normal sockets and unreserved communication ports administrator-level privileges are not required. In addition, direct communication is instant and when it is also connection-oriented the communication is reliable and efficient as well. However, a significant downside with any communication method relying on predefined communication ports is that the traffic destined to those ports is easily detected and consequently blocked especially once the purpose of such traffic is known. From this follows that agents and handlers using this type of communication methods can be disabled and removed. The method could be slightly improved by using *reserved* and highly *common communication ports*, such as the http, which normally is bind to port number 80. In many situations the traffic to these ports is considered as benign and thus the traffic do not stand out as notably as the traffic to more peculiar ports does. The minor downside is that the binding of reserved communication ports in modern operating systems requires administrator privileges. None of the studied DDoS attack tools used reserved communication ports as default, only the Blitznet handler (Blitznet 1999) had the possibility of defining which port to listen at runtime.

Although the use of predefined communication ports is not favourable when stealth is an issue, the use of predefined protocol types may be favourable. For example, encapsulating data into ICMP Echo-Reply packets in a similar way the DDoS attack tool TFN does is relatively fine way of masquerading the communication data (Tribe Flood Network 1999). In TFN the predefined command was represented as an integer inserted into the ID header field of

ICMP Echo-Reply packet. Since ICMP is highly used and useful protocol for network diagnostics and error reporting (RFC 792 1981), DDoS network control data encoded into ICMP packets is difficult to detect. Moreover, ICMP Echo-Reply is a response to ICMP Echo-Request, which is often used as a tool for network diagnostics and due to that networks may not block ICMP Echo-Reply packets.

5.3.2 Direct, instant and stealth communication

The greatest deficiency with the plain direct communication in DDoS networks is the complete lack of stealth. Augmenting the plain direct communication with additional mechanisms (see subparagraph 5.4.2) such as encryption to protect the transferred data does not ultimately enhance the mechanism much, since the communication can still be detected and blocked as easily as before. However, with the help of modern operating systems direct and instant communication can be made stealth as well.

Slightly more evolved DDoS attack tools, such as TFN2k were incorporated with the functionality of using *undefined communication ports* as a communication method (Tribe Flood Network - 2k edition 1999). Instead of hard wiring protocol types and port numbers these methods decide the values dynamically either at start or runtime. For instance, TFN2k is capable of randomly selecting the communication protocol amongst TCP, UDP and ICMP and randomly generating header values to the selected protocol for each packet. The actual messages are encrypted to the packet's payload. In such scenario the recipient cannot bind communication ports to listen for incoming connections and thus the recipient does not have its own network data queue to read. From this follows that the recipient is forced to read every packet the system receives.

This usually is not a problem, since with modern operating systems using modern network interfaces all data received through the network interfaces can be obtained regardless of the process to which the received data was meant. Thus, a TFN2k agent with required privileges can inspect the packets the system receives to check the presence of an identifiable payload. However, inspecting all of the network data the system receives is inefficient.

The advantage of encapsulating control messages into dynamically defined packets is significant, as only the payload of the packet indicates the purpose of the packet. In case the payload is further encoded and encrypted the possibility of third party intercepting the exchange of DDoS network control messages decreases dramatically, since without knowing the encoding and the cipher the payload cannot be inspected. This is the method that was attempted in TFN2k (Tribe Flood Network - 2k edition 1999). However, Barlow and Thrower (2000) noted that the implementation of encoding procedure was flawed in such a way that it could be used to detect coordination traffic in TFN2k network.

As with the communication using predefined communication ports and protocols, communication based on undefined communication ports and protocols is instant, but not as reliable. The reliability is decreased, since no real connections between the communicating nodes are formed and due to the possibility that alternating the properties of packets could cause more firewall collisions and consequently packet loss. Because of these reasons two-way communication is also more difficult to implement.

5.3.4 Indirect, instant and stealth communication

The coordination traffic can be further obfuscated by the use of indirect, instant and stealth communication mechanisms, which rely on *controlled forgery of source and destination addresses*. In principle, no direct connections between communicating parties are formed, as the messages the parties exchange do not go to the real destinations as such. In that way, it is very difficult for a third party to learn the locations of the participants of the DDoS network only by observing network traffic.

Forging source address is a clean way of obscuring the actual location of a sender to certain extent. Similarly, the destination address can be forged to conceal the real location of a recipient. Usually both data-link and network layer addresses can be forged. In regard to coordination traffic, only TFN2k of the contemporary DDoS attack tools forges the source network layer address and none of the studied DDoS attack tools forge the destination network layer address. None of the contemporary DDoS attack tools forges the data-link addresses.

A problem with forged addresses is that the recipients have to be able to observe or "sniff" in other words the traffic of the network where they are located. This also restricts the forged addresses to belong to the network segments where the recipients are located, which in turn obliges the communicating parties to resort to *controlled IP address spoofing*. Another problem with forged addresses is the increased difficulty of engaging into two-way communication if such is required, as without further measures the other party do not know the real location of the other and thus cannot reply. There are various methods to overcome the difficulty including *controlled IP address*

spoofing, *recipient address encoding in control messages* and *constant knowledge of the location of the recipient*. None of the studied DDoS attack tools used forged source addresses with two-way communication methods.

Controlled IP address spoofing is a method in which only the desired portion of an IP address is forged. As mentioned, this can be adopted to circumvent egress and ingress filters, but it can be used in stealth communication as well. Forging the IP address to be within the Ethernet segment the receiver is located enables the receiver to capture the message regardless the forged address. Even in switched networks a method generally called as ARP-poisoning (see for example Whalen 2001, 3-4) can be adopted, which for instance can be used to capture network traffic that is not destined to the monitoring host. Controlled spoofing is a requirement when the destination's IP address is forged to enable the receiver to capture the transmitted data. In this case the destination's forged IP address is still required to be within the network segment where the receiver is located. As mentioned, controlled spoofing is also a possibility when two-way communication is attempted. In this case the sender can forge its IP address to be only within the network segment where the sender is located. Based on this address the receiver can deduce the network segment to which the response should be sent. An idea that has some similarities to controlled IP address spoofing was presented by Simple Nomad (Simple Nomad 2001, 2-4).

Encoding recipient address into control messages is another method that can be used in coordination traffic when the source address field in the IP header is false. Using this method the address to which the host should send the response is encoded into the control messages in some form. However, as a stealth method this is not as efficient as controlled spoofing, since in case the real destination address is being used the real location can be learned by observing

where the host is transmitting traffic. However, this method could be used with controlled spoofing and as a method of dynamically defining the recipient.

Finally, the addresses to which hosts are supposed to respond could be statically coded to the node executables or configured via initialization parameters as well. Thus, *constant knowledge of the location of the recipient* is a possible method, even though more limited than the other two.

Even when two-way communication is not implemented and generally not needed it is often important to ensure proper message reception. Messages can be lost in transit, especially in case the receiver is using a notable amount of its bandwidth capabilities, which is likely to increase the amount of packet loss. A method to this is to *increase the probability of message reception*, which can be accomplished by transmitting the message more than once. This method was implemented in TFN2k, which sends each control message twenty times (Tribe Flood Network - 2k edition 1999).

5.3.3 Indirect, delayed and stored communication

Indirect, delayed and stored communication mechanisms are usually referred as drop zone -based communication mechanisms. The term *drop zone* in turn is used to refer to a place where one side stores information for others to retrieve it irrespective of time. The easiest setup of a drop zone -based communication mechanism in a DDoS network is a set of servers placed by the attacker, which addresses are coded into the DDoS agents. The agents then check the servers for new instructions in predefined or random intervals. None of the studied DDoS attack tools use drop zones as a communication mechanism.

Probably the greatest advantage of drop zone -based mechanisms is the complete indirectness, which means that no direct links between the communicating parties are formed. Due to the indirectness the communicating parties cannot determine the identities of each other only by observing the exchanged data and traffic flows. The delayed communication can also be an advantage, since the nodes can initiate communication attempts independently when it is considered suitable.

The disadvantage of this is that drop zone -based mechanisms cannot be used in time critical circumstances. The drop zone -based mechanisms also underline the greatest deficiencies of any mechanism that concentrates communication data to a small number of nodes. First, any such mechanism does not scale to large networks due to the limited bandwidth of small serving node set. In addition, large amount of traffic destined to a few hosts may indicate activity worth observation. Second, the operability of the network depends on the small number of serving nodes, which can make the network relatively easy to disable. Third, in case any of the serving nodes is exposed, every node connecting to it is under a risk of identity disclosure.

5.3.5 Public mediums

As a mechanism for stealth communication, various widely used public services may prove to be suitable, as the communication data can be easily merged into the masses of other similar traffic. There is vast number of different public mediums available that can be used in different purposes. The already discussed Internet Relay Chat (IRC) is one such medium (see subparagraph 5.1.2) and it is the only public medium used by the studied DDoS attack tools.

No evidence was found that any other public medium is being used as a coordination mechanism in DDoS attacks.

However, there are many possibilities, as any public or private forum, chatting or messaging environment or anonymous email could be used. In particular, every communication medium with large user base is a considerable option, as the messages of these environments can be concealed into the bulk of the other messages.

For instance, Usenet is a worldwide collection of newsgroups that are categories of various kind under which users can send messages relating to that particular subject the category belongs For more information see for instance Moraes (Moraes 1999) and Erickson (1993, 1-2). Today Usenet consists of thousands of servers and it is one of the largest actively used messaging environments, which is why using Usenet's newsgroups as drop zones for malicious activity can be a fine stealth communication method. If the posted messages are constructed well the purpose of using the messages as a stealth communication method may stay unnoticed. As an example, posting images along with some casual message content and applying steganography to hide coordination data to the images could be a fine way to masquerade the coordination traffic (for more information about steganography see for instance Johnson and Jajodia 1998).

The advantages and disadvantages of the used public mediums are dependant of the used medium. For example, mediums that function as drop zones share some of the advantages and disadvantages of the indirect, delayed and stored mechanisms. Similarly, mediums that transmit communication data straight to the participants share some of the properties of direct and instant communication mechanisms. However, many public mediums have the unique

characteristic of being owned or controlled by some other entity besides the users of the medium. Because of this some public mediums may be administrated, some may store the posted messages for an arbitrary amount of time and some may route the communication data through the servers of the provider of the communication medium. From this follows that inconsiderate actions are more likely to be noticed by someone, which in turn raises the risk of exposing the DDoS network activity and the identities of the participants of the DDoS network.

5.3.6 Static instructions

Static instructions as a communication method refer to predefined coordination data hard coded into the network node executable of interest. The communication in this case is the most simplistic, as it involves only the original programmer of the node software and the running executable. Static instructions are fast and simple to implement and they do not exhibit the possibilities of attacker's identity disclosure or seizure of the DDoS network. However, as it was discussed in the subparagraph 5.1.3, the usability and life span of such nodes and consequently the usability and lifespan of the network built of these nodes are low.

5.3.7 Summary

In this paragraph the coordination in DDoS networks was discussed in detail. The discussion showed the importance of appropriate coordination in DDoS networks and presented the different coordination mechanism classes in depth. A significant portion of the discussion was focused in detailing the advantages and disadvantages of the coordination mechanisms. These discussed

advantages and disadvantages of the different communication mechanisms used in DDoS networks are summarized into TABLE 4.

TABLE 4. The Advantages and Disadvantages of Different Communication Mechanisms.

Communication method	*Advantages*	*Disadvantages*
Direct and instant communication	• Simple to implement, • administrator privileges are not required and • instant, reliable as well as efficient data transfer	• Relatively easy to detect and block
Direct, instant and stealth communication	• Hard to detect, • difficult to block and • instant data transfer	• Slightly unreliable, • slightly inefficient and • proper two way communication may be slightly difficult to implement
Indirect, instant and stealth communication	• Hard to detect, • difficult to block and • partial indirectness obscures the locations of the communicating parties	• Implementation overhead, • inefficient • slightly unreliable and • proper two way communication may be considerably difficult to implement
Indirect, delayed and stored communication	• Complete indirectness and • nodes can search as well as provide coordination updates independently	• Limited scalability, • cannot be used in time critical situations and • tight coupling with the operability of the network
Public mediums	• Medium dependant	• Medium dependant and • may require additional diligence
Static instructions	• Does not expose information of any other nodes and • does not expose the identity of the attacker	• Usability and lifespan of the nodes and of the networks built of the nodes are low

5.4 Additional functionality

In contrast to the DoS attack mechanisms, DDoS network mechanisms are not as categorical by nature and due to that they could be categorized in several different ways. Although the previously presented DDoS network mechanism categories are always present in the construction and in the consequent usage of DDoS networks, these categories do not cover every mechanism that could be

part of the DDoS network mechanisms. Many of the DDoS network mechanisms cannot be generally categorized, since they are not requisites for the operability of the DDoS networks. However, these mechanisms may still provide additional value to the operations of the DDoS networks. Because of that the category of *additional functionality* was included.

Additional functionality is a large class, since in principle it includes every mechanism that cannot be categorized to the other three classes but that adds some additional functionality to the DDoS networks. In this paragraph a few of the additional mechanisms the studied DDoS attack tools include are discussed.

5.4.1 Update mechanisms

The function of the update mechanisms is either to update the entire node executable or some specific features of the node executable. In this context the update mechanisms do not refer to the commands that update the attack or coordination parameters the nodes dispatch to each other.

A major deficiency with the contemporary DDoS networks is the lack of dynamics in operations besides the basic attack and coordination parameters, such as the verbosity level of reporting and the type of generated attack traffic. The lack of dynamics decreases the usability and the lifetime of the network. For instance, when the flaw in the coordination traffic of TFN2k networks was detected (Barlow and Thrower 2000) the owners of TFN2k networks could have applied a patch to correct the flaw, if TFN2k would have had an appropriate update mechanism. The update mechanism could have used the existing DDoS network structure to automatically propagate the update to the handlers and to the agents and hence correct the flaw that ultimately enabled relatively easy

detection of TFN2k coordination traffic. From the studied DDoS attack tools only Knight (2001) had the option to update the node executable. None of the studied DDoS attack tools was designed in a modular way that would have enabled the dynamic update of modules responsible of specific functionalities.

5.4.2 Stealth mechanisms

Besides the actual communication mechanisms that eventually are responsible of exchanging the coordination data, there are several additional mechanisms that can be mounted for instance to increment the stealth of the communication.

Encryption is one often used mechanism, as the overhead the implementation and the use of encryption introduce is small compared to the security the encryption provides. For that reason many of the publicly available DDoS attack tools, such as TFN2k (Tribe Flood Network - 2k edition 1999), Stacheldraht (2000) and Trin00 (1999) used various encryption methods years ago. However, it has been studied that the bit entropy of encrypted data differs notably from unencrypted data, which enables the encrypted data to be detected (Shamir and Someren 1999, 5-6). From this follows that the encrypted network traffic might stand out from the traffic streams when evaluating the bit entropy, as was noted by (Nazario et al. 2001, 16-17). To obscure the presence of encryption, proper traffic encoding is a minimum measure to enable the encrypted traffic to fit better into normal traffic patterns to avoid anomalies in bit entropy as much as possible.

Another additional stealth method is to emit *decoy traffic* along with the real coordination traffic to disguise the coordination traffic to the traffic streams more efficiently. Typically the decoy traffic would be like the real coordination

traffic, but with completely bogus or missing payload. Only TFN2k of the studied DDoS attack tools used decoy traffic (Tribe Flood Network - 2k edition 1999).

More intelligent stealth mechanisms could be tailored to consider the locations of the communicating nodes and traffic dynamics of those locations in detail. For instance, the communicating nodes could initiate communication when a notable increase in the amount of network traffic is detected. Furthermore, the communicating nodes could tailor the packets according to the other observed network traffic.

5.5 Possible evolutions

In this paragraph a few ideas regarding possible evolutions in DDoS network mechanisms are discussed.

5.5.1 Derivatives of agent-handler model

The ideas of *multi-handler level design* and *linking handlers horizontally* amongst each other were briefly mentioned earlier (see subparagraph 5.1.1). Even though all of these publicly available DDoS attack tools lack this type of functionality, the functionalities could be implemented and they might be even preferable. Linking handlers has the benefit of reducing higher level messaging, but at the same time it increases the chances of collateral damage (see subparagraph 5.1.1). Therefore, it does not seem as a good option. However, considerate balance between right *depth*, which refers to the number of handler levels and *width*, which refers to the possibly varying number of handlers at each level may be beneficial.

The *depth* could be increased by adding handlers of other handlers. The increased depth would lessen the attacker's need to contact multiple handlers as well as increase the difficulty of tracing the attacker's approximate location. However, the previously mentioned risk regarding the *handler dependence* should be carefully considered. In a similar fashion, all agents could be directly controlled by a single entity, but this also increases the dependence of the controller and the risk of exposing the controller's identity due to the raised level of coordination traffic the controller transmits. The *width* could be increased by adding more handlers to the same level in hierarchy. That is, the handlers belonging to the same level would be controlled by the same entity, whether it would be another handler or the attacker. The wider the handler levels the less collateral damage will occur depth-wise in case of handler loss, as each handler would control smaller number of agents or other handlers. However, in case the model is strictly followed handlers do not participate in attacks, which will mean smaller number of attacking agents and thus less attack strength.

5.5.2 Enhancements to IRC-based model

As stated in subparagraph 5.1.2, one significant disadvantage of the basic IRC-based model is its inexistent protection against network seizure and sensitive information disclosure in case of presence exposure (see subparagraph 5.1.2). The clear text channel messages form a great risk, which is why private program code and *encrypted channel messages* would lessen the risks of information confidentiality breach and network seizure. With these measures channel intrusion and channel takeover would not be enough to disclose the coordination data sent to the channel. In addition, instead of using only channel messages for coordination, *private messages* (see Oikarinen 1993, 32) could be

considered. However, the use of private messages would require each agent to be validated with a secret of some type to be added to the controller's private message list. With the use of validation scheme and private messages no channels would be required, which in turn would make the controller the only entity that has the knowledge of all participants of the DDoS network. Without the validation procedure channels would be required for the controller to enumerate the members of the private message list, which would reduce the use of private messages to be similar to the use of channel messages.

The property of being undetected is important in every DDoS network model, however, on the contrary to the other discussed DDoS network models the IRC-based model is built on it. Even with encrypted and private messages the disclosure of the DDoS network does nothing to prevent the owners of the agents or the IRC operators being contacted. To decrease the probability of detection, *automated IRC server/network cycling* could be mounted to hide the presence of DDoS network better due to the constant movement. Attackers could *create their own IRC-servers* as well. Linking own servers to IRC network can be done and when the server is otherwise appropriately administered, it would be unlikely anyone would notice the other use of the server. As a relatively advanced and complex measure, an arbitrary *agent-handler hierarchy could be created* into the IRC-based DDoS network as well. For example, in such scheme channels could reflect the levels of the hierarchy.

5.5.3 Advanced agents and agent networks

Generally, the contemporary tools specifically designed for DDoS attacks are relatively unsophisticated in terms of DDoS network mechanisms. The vast majority of the tools rely on manual compromising of hosts and inadequate

communication mechanisms. Similarly, the same tools have very limited functionality and they lack the ability to operate independently.

The past few years have marked a new era of computer security threats, as automated intrusion agents (AIA) or computer worms in other words have gained increasingly more ground as a notable threat to any computer connected to a network of some kind, especially to the Internet. Although large networks of automatically compromised hosts have been created, these networks and the agents behind the intrusions are relatively primitive in regard to what these networks and agents could be. In broad outline, the AIAs so far have been based on four main principles.

First, they attempt to *infect* potential targets. Second, they attempt to use the successfully infected machines to *propagate* further, usually as fast as possible. Third, if they do have some other functionality implemented, *the functionality has so far been statically inserted* into the worm and cannot be modified without replacing the whole worm instance. Fourth, *they all are monolithic* in nature, which mean that all AIA instances are exact copies of the original worm. These principles alone dictate heavy restrictions over the usability, controllability, agent capability and lifetime of these agents. Certainly, there are many more flaws in contemporary AIAs, some of which Nazario et al. (Nazario et al. 2001) eloquently addressed. However, many of these flaws are technical and thus cannot be separated as clearly as the four more paradigmatic principles.

Already 2001 Nazario et al. (2001, 13-20) theorized the idea of "future worms" or advanced automated intrusion agents (AAIA), but still it appears such agents are yet to be witnessed. On the contrary to traditional agents, advanced agents do not necessarily have a static purpose and they do not have to be monolithic.

Their capabilities can be extended dynamically, they can distribute tasks with other worm instances, they can have different roles within the agent network and most importantly, their objectives can be briefed dynamically. Advanced agents can dynamically learn about the environment they are in and adjust their behaviour accordingly. They can be dynamically augmented with new functionality. They can work cooperatively with other advanced agents for a common objective, such as compromising hosts A, B and C, but in a manner that would not expose their presence. They can go hibernate in case their services are not needed and be revoked again when desired. They can even terminate themselves if they determine the network too hostile to live and hence the risk of exposure and capture too great.

The advanced agents have vast potential, since they are not dictated by the four primal attributes of the traditional agents, but instead their purpose could be to operate as cooperative stealth agents in more fine-tuned and dynamically defined objectives. In principle, it is only the imagination and skill of the controller of the advanced agents that limit their potential. As a result of this, advanced agents could be used to create DDoS networks of arbitrary size with the ability to adjust the attack parameters dynamically and independently.

For instance, the advanced agents could first analyse the locations they are and based on that determine the proper operation parameters. The parameters could include proper coordination tactics and dynamic activity levels to evade detection, signatures of valid traffic for creating attack traffic that appears valid in the current location and the time intervals of switching hosts to evade capture. Second, the agents could collaborate and analyse the mission objectives and based on that determine for instance what is the required attack strength. In case the current agent network does not have the required attack strength the

agents could again collaborate and compromise more hosts to achieve the required attack strength. In case of the opposite, only the required and changing portion of the agent network would be attacking simultaneously.

6 OVERVIEW OF CURRENT COUNTERMEASURES AGAINST DDOS ATTACKS

In this chapter a brief overview of defence mechanisms against DDoS attacks that target bandwidth is presented. Since DDoS attacks are most commonly about consuming bandwidth and as the bandwidth consumption attacks are the most difficult to defend against, defence to DDoS attacks that target the other two DoS attack types will not be discussed here. However, it should be noted that the exact properties of attack traffic generated in bandwidth consumption attacks might fill the criteria of the two other DoS attack types.

As it was already mentioned, there are no absolute defence solutions to bandwidth consumption attacks; however, several defence methods might be effective when they are properly implemented. Still, the technical defence methods are only a part of well-constructed risk management, which should also include business decisions. Householder et al. (2001) articulated these issues well and their paper is recommended as a starting point for designing defence against DoS attacks and integrating these issues into an organization's risk management plan. In this study these issues will not be further discussed.

It should also be pointed out that platforms have no significant differences in countering DDoS attacks. The discussion of adequate security has every now and then led to intense arguments between supporters of different platforms. An operating system with many remotely exploitable flaws is an ideal target for attacks that target software. Similarly, attacks that target protocols can be countered more efficiently with specific techniques only present in some operating systems. However, to bandwidth consumption attacks these issues

bear no resemblance, since bandwidth is an independent resource separate from the software that uses it.

When looking the countermeasures technically the countermeasures can be categorized by the time of invocation relating to the beginning and the end of an attack. In this study it was chosen to call these three stages *preventive*, *reactive* and *post-active* chronologically. These stages are briefly summarised in the following paragraphs. First, however, a few words about the DDoS attack traffic.

6.1 Issues of attack traffic

Most commonly DDoS attacks fall into the category of *attacks that target bandwidth,* which in essence refer to attacks that attempt to consume as much of the target's networking resources as possible by transmitting meaningless packets to the target (see paragraph 4.1). This attack mechanism is a logical choice for DDoS attacks, as a large number of hosts is likely to be sufficient in creating high enough traffic volume to saturate the target and thus causing DoS. In addition, there is not much reason to perform software targeted attacks with large number of hosts, as usually only a few packets is required to cause the desired outcome. Further, even though certain attacks targeting protocols may require several hosts to be performed, bandwidth consumption attacks are often easier to perform, more damaging and harder to prevent, which is why the attack type is usually preferred (see chapter four).

Considering the DDoS attack traffic, it is relevant to notice that it is not exactly similar to the attack traffic of singular bandwidth consumption attacks. Within

individual packets there rarely are any discrepancies, however, in traffic flows there is likely to be divergence.

To illustrate, consider a massive DoS attack launched from a single base. Most likely, the attack traffic flow travels constantly through almost the same route (only minor variations in routing are likely to occur in normal conditions) and eventually arrives at the same interface at the target. Due to the constant route and the heavy traffic flow the attack traffic might be identifiable assuming inter-domain cooperation between service providers responsible of routing the attack traffic is possible and the attack traffic has some distinguishable quality. In that case appropriate traffic filters could be installed without causing much damage to other traffic flows. However, uniform and conclusive attack traffic identification can never be guaranteed, as attack traffic may not have any distinguishable quality compared to other traffic flows (see paragraph 3.1). It is possible that reasonably sophisticated attackers could create such traffic.

Nevertheless, the links routing large amounts of traffic towards the target could be instructed to begin dropping the traffic heading to the target, even when resulting in loss of valid traffic. With this measure the other clients not depending of those links could maintain a better state of operation. The problem is that service providers might not engage into such an act, as it would cause damage to their legitimate clients and thus injure their own businesses.

Regardless, heavy traffic filtering is a valid defence method in case of a single-based DoS attack. In case of a highly distributed attack the above-mentioned measures are not generally feasible. The degree of distribution has an effect to the number of routes the attack traffic traverses as a whole, which is why there might not be any single observable traffic route carrying a massive amount of

packets towards the target. Consequently, traffic filtering would have to be performed to a greater number of links more blindly, which would result in increased loss of normal traffic.

6.2 Preventive countermeasures

Preventive mechanisms refer to the actions performed *prior to an attack* either to eliminate the possibility of being a target of attacks or to aid the target to endure the effects of attacks sufficiently (The notion of "sufficient" refers to a subjective decision of what has been seen as "acceptable" in an arbitrary situation). The role of a preventive viewpoint is emphasized here, as attacks can be prevented by removing the components that are required for it.

For instance, system administrators could take the appropriate steps to secure and keep auditing their networked machines to decrease the likelihood of getting those machines compromised and afterwards used as participants in malicious activity. Measures like these could be viewed as shared responsibilities amongst the Internet users, even though the threat of DDoS would not be considered relevant by all and as securing hosts does nothing to protect those hosts of being a target of DDoS.

Several preventive countermeasures were presented by Householder et al. (2001, 4-12) and by Mirkovic et al. (2002, 6-10), from which the most important can be summarized as *planning a proper risk management strategy*, *making arrangements with internet service providers*, *balancing load of the key servers*, *acquiring abundance of bandwidth*, *filtering of all unnecessary traffic*, *enabling appropriate server protection mechanisms* and *hiding the inners of a network from outsiders*.

Planning a proper risk management strategy is a matter of preparing for attacks, determining what should be protected, how and at what cost. It is a plan of procedures that guides the responses to various attacks and the recovery of possible damages. It should estimate the effects different types of attack scenarios might have from business level issues to technical level details.

Load balancing is a term referring to key services being distributed to multiple locations. Thus, in case an attack is primarily engaged against a certain server or servers, the other servers may still be able to operate sufficiently.

Acquiring abundance of bandwidth is probably the most expensive, but perhaps the only feasible solution even in extreme conditions. The aim is to acquire as much of bandwidth and other resources to retain operability even in case of a powerful attack.

Filtering of all unnecessary traffic is a method addressing the problem in the most primal point of view. Filtering of all unnecessary traffic is a precaution for protecting own host or hosts from being compromised and perhaps consequently used in DDoS. In preventive sense the method does nothing to protect the hosts in question being attacked. Most commonly this method is about separating the intranet from the Internet by allowing only certain type of traffic possibly from specific locations from the Internet to enter the intranet.

Enabling appropriate server protection mechanisms refers to methods that improve the server's capabilities to endure attacks that otherwise would injure the server. One such a method is called “syncookies” (Bernstein 1996) available in modern Linux kernels, which is used to defend against a previously mentioned

DoS attack named “SYN-flooding”. It should be noted that against bandwidth consumption attacks mechanisms of this type cannot be used.

Hiding the inners of a network from outsiders is a method referring to the traditional “security through obscurity” paradigm. Regardless of the negative feedback towards this ideology, it still is a way to improve security. For instance, supposing the most important hosts of the network can be identified it is likely attacks will be targeted to these hosts, as thus the attacks would probably cause the greatest damages. On the other hand, if the network's structure and the identities of the most important hosts are concealed, the attacks are less likely to hit the most sensitive parts of the network, which may be enough for the target to endure the attacks. However, this method is only an additional level of security, which may be worthwhile when placed above sound security architecture.

6.3 Reactive countermeasures

Reactive mechanisms refer to the actions performed to mitigate the effects of one or more ongoing attacks and they consist of *detection* and *response* procedures. The most important methods of this class are briefly discussed in this paragraph.

Detection is the process of determining is the target under an attack; an attack must first be *detected* in order to level an appropriate defensive *response*. Typically, only elevated number of received packets is not necessarily a sign of ongoing bandwidth consumption attack. Traffic volumes fluctuate between different day times; there are “rush-hours” in the Internet as well. Similarly, major events may result in a massive, sudden usage of some particular service.

These situations are often referred as *flash-crowds* (Jung et al. 2002, 1). One example of flash-crowds was witnessed in September 11 2001 (Eisenberg and Partridge 2003; Jung et al 2002, 1).

When the first DoS attack tools targeting bandwidth appeared in the Internet the attack traffic they generated was often noticeably different from normal traffic. The difference, for instance, might have been visible as highly unusual or even completely erroneous protocol header values that would have not been observed in any other situation. For that reason, the detection was easy. However, as it has been already mentioned, there is no reason why traffic aimed to consume bandwidth maliciously should look any different compared to normal traffic (see paragraph 3.1). Based on that, it is probable that in the future DoS attack tools will aim to generate as normal looking traffic as possible to avoid detection. Consequently, the damage caused to legitimate traffic by responsive procedures will increase when separation between normal and attack traffic cannot be accurately done.

As with the attacks that target software (see subparagraph 4.1), the detection can be based on searching anomalies in *individual packets* or in *packet streams*, which in the most simplistic case means searching of errors or known attack signatures from the headers of the packets. For instance, some DoS attack tools contain errors in proper checksum calculations (for instance, see the source code of the TFN 1999). As this should not be acceptable behaviour of a normal program, any packet containing erroneous checksums could be dropped without further investigations. Detection can also be based on statistical pattern recognition (see subparagraph 4.1.4), where statistics could be used to estimate what kind of traffic is to be expected at different day times. High enough divergence between observed and expected traffic patterns violating the

programmed thresholds could be treated as a positive sign of an ongoing attack. The statistics to which the thresholds are based can be calculated either statically or dynamically (see subparagraph 4.1.4).

In the static calculation statistics are continuously calculated and compared to the "clean sample" obtained from conceivably valid traffic previously. When a pre-programmed threshold of anomalous events is breached or an event differs too much from the model values an event of positive detection will follow. In the dynamic calculation the sample on which the threshold values are based is continuously updated based on the traffic patterns observed. The difference between static and dynamic calculation is that the dynamic calculation is more considerate to network dynamics. The dynamic calculation enables automatic adjustment to a changing environment. However, the method is also more prone to malicious training. A careful injection of traffic with desired characteristics to the network would slowly change the system threshold values to the desired direction. Thus, the system is eventually mistrained to accept traffic patterns it would have not accepted before.

Response is the process of reaction after the detection procedure has verified that there is an attack in progress. Responsive methods have received a fair amount of attention probably because they intersect the field of controlling traffic aggregates and link congestion closely (Congestion is common situation due to heavy network usage, even without any malicious activity present). The majority of responsive methods include *traffic filtering* in some form. Most of the remaining responsive methods relate somehow to tracing the approximate attack source, often referred as *IP-tracing* or *IP-traceback* (see subparagraph 4.3.3). Besides these, countermeasures might be dynamically defined based on the exact attack attributes observed.

The key questions regarding reactive methods are *how* and *at what cost* they accomplish their task. For example, there are many ways to decide what traffic to filter and how aggressive the filtering should be. The success of defence is not defined by how efficiently the attack traffic is filtered, but how exclusively the attack traffic is filtered. In a similar manner, the accuracy of IP-tracing procedures is not the only indicator of success. Overhead, implementation costs, compatibility with current infrastructure and privacy are important issues as well.

6.4 Post-active countermeasures

Post-active methods refer to the actions performed after an attack has occurred attempting to mitigate the threat of DDoS in the future. Most commonly post-active methods are about tracing the attacker as well as analysing the vulnerabilities the attack exploited and engaging into repairs accordingly. The vulnerabilities in this case could be anything from poorly designed network structures to software flaws that enabled the usage and subsequent success of the attack.

Tracing one or more attackers is a task that relies heavily on the information gathered during the attack. Most commonly post-active traceback is performed manually by analysing log information created during the attack in collaboration with the owners of upstream links. The presence of appropriate log information throughout the attack traffic route is crucial and thus it is required that the owners of upstream links store log information as well. Since the attack traffic route may be of arbitrary length, the process of obtaining the log files may be difficult. Moreover, due to the amount of log information, log files are often deleted at the end of a workday or within a few days, which is

why post-active traceback is often possible for only a short period. However, IP-tracing can be done reactively as well, which quite often is easier. Most of the proposed reactive tracing methods (see paragraph 4.3.3) are highly automated, which require human intervention only in special cases. Still, these methods face a similar problem; the methods require wide enough deployment to be usable.

It should also be pointed out that the result of IP-traceback is an approximation of the attack source in the best case. The result can only pinpoint the network to which the attacking host belongs. IP-traceback cannot pinpoint the exact location of the attacking source with certainty due to the Internet's property of complete anonymity (see subparagraph 3.1.3).

Although IP-traceback may be useful in case of singular DoS attack, in case of a distributed attack IP-traceback cannot be much of assistance. First of all, there might be thousands of hosts attacking, which means IP-traceback would have to be performed on each of the attacking hosts. Second, the indirect nature of DDoS makes the locating of the actual attacker, not the machines that ultimately generate the traffic very difficult.

6.5 Summary

In this chapter a brief overview of the countermeasures against DDoS attack mechanisms was presented. Most commonly DDoS attacks are bandwidth consumption attacks, however, the distributed and indirect nature of the attack increases the difficulty of defence. Nonetheless, there are countermeasures that can be mounted and these countermeasures can be categorized by the time of invocation concerning the beginning and the end of an attack. A categorization

of this type was presented in this study and it consists of three stages. These stages were labeled as preventive, reactive and post-active. The objectives and the methods to accomplish the objectives of are summarized into TABLE 5.

TABLE 5. The Stages of Countermeasures Against DDoS Attacks.

Stage	*Objectives*	*Methods*
Preventive	• Eliminate the possibility of DDoS attacks and • raise the level of readiness	• Planning risk management, • arrangements with service providers, • load balancing, • acquiring abundance of bandwidth, • hiding the inners of the own networks, • securing own machines and • applying server protection mechanisms
Reactive	• Detect ongoing attacks, • mitigate the effects of the ongoing attacks and • trace the attackers	• Intrusion detection, • traffic filtering and • IP traceback
Post-active	• Repair the damages, • analyse the effects of the endured attack, • mitigate the threat of future attacks and • trace the attackers	• Analysis of the exploited vulnerabilities, • defence mechanisms update according to the analysis and • IP traceback

7 SUMMARY

DoS attacks are a global problem and although they usually occur in the Internet, DoS attacks could occur in any other network as well. In terms of information security, DoS attacks target availability and by definition, DoS attacks cannot attack against integrity or confidentiality. The public emergence of DoS attacks occurred in the midst of 1990's, when the CERT released the first advisories regarding DoS attacks. Since then the DoS attacks have become common and one of the most damaging attack types there is. There are numerous methods of performing DoS attacks; however, all of those methods can be categorized to groups of attacks that target software, attacks that target protocols and attacks that target bandwidth. These attack types are referred as DoS attack mechanisms. Attacks that target software aim to exploit flaws in software, attacks that target protocols aim to exploit flaws in protocol specifications and attacks that target bandwidth aim to consume network resources. From these attack types the attacks that target bandwidth are the most severe, since it is the only attack type that has no absolute defence solution to be found without changes to the core of the Internet.

DDoS attacks are a subset of DoS attacks and they are characterized by multiple attacking hosts and coordination. Coordination defines the methods of passing information amongst the nodes in the DDoS network. In essence, coordination is a combination of technical communication mechanisms, such as passing encrypted data in emails and network model, which defines the DDoS network structure. In addition to the actual DoS attack mechanisms the execution of DDoS attacks depends of DDoS network mechanisms. DDoS network mechanisms consist of choosing the network model, creating the network,

coordinating the attack within the nodes of the network and additional functionality, which includes every supplemental mechanism that is not mandatory in a way the mechanisms in the other three classes are. Together with the DoS attack mechanisms DDoS network mechanisms form the classification labeled as DDoS attack mechanisms.

There is currently no absolute defensive solution to DDoS attacks that target bandwidth, which is the most common form of DDoS attacks. This notion is due to the core principles of the Internet, which make it impossible to distinguish malicious traffic from normal traffic, make complete user anonymity possible and allow an arbitrary host to send limitlessly traffic to any other host connected to the Internet. The distributed and indirect nature of the attack increases the difficulty of defence further. However, there are methods that can alleviate the problem. These countermeasures can be categorized by their time of invocation in relation to the beginning and the end of an attack. The categories are preventive, reactive and post-active. Preventive mechanisms consist of actions aimed to remove the threat of DDoS and mitigate the effects of DDoS. Reactive mechanisms consist of detection and responsive procedures, which are invoked accordingly; response cannot occur until an attack is detected. Reactive mechanisms most commonly are about traffic filtering and tracing the source of the attack. Post-active mechanisms consist of actions aimed to mitigate the threat of DDoS in the future. Post-active mechanisms often are about tracing the source of the attack and repairing the damages.

When considering the future of DDoS attack mechanisms there is still much to evolve, as the contemporary DDoS attack tools are unsophisticated in both DoS attack and DDoS network mechanisms. These tools lack the ability to consider the traffic dynamics and hence resort creating attack traffic that is

distinguishable. Similarly, these tools are based on network models, network creation and coordination mechanisms that in general are deprecated and inadequate. In essence, these tools are unable to consider situation dynamics. This study briefly presented the concept of an advanced automated intrusion agent, which presents a formidable threat of realizing very large-scale DDoS attacks that can be dynamically controlled while minimizing the possibility of seizure and consequent shut down of the DDoS network.

A major incentive for this study was the lacking of research regarding DDoS attack technology. Defence against DDoS attacks has been studied; however, the efficiency of the proposed defence methods is under question if the attributes that form the attacks are not scientifically established. Comprehensive classification that clearly and logically categorizes the DDoS attack mechanisms into distinct classes is a way to approach the problem. This type of classification answers to the question what to anticipate from an arbitrary DDoS attack, since it comprehensively depicts the components of DDoS attacks. In this study the field of DDoS attacks was discussed in detail, the core principles of DDoS attacks were formulated and a new classification of DDoS attacks meeting the aforementioned criteria was created. The classification was labelled as DDoS attack mechanisms and it is the main single contribution of this study. The classification was aimed to clarify the various aspects of DDoS attacks, and due to the previously stated reasons, it is a novel addition to the field of DoS attacks and computer security. This study also defined and detailed the most important concepts related to the subject, which previously have lacked proper definitions. Lastly, this study discussed some of the ways the various DDoS attack mechanisms could be evolved in the future.

As for the future research, since wireless networks are becoming increasingly common in almost everywhere imaginable, it is probable that DoS attacks will expand to those networks in the near future. Research focusing on DoS and DDoS attacks in the wireless environments would thus be valuable. In addition, in the upcoming arrival of advanced automated intrusion agents a study detailing the capabilities of such agents and the use of them in DDoS attacks would be beneficial in understanding how to develop countermeasures against the attacks of the next generation. However, basic research regarding the defences against DDoS is still required and probably the issue of the most importance.

This study showed that the current Internet infrastructure cannot handle the threat of DDoS well. The hypothesised completely DDoS-resistant network infrastructure is a subject of great significance, since when the next versions of the public Internet will be devised in the future, the history has shown that the threat of DDoS should be well considered. However, better defences against DDoS attacks in the contemporary Internet should still be researched. With comprehensive understanding of the core mechanisms of DDoS attacks actual attack mechanisms can be anticipated and the use of them can be better detected.

Finally, a framework for dismantling the mechanisms of actual DDoS attack methods based on the classification presented in this study could be devised. Such a framework would enable an arbitrary DDoS attack method to be disassembled into components defined in the framework. The disassembled attack method would be in turn suitable for further analyses and other operations, such as runtime adjustment of defence parameters according to the combination of mechanisms the attack method uses. On the other hand, a

framework of that type would also enable compilation of DDoS attack methods with varying mechanism combinations. It is possible that these compilations would allow elaborate tests and analyses to be performed, which consequently could result in indications of optimal attack mechanism combinations for an arbitrary situation.

REFERENCES

Albert R., Hawoong J. and Barabasi A-L., "The Internet's Achilles' Heel: Error and attack tolerance of complex networks," Nature 406, pp.378-382, Jul. 2000.

Bailey M., Cooke E., Jahanian F., Watson D. and Nazario J., "The Blaster Worm: Then and Now," IEEE Security and Privacy, Vol. 3, no. 4, pp. 26-31, Aug. 2005.

Barlow J. and Thrower W., "TFN2K – An analysis" <http://www.packetstormsecurity.org/distributed/TFN2k_Analysis-1.3.txt>, Mar. 2000. [referenced 10.10.2005]

BBC News, "Yahoo brought to standstill," <http://news.bbc.co.uk/1/hi/sci/tech/635048.stm>, Feb. 9, 2000. [referenced 19.2.2004]

Bellovin S., "Security Problems in the TCP/IP Protocol Suite," ACM Computer Communications Review, Vol. 19, no. 2, pp. 32-48, Apr. 1989.

Bellovin S., "Distributed Firewalls," ;login: magazine, pp. 39-47. Nov. 1999.

Bellovin S., "ICMP Traceback Messages," Internet draft, <http://www.research.att.com/~smb/papers/draft-bellovin-itrace-00.txt>, Sep. 2000. [referenced 5.9.2004]

Berghel H., "Malware Month," Communications of the ACM, Vol. 46, no. 12, pp. 15-19, Dec. 2003.

Bernstein D., "SYN cookies", <http://cr.yp.to/syncookies.html>, Sep. 1996. [referenced 6.3.2004]

Blitznet, <http://www.packetstormsecurity.org/distributed/blitznet.tgz>, 1999. [referenced 3.10.2005]

Blumenthal M. and Clark D., "Rethinking the Design of the Internet: The End-to-End Argument vs. the Brave New World," ACM Transactions on Internet Technology, Vol. 1, pp. 70-109, Aug. 2001.

CERT Coordination Center, "CERT Advisory CA-1996-01 UDP Port Denial-of-Service Attack," <http://www.cert.org/advisories/CA-1996-01.html>, Feb. 1996. [referenced 30.1.2004]

CERT Coordination Center, "CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks, " <http://www.cert.org/advisories/CA-1996-21.html>, Sep. 1996. [referenced 30.1.2004]

CERT Coordination Center, "CERT Advisory CA-1996-26 Denial-of-Service Attack via Ping, " <http://www.cert.org/advisories/CA-1996-26.html>, Dec. 1996. [referenced 30.1.2004]

CERT Coordination Center, "CERT Advisory CA-1997-28 IP Denial-of-Service Attacks, " <http://www.cert.org/advisories/CA-1997-28.html>, Dec. 1997. [referenced 30.1.2004]

CERT Coordination Center, “CERT Advisory CA-1998-01 Smurf IP Denial-of-Service Attacks, “ <http://www.cert.org/advisories/CA-1998-01.html>, Jan. 1998. [referenced 30.1.2004]

CERT Coordination Center, “CERT Incident Note IN-1999-04,“ <http://www.cert.org/incident_notes/IN-99-04.html>, Dec. 1999. [referenced 30.1.2004]

CERT Coordination Center, “CERT Incident Note IN-1999-05,“ <http://www.cert.org/incident_notes/IN-99-05.html>, Dec. 1999. [referenced 30.1.2004]

CERT Coordination Center, “CERT Incident Note IN-1999-07,“ <http://www.cert.org/incident_notes/IN-99-07.html>, Dec. 1999. [referenced 30.1.2004]

CERT Coordination Center, “CERT Advisory CA-1999-17 Denial-of-Service Tools, “ <http://www.cert.org/advisories/CA-1999-17.html>, Dec. 1999. [referenced 30.1.2004]

CERT Coordination Center, “CERT Advisory CA-2000-01 Denial-of-Service Developments,“ <http://www.cert.org/advisories/CA-2000-01.html>, Jan. 2000. [referenced 30.1.2004]

CERT Coordination Center, “Denial of Service Attacks,” <http://www.cert.org/tech_tips/denial_of_service.html>, Jun 2001. [referenced 16.1.2004]

CERT Coordination Center, “CERT Advisory CA-2002-27 Apache/mod_ssl Worm,” <http://www.cert.org/advisories/CA-2002-27.html>, Oct. 11, 2002. [referenced 20.3.2005]

Clark D., Sollins K., Wroclawski J. and Braden R., “Tussle in Cyberspace: Defining Tomorrow's Internet,” Proceedings of the 2002 conference on Applications, technologies, architechtures and protocols for computer communications, pp. 347-356, Aug. 2002.

Davies D., “Historical Note on the Early Development of Packet Switching,” <http://www.cs.utexas.edu/users/kata/HISTORY/DAVIES/Davies01.pdf>, 1982. [referenced 4.1.2004]

DDoS-ca.org <http://www.ddos-ca.org/>. [referenced 9.3.2004]

Dean D., Franklin M. and Stubblefield A., “An Algebraic Approach to IP Traceback,” ACM Transactions on Information and System Security (TISSEC), pp. 119-137, May 2002.

Dietrich S,. Long N. and Dittrich D., “An analysis of the ``Shaft'' distributed denial of service tool,” <http://www.packetstormsecurity.org/distributed/shaft_analysis.txt>, Mar. 2000. [referenced 3.9.2005]

Distributed DNS Flooder v0.1b (ddnsf), <http://www.packetstormsecurity.org/distributed/ddnsf.tar.gz>, 2001. [referenced 3.9.2005]

Dittrich D., "Distributed Denial of Service (DDoS) Attacks/Tools," <http://staff.washington.edu/dittrich/misc/ddos/>, 2005. [referenced 17.11.2004]

Dittrich D., "Analysis of the "Power bot," <http://staff.washington.edu/dittrich/misc/power.analysis.txt>, 2001. [referenced 3.9.2005]

DOSnet.c, <http://www.packetstormsecurity.org/distributed/DOSnet.c>, 2002. [referenced 3.9.2005]

Drdos v2.0, <http://www.packetstormsecurity.org/distributed/drdos_v2.0.tar.gz>, 2002. [referenced 3.10.2005]

Eisenberg J. and Partridge C., "The Internet Under Crisis Conditions: Learning from September 11," <http://intel.si.umich.edu/tprc/papers/2003/195/net911.pdf>, 2003. [referenced 6.3.2004]

Erickson C., "USENET as a Teaching Tool," Proceedings of the twenty-fourth SIGCSE technical symposium on Computer science education, pp. 43-47, Mar. 1993.

Federal Networking Council (FNC) Resolution, "Definition of the Internet," <http://www.itrd.gov/fnc/Internet_res.html>, Oct. 1995. [referenced 15.1.2004]

Ferguson P. and Senie D., "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," RFC 2827, <http://www.faqs.org/rfcs/rfc2827.html>, May 2000. [referenced 17.3.2004]

Festa P., "'Smurf' attack hits Minnesota," <http://digitalcity.com.com/2100-1001-209209.html?legacy=cnet&tag=rltdnws>, CNET, Mar. 17, 1999. [referenced 20.3.2005]

Flitz, <http://www.packetstormsecurity.org/distributed/flitz-0.1.tgz>, 2001. [referenced 3.9.2005]

Free On-Line Dictionary of Computing (FOLDOC), <http://foldoc.doc.ic.ac.uk/foldoc/index.html>, 1993. [referenced 16.1.2004]

Garber L., "Denial-of-Service Attack Rip the Internet," IEEE Computer, vol. 33, no. 4, pp. 12-17, Apr. 2000.

Glave J., "Smurfing cripples ISPs," <http://www.wired.com/news/technology/0,1282,9506,00.html>, Wired News, Jan. 7, 1999. [referenced 20.3.2005]

Goodrich M., "Efficient Packet Marking for Large-Scale IP Traceback," Proceedings of the 9th ACM conference on Computer and communications security, pp. 117-126, Nov. 2002.

Gordon L., Loeb M., Lucyshyn W. and Richardson R., "CSI/FBI Computer Crime and Security Survey 2004," <http://www.gocsi.com>, 2004.

Gresty D., Shi Q. and Merabti M., "Requirements for a General Framework for Response to Distributed Denial-of-Service," Requirements of the 17th Annual Conference on Computer Security Applications, pp. 422, Dec. 2001.

GT Bot (Global Threat), <http://swatit.org/bots/gtbot.html>, 2003. [referenced 3.9.2005]

Houle K. J. and Weaver G. M., "Trends in Denial of Service Attack Technology," CERT Coordination Center, Oct. 2001.

Householder A., Manion A., Pesante L., Weaver G. and Thomas R., "Managing the Threat of Denial-of-Service Attacks," CERT Coordination Center, Oct. 2001.

Howard J., "An Analysis of security incidents on the Internet 1989 – 1995," Carnegie Mellon University, Carnegie Institute of Technology, <http://www.cert.org/research/JHThesis/Start.html>, Apr. 1997. [referenced 16.1.2004]

Ioannidis S., Keromytis A., Bellovin S. and Smith J., "Implementing a Distributed Firewall," Proceedings of the 7th ACM conference of Computer and communications security, pp. 190-199, Nov. 2000.

Johnson N. and Jajodia S., "Exploring Steganography: Seeing the Unseen," IEEE Computer, Vol. 31, no. 2, pp. 26-34, Feb. 1998. [referenced 10.10.2005]

Jonsson E., "An Integrated Framework for Security and Dependability," Proceedings of the 1998 workshop on New security paradigms, pp. 22-29, Jan. 1998.

Jung J., Krishnamurthy B. and Rabinovich M., "Flash Crowds and Denial of Service Attacks: Characterizations and Implications for CDNs and Web Sites," Proceedings of the eleventh international conference on World Wide Web, pp. 293-304, May 2002.

Kabay M., "Distributed Denial-of-Service Attacks, Contributory Negligence and Downstream Liability," ACM Ubiquity Vol. 2, <http://www.acm.org/ubiquity/views/m_kabay_1.html>, Feb. 2000. [referenced 5.2.2004]

Kaiten, <http://www.packetstormsecurity.org/irc/kaiten.c>, 2001. [referenced 3.9.2005]

Knigth, <http://www.packetstormsecurity.org/distributed/knight.c>, 2001. [referenced 3.9.2005]

Kaleton Internet, "Combination of Misuse and Anomaly Network Intrusion Detection Systems," Version 1.0, <http://www.kaleton.com/research/kaletonidspaper.pdf>, Mar. 2002. [referenced 13.11.2004]

Kargl F., Maier J. and Weber M., "Protecting Web Servers from Distributed Denial of Service Attacks," Proceedings of the tenth international conference on World Wide Web, pp. 514-524, Apr. 2001.

Kuzmanovic A. and Knightly E., "Low-Rate TCP-Targeted Denial of Service Attacks," Proceeedings of the 2003 conference on Applications, technologies, architechtures, and protocols for computer communications, pp. 75-86, Aug. 2003.

Lejeune M., "Awareness of distributed denial of service attacks' dangers: role of Internet pricing mechanisms," Netnomics, vol. 4, no. 2, pp. 145-162, Nov. 2002.

Lewandowski S., "Frameworks for Component-Based Client/Server Computing," ACM Computing Surveys, Vol. 30, no. 1, Mar. 1998.

Mahajan R., Bellovin S., Floyd S., Ioannidis J., Paxson V. and Shenker S., "Controlling High Bandwidth aggregates in the Network," ACM SIGCOMM Computer Communication Review, Vol. 32, no. 3, pp. 62-73, Jul. 2002.

Marchesseau M., ""Trinity" - distributed-denial-of-service attack tool," <http://www.giac.org/certified_professionals/practicals/gsec/0123.php>, 2000. [referenced 3.9.2005]

Moore D., Voelker G. and Savage S., "Inferring Internet Denial-of-Service Activity," Proceedings of the 2001 USENIX Security Symposium, 2001.

Moore D., Paxson V., Savage S., Shannon C., Staniford S. and Weaver N., "Inside the Slammer worm," IEEE Security & Privacy, Vol. 1, no. 4, pp. 33-39, Aug. 2003.

Moraes M., "What is Usenet," <http://www.faqs.org/faqs/usenet/what-is/>, Dec. 1999. [referenced 10.10.2005]

Mirkovic J., Martin J. and Reiher P., "A Taxonomy of DDoS Attacks and DDoS defence Mechanisms," UCLA Computer Science Department, Technical report no. 020018. <http://www.lasr.cs.ucla.edu/ddos/ucla_tech_report_020018.pdf>, 2002. [referenced 12.1.2004]

Mixter, "TFN – Tribal Flood Network", <http://www.packetstormsecurity.org/distributed/tfn.tgz>, 1999. [referenced 8.12.2004]

Mstream, <http://www.packetstormsecurity.org/distributed/mstream.txt>, 2000. [referenced 3.9.2005]

Naugle M., "Illustrated TCP/IP: a graphic guide to the protocol suite," John Wiley & Sons, Inc., 1998.

Nazario J., Anderson J., Wash R. and Connelly C., "The Future of Internet Worms," <http://www.crimelabs.net/docs/worms/worm.pdf>, Jul. 2001. [referenced 12.1.2004]

Oikarinen J., "RFC 1459 – Internet Relay Chat Protocol," <http://www.faqs.org/rfcs/rfc1459.html>, May 1993. [referenced 27.3.2004]

Omega v3 Beta, <http://www.packetstormsecurity.org/distributed/omegav3.tgz>, 2000. [referenced 3.9.2005]

Oppliger R., "Internet Security: Firewalls and Beyond," Communications of the ACM, Vol. 40, no. 5, pp. 92-102, May 1997.

Orman H., "The Morris Worm: A Fifteen-Year Perspective," Security & Privacy Magazine, IEEE, Vol. 1, no. 5, pp. 35-43, Sep.-Oct. 2003.

Paxson V., "An Analysis of using reflectors for distributed-denial-of-service attacks," ACM SIGCOMM Computer Communication Review, Vol. 31, no. 3, pp. 38-47, Jul. 2001.

Phrack Magazine, "Project Neptune," <http://www.phrack.org/phrack/48/P48-13> no. 48, file 13, Sep. 1996. [referenced 18.3.2005]

Poulsen K., "FBI Busts alleged DDoS Mafia," <http://www.securityfocus.com/news/9411>, SecurityFocus Aug. 26, 2004. [referenced 16.10.2004]

Peer-to-peer UDP Distributed Denial of Service (PUD), <http://www.packetstormsecurity.org/distributed/pud.tgz>, 2002. [referenced 3.9.2005]

RFC 791, "RFC 791 – Internet protocol", <http://www.ietf.org/rfc/rfc0791.txt>, Sep. 1981. [referenced 3.10.2005]

RFC 792, "RFC 792 – Internet control message protocol", <http://www.ietf.org/rfc/rfc0792.txt>, Sep. 1981. [referenced 10.10.2005]

RFC 793, "RFC 793 – Transmission Control Protocol," <http://www.faqs.org/rfcs/rfc793.html>, Sep. 1981. [referenced 28.4.2004]

RFC 1034, "RFC 1034 – Domain Names – Concepts and Facilities," <http://www.faqs.org/rfcs/rfc1034.html>, Nov. 1987. [referenced 3.10.2005]

RFC 1035, "RFC 1035 – Domain Names – Implementation and Specification," <http://www.faqs.org/rfcs/rfc1035.html>, Nov. 1987. [referenced 3.10.2005]

Richardson R., "CSI/FBI Computer Crime and Security Survey 2003," <http://www.gocsi.com>, 2003.

Roberson J., "Jolt 1.0", Source code for the DoS attack tool named Jolt, <http://packetstormsecurity.nl/Exploit_Code_Archive/jolt.c>, 1997. [referenced 21.6.2004]

Roberts L., "The Arpanet and computer networks," Proceedings of the ACM Conference on the History of personal workstations, pp. 51-58, 1986.

Roesch M., "SNORT – Lightweight Intrusion Detection for Networks," Proceedings of the 13th Conference on Systems Administration, pp. 229-238, Nov. 1999.

Savage S., Wetherall D., Karlin A. and Anderson T., "Practical Network Support for IP Traceback," ACM SIGCOMM Computer Communication Review, Proceedings of the conference on Applications, Technologies, Architechtures, and Protocols for Computer Communications, pp. 295-306, Aug 2000.

Schollmeier R., "A Definition of *Peer-to-Peer* Networking for the Classification of *Peer-to-Peer* Architechtures and Applications," Proceedings of the First International Conference on Peer-to-Peer Computing (P2P'01), pp. 101-102, IEEE Computer Society, Aug. 2001.

Seattle Post-Intelligencer, "Hacker discrupts Web sites," <http://seattlepi.nwsource.com/business/yaho09.shtml>, Feb. 9, 2000. [referenced 21.3.2005]

Sekar R., Frullo G., Shanbhag T., Tiwari A., Yang H. and Zhou S., "Specification-based Intrusion Detection: A New Approach for Detecting Network Intrusions," Proceedings of the 9th ACM conference on Computer and communications security, pp. 265-274, Nov. 2002

Shamir A. and Someren N., "Playing hide and seek with stored keys," Lecture Notes In Computer Science; Vol. 1648. Proceedings of the Third International Conference on Financial Cryptography, pp. 118-124, 1999.

Simple Nomad, "Strategies for Defeating Distributed Attacks," <http://downloads.securityfocus.com/library/sn_ddos.doc>, 2001. [referenced 11.10.2005]

Skydance v3.6, <http://www.packetstormsecurity.org/distributed/skd36.zip>, 2001. [referenced 3.9.2005]

Slapper, alias apache-worm, bugtraqworm, Modap, <http://packetstormsecurity.org/0209-exploits/bugtraqworm.tgz>, 2002. [referenced 20.3.2005]

Snoeren A., Sanchez L., Jones C., Tchakountio, Schwartz, Kent S. and Strayer T., "Single Packet IP Traceback," IEEE/ACM Transactions on Networking (TON), Vol. 10, no. 6, pp. 721-734, Dec 2002.

Sommer R. and Paxson V., "Enhancing Byte-Level Network Intrusion Detection Signatures with Context," Proceedings of the 10th ACM conference on Computer and communication security, pp. 262-271, Oct. 2003.

Song D. and Perrig A., "Advanced and Authenticated Marking Schemes for IP Traceback," Computer Science Division (EECS) University of Berkeley, California, Technical Report No. UCB/CSD-00-1107, Jun 2000.

Spech S. and Lee R., "Taxonomies of Distributed Denial of Service Attacks, Tools and Countermeasures," Princeton University Department of Electrical Engineering, Technical report CE-L2003-004, May 2003.

StacheldrahtV4, <http://www.packetstormsecurity.org/distributed/stachel.tgz>, 2000. [referenced 3.9.2005]

Staniford S., Paxson V. and Weaver N., "How to 0wn the Internet in Your Spare Time," Proceedings of the 11th USENIX Security Symposium, 2002.

Stolfo S., Lee W., Chan P., Fan W. and Eskin E, "Data mining-based intrusion detectors: an overview of the columbia IDS project," ACM SIGMOD Record, pp. 5-14, Dec. 2001.

Sundaram A., "An Introduction to Intrusion Detection," <http://www.acm.org/crossroads/xrds2-4/intrus.html>, 1996. [referenced 8.12.2004]

The Jargon File, version 4.4.7, <http://www.catb.org/~esr/jargon/>, 2003. [referenced 18.2.2004].

The Register, "Cloud Nine blown away, blames hack attack," <http://www.theregister.co.uk/2002/01/22/cloud_nine_blown_away_blames>, Tue. 22nd Jan. 2002. [referenced 10.10.2004]

Tribe Flood Network (TFN), <http://packetstormsecurity.org/groups/mixter/tfn.tgz>, 1999. [referenced 20.3.2005]

Tribe FloodNet – 2k edition (TFN2k), <http://packetstormsecurity.org/distributed/tfn2k.tgz>, 1999. [referenced 20.3.2005]

Trin00, <http://www.packetstormsecurity.org/distributed/trinoo.tgz>, 1999. [referenced 3.9.2005]

Tynan D., "Sobig May Be Working for the Spammers," <http://www.pcworld.com/news/article/0,aid,112261,00.asp>, PCWorld, Aug. 29, 2003. [referenced 20.3.2005]

Vigna G., Valeur F and Kemmerer R, "Designing and implementing a family of intrusion detection systems," Proceedings of the 9th European software engineering conference held jointly with 10th ACM SIGSOFT international symposium on Foundations of software engineering, p. 88-97, Sep. 2003.

Vivo M., Vivo G., Koeneke R. and Isern G, "Internet Vulnerabilities Related to TCP/IP and T/TCP," ACM SIGCOMM Computer Communication Review, Vol. 29, no. 1, pp. 81-85, Jan. 1999.

Voyiatzis A. and Serpanos D., "Pulse: A Class of Super-Worms against Network Infrastructure," Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on, 2003, pp. 28-33.

Washingtonpost.com, “Attack on Internet Called Largest Ever,” <http://www.washingtonpost.com/ac2/wp-dyn?pagename=article&contentId=A828-2002Oct22&notFound=true>, Oct. 22, 2002. [referenced 20.3.2005]

Whalen S., “An introduction to Arp Spoofing”, <http://packetstormsecurity.org/papers/protocols/intro_to_arp_spoofing.pdf>, Apr. 2001. [referenced 10.10.2005]

Wired News, “Yahoo on Trail of Site Hackers,” <http://www.wired.com/news/business/0,1367,34221,00.html>, Feb. 8, 2000. [referenced 19.2.2004]

Wordnet 2.0, <http://www.cogsci.princeton.edu/cgi-bin/webwn>. [referenced 16.1.2004]

APPENDIX 1. CONCEPTS CREATED OR MODIFIED IN THIS STUDY

Denial-of-Service

Denial-of-Service (DoS) is an event or a situation, in which a legitimate client cannot access the requested service to which the client is entitled to and which should be available.

Distributed Denial-of-Service

Distributed Denial-of-Service (DDoS) attack is a DoS attack, in which a multitude of hosts performs DoS attacks in a coordinated manner to one or more targets.

Distributed Denial-of-Service network

DDoS network is a network of hosts that are being controlled by a same static entity using the same control interface to administrate DDoS attacks.