TKK Dissertations 32
Espoo 2006

MITIGATING DENIAL OF SERVICE ATTACKS IN COMPUTER NETWORKS

Doctoral Dissertation

Jarmo Mölsä

Helsinki University of Technology
Department of Electrical and Communications Engineering
Communications Laboratory

TKK Dissertations 32
Espoo 2006

MITIGATING DENIAL OF SERVICE ATTACKS IN COMPUTER NETWORKS

Doctoral Dissertation

Jarmo Mölsä

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Department of Electrical and Communications Engineering for public examination and debate in Auditorium S5 at Helsinki University of Technology (Espoo, Finland) on the 5th of June, 2006, at 12 noon.

Helsinki University of Technology
Department of Electrical and Communications Engineering
Communications Laboratory

Teknillinen korkeakoulu
Sähkö- ja tietoliikennetekniikan osasto
Tietoliikennelaboratorio

Distribution:
Helsinki University of Technology
Department of Electrical and Communications Engineering
Communications Laboratory
P.O. Box 3000
FI - 02015 TKK
FINLAND
URL: http://www.sahko.tkk.fi/
Tel. +358-9-4511
E-mail: jarmo.molsa@kolumbus.fi

ISBN 951-22-8214-3
ISBN 951-22-8215-1 (PDF)
ISSN 1795-2239
ISSN 1795-4584 (PDF)
URL: http://lib.tkk.fi/Diss/2006/isbn9512282151/

TKK-DISS-2143

Picaset Oy
Helsinki 2006

HELSINKI UNIVERSITY OF TECHNOLOGY P. O. BOX 1000, FI-02015 TKK http://www.tkk.fi	ABSTRACT OF DOCTORAL DISSERTATION

Author Jarmo Mölsä

Name of the dissertation

Mitigating denial of service attacks in computer networks

Date of manuscript May 17, 2006	Date of the dissertation June 5, 2006

☐ Monograph ☒ Article dissertation (summary + original articles)

Department	Department of Electrical and Communications Engineering
Laboratory	Communications Laboratory
Field of research	Network security
Opponents	Professor William Hutchinson (Edith Cowan University, Australia), and
	Professor Aki-Mauri Huhtinen (National Defence College, Finland)
Supervisor	Docent Jorma Jormakka (Helsinki University of Technology)

Abstract

This dissertation studies how to defend against denial of service (DoS) attacks in computer networks. As it is not possible to prevent these attacks, one must concentrate on mitigating them. A comprehensive approach for mitigating DoS attacks is presented here. This approach is based on understanding both attack and defense mechanisms, and selecting a cost-effective set of defenses using risk management. Defense mechanisms are, however, not available against all possible attack types. For example, organization-specific servers of the Domain Name System are typically not well-managed, and DoS attacks against these kinds of name servers can easily be successful. This dissertation describes and simulates a new defense mechanism for protecting these authoritative name servers.

A new approach in implementing defenses is cross-layer security, in which information from different protocol layers is used in a coordinated fashion instead of separating layers strictly. Two cross-layer designs are presented here for mitigating range attacks in ad hoc networks. The range attack is a new DoS attack where an attacker modifies the transmission range of a wireless node periodically. This overloads an ad-hoc routing protocol. The simulation results indicate the usefulness of cross-layering in mitigating the range attack. Resilience of different ad-hoc routing protocols against the range attack is also analyzed here. According to the simulation results this resilience is situation dependent. Depending on the quality of service requirements of the primary application, different ad-hoc routing protocols are resistant against the range attack.

Game theory is used here to study the selection of defense strategies. The analysis shows that it can be beneficial to use different defense strategies randomly, one at a time. The game theoretic approach points out a possibility for using meta-strategies where an attacker can force a victim to perceive the benefits and weaknesses of a defense mechanism in an unrealistic way.

Evaluations of existing defense mechanisms are typically carried out under ideal conditions or relevant risks are completely ignored. This dissertation presents a taxonomy of criteria for evaluating defense mechanisms against DoS attacks. This taxonomy gives a list of issues to be considered during an evaluation process. The effectiveness of rate limiting is evaluated here with special attention paid to the damage on legitimate traffic.

Keywords Network security, denial of service attacks, attack mechanisms, defense mechanisms.

ISBN (printed) 951-22-8214-3	ISSN (printed) 1795-2239
ISBN (pdf) 951-22-8215-1	ISSN (pdf) 1795-4584
ISBN (others)	Number of pages 125 p. + app. 99 p.

Publisher Communications Laboratory, Helsinki University of Technology

Print distribution

☒ The dissertation can be read at http://lib.tkk.fi/Diss/2006/isbn9512282151/

TEKNILLINEN KORKEAKOULU PL 1000, 02015 TKK http://www.tkk.fi	VÄITÖSKIRJAN TIIVISTELMÄ

Tekijä Jarmo Mölsä

Väitöskirjan nimi

Suojautuminen palvelunestohyökkäyksiltä tietokoneverkoissa

Käsikirjoituksen jättämispäivämäärä 17.5.2006	Väitöstilaisuuden ajankohta 5.6.2006

☐ Monografia ☒ Yhdistelmäväitöskirja (yhteenveto + erillisartikkelit)

Osasto	Sähkö- ja tietoliikennetekniikan osasto
Laboratorio	Tietoliikennelaboratorio
Tutkimusala	Tietoverkkoturvallisuus
Vastaväittäjät	Professori William Hutchinson (Edith Cowan University, Australia) ja
	Professori Aki-Mauri Huhtinen (Maanpuolustuskorkeakoulu, Suomi)
Työn valvoja	Dosentti Jorma Jormakka (Teknillinen korkeakoulu)

Tiivistelmä

Tämä väitöskirja tutkii suojautumista tietokoneverkoissa tapahtuvilta palvelunestohyökkäyksiltä. Palvelunestohyökkäysten tavoitteena on joko estää palvelun käytettävyys siihen oikeutetuilta käyttäjiltä tai viivästää aikakriittisiä toimintoja. Palvelunestohyökkäykset tietokoneverkoissa perustuvat yleensä kohteen ylikuormittamiseen valtavalla määrällä tarpeetonta verkkoliikennettä. Toinen vaihtoehto palvelunestoon on väärinkäyttää ohjelmistoissa, protokollissa tai järjestelmissä olevia virheitä.

Tässä työssä kuvataan ja analysoidaan uusia suojausmenetelmiä palvelunestohyökkäysten vaikutusten lieventämiseksi, tutkitaan vaihtoehtoisten suojausmenetelmien valintaan liittyviä tekijöitä ja esitetään uusi joukko kriteereitä suojausmenetelmien käyttökelpoisuuden arvioimiseksi.

Palvelunestohyökkäyksiltä suojautumiselle esitetään tässä työssä kokonaisvaltainen lähestymistapa, joka korostaa sekä erilaisten hyökkäys- ja suojautumismenetelmien tuntemusta että kustannustehokkaan suojausmenetelmäjoukon valitsemista. Monia palvelunestohyökkäyksiä vastaan ei kuitenkaan ole saatavilla suojausmenetelmiä. Tässä työssä on kuvattu uusia menetelmiä suojata sekä Internetin nimipalvelua tulvintaan perustuvalta palvelunestohyökkäykseltä että langattomia ad hoc verkkoja uudelta signaalin kantaman vaihteluun perustuvalta hyökkäykseltä.

Väitöskirja tuo esille eri suojausmenetelmien mahdollisen tilanneriippuvaisuuden. Yksittäisen suojausmenetelmän tehokkuus ja käyttökelpoisuus voi riippua verkossa käytettävien sovellusten vaatimuksista. Myös peliteoriaa sovelletaan vaihtoehtoisten suojautumisstrategioiden valintaan.

Työssä esitetään joukko kriteereitä, jotka tulisi huomioida eri suojausmenetelmien käyttökelpoisuutta arvioitaessa. Arvioinnissa on huomioitava mm. väärien hälytysten aiheuttamia haittoja laillisille käyttäjille, suojausmenetelmistä aiheutuvia suorituskykyhaittoja ja mahdollisia väärinkäyttömahdollisuuksia. Lopuksi tässä työssä arvioidaan kaistanrajoituksen käyttökelpoisuutta tulvintaan perustuvien palvelunestohyökkäysten torjunnassa.

Asiasanat Tietoverkkoturvallisuus, palvelunestohyökkäykset, hyökkäysmekanismit, suojausmekanismit.	
ISBN (painettu) 951-22-8214-3	ISSN (painettu) 1795-2239
ISBN (pdf) 951-22-8215-1	ISSN (pdf) 1795-4584
ISBN (muut)	Sivumäärä 125 s. + liit. 99 s.

Julkaisija Tietoliikennelaboratorio, Teknillinen korkeakoulu

Painetun väitöskirjan jakelu

☒ Luettavissa verkossa osoitteessa http://lib.tkk.fi/Diss/2006/isbn9512282151/

ACKNOWLEDGEMENTS

I wish to express my gratitude to supervisor Professor Jorma Jormakka for his guidance and advice.

I am deeply grateful to the pre-examiners Dr. Tuomas Aura from Microsoft Research in UK and Professor William Hutchinson from the Edith Cowan University in Australia for their valuable effort in reading the manuscript and for giving excellent comments on the work.

I wish to express my gratitude to the Communications and the Networking Laboratories of the Helsinki University of Technology for providing me the opportunity to carry out this Ph.D. work.

The funding of this work has been received from different sources, in addition to the Communications and Networking Laboratories. I had a full-time position for two years at the Graduate School on Networks for Information Society (GSNIS). Part of the papers have been written for the ad hoc project funded by the Finnish Defence Forces.

A Ph.D. work would not be possible without the involvement of many different persons. I wish to express my sincere gratitude to Dr. Jouni Karvo, M.Sc. Mikko Kiviharju, Dr. Rauno Kuusisto, Lic.Sc. Boris Makarevitch, M.Sc. Risto Mutanen, Dr. Ilkka Norros, M.Sc. Cyril Onwubiko, and Dr. Pertti Raatikainen.

Naturally there are lots of other people who have helped me one way or the other, for example, all anonymous reviewers, staff at the Communications Laboratory, staff at the Networking Laboratory, staff at the National Defence College, and many colleagues at international conferences. I want to thank you for making it easier to achieve these results.

The support from my family, parents, brothers, and other friends has been irrecoverable, and has reminded me that there is life beyond the Ph.D. work.

Availability is not only related to computerized services, but also to human relationships. Despite of the limited free time, the time with my young daughters Tanja and Katariina was a great way of escaping the dead-line controlled world. Lastly, thanks to my beloved wife Kirsti. You organized all things at home while I was either physically or mentally away.

Helsinki, May 2006.

Jarmo Mölsä

CONTENTS

LIST OF PUBLICATIONS

P1 Jarmo Mölsä, "Mitigating denial of service attacks: A tutorial," *Journal of Computer Security*, vol. 13, no. 6, pp. 807–837, 2005.

P2 Jarmo Mölsä, "Mitigating DoS attacks against the DNS with dynamic TTL values," in *Proceedings of the Ninth Nordic Workshop on Secure IT Systems*, Espoo, Finland, Nov. 2004, pp. 118–124.

P3 Jarmo Mölsä, "Cross-layer designs for mitigating range attacks in ad hoc networks," in *Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Networks*, Innsbruck, Austria, Feb. 2006, pp. 64–69.

P4 Jarmo Mölsä, "Increasing the DoS attack resiliency in military ad hoc networks," in *Proceedings of the IEEE MILCOM*, Atlantic City, New Jersey, USA, Oct. 2005.

P5 Jarmo Mölsä, "A taxonomy of criteria for evaluating defence mechanisms against flooding DoS attacks," in *Proceedings of the First European Conference on Computer Network Defence*, Pontypridd, Wales, UK, Dec. 2005, pp. 13–22.

P6 Jarmo Mölsä, "Effectiveness of rate-limiting in mitigating flooding DoS attacks," in *Proceedings of the Third IASTED International Conference on Communications, Internet, and Information Technology*, St. Thomas, US Virgin Islands, USA, Nov. 2004, pp. 155–160.

P7 Jorma Jormakka and Jarmo Mölsä, "Modelling information warfare as a game," *Journal of Information Warfare*, vol. 4, no. 2, pp. 12–25, Sept. 2005.

AUTHOR'S CONTRIBUTION

Publications P1–P6:

The author of this dissertation has been completely responsible for all of these six publications, from the initial idea to the final submission (including the presentation of conference papers).

The publication P7:

This publication is joint work with prof. Jorma Jormakka, whose contributions were the following:

- The initial idea about the whole paper.
- A draft version for the sections about the terrorist game and the vandal game.
- The idea and final text about the rebel game.
- Part of the following sections: abstract, information warfare as a game, related work, and references.

The author of this dissertation contributed the following in the publication P7:

- The final title of the paper.
- The idea and final text about the evildoer game.
- The idea and final text about meta-strategies.
- The following complete sections: introduction, the overview about game theory, and conclusions.
- Revised versions of the terrorist game and the vandal game.
- Part of the following sections: abstract, information warfare as a game, related work, and references.
- The final editing, including the improvements based on the reviewer feedback.

ABBREVIATIONS

ACK	ACKnowledgement flag in the TCP header
ANSI	American National Standards Institute
AODV	Ad hoc On-demand Distance-Vector routing protocol
AQM	Active Queue Management
ARP	Address Resolution Protocol
BGP	Border Gateway Protocol
BIND	Berkeley Internet Name Domain
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CDF	Cumulative Distribution Function
CDN	Content Delivery Network, or Content Distribution Network
CERT	Computer Emergency Response Team, or Computer Emergency Readiness Team
CPU	Central Processing Unit
CTS	Clear To Send
DCF	Distributed Coordinated Function
DDoS	Distributed Denial of Service
DNS	Domain Name System
DNSSEC	DNS SECurity extensions
DoS	Denial of Service
DPF	Distributed Packet Filtering
DSDV	Destination Sequenced Distance-Vector routing protocol
DSL	Digital Subscriber Line
DSR	Dynamic Source Routing protocol
EULA	End User License Agreement
FIN	FINish flag in the TCP header
FTP	File Transfer Protocol
gTLD	generic Top Level Domain
HIDS	Host Intrusion Detection System
HTTP	Hyper Text Transfer Protocol
i3	Internet Indirection Infrastructure
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IGMP	Internet Group Management Protocol
IIS	Internet Information Server
IOS	Internet Operating System
IP	Internet Protocol
IPsec	Internet Protocol security
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IRC	Internet Relay Chat
IRR	Internet Route Registry
ISO	International Organization for Standardization
ISP	Internet Service Provider
LAN	Local Area Network

MAC	Medium Access Control
MSC	Message Sequence Chart
MSS	Maximum Segment Size
MTU	Maximum Transfer Unit
NIDS	Network Intrusion Detection System
NS	Name Server
ns-2	The network simulator, version 2
OODA	Observe-Orient-Decide-Act
OS	Operating System
OTcl	Object-oriented Tool command language
P2P	Peer-to-Peer
PRNG	Pseudo-Random Number Generator
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RADIUS	Remote Authentication Dial In User Service
RAS	Reliability, Availability, Serviceability
RFC	Request For Comments
ROC	Receiver Operating Characteristics
RLS	Rate Limiting System
RST	ReSeT flag in the TCP header
RTS	Request To Send
RTT	Round Trip Time
SCTP	Stream Control Transmission Protocol
SDL	Specification and Description Language
SOS	Security Overlay Structure
SQL	Structured Query Language
SSL	Secure Sockets Layer
SYN	SYNchronize flag in the TCP header
TCP	Transmission Control Protocol
TLD	Top Level Domain
TTL	Time To Live
UDP	User Datagram Protocol
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network
WMF	Windows MetaFile
WWW	World Wide Web

1 INTRODUCTION

This dissertation studies denial of service (DoS) attacks in computer networks. The goal of these attacks is to prevent availability of network services from their legitimate users. This dissertation presents a structured view on possible attack and defense mechanisms, describes some new defense mechanisms, and provides new information on selecting and evaluating defense mechanisms.

Defending against DoS attacks is network and computer security. As scientific disciplines, network and computer security are relatively new. An indication of this is that even computer security terminology has not yet stabilized [12, 28]. Computer and network security were first studied in the early 1970s, and some of these earliest security papers are listed and available in [26].

Denial of Service attacks are a timely and extremely important research topic. According to the CSI/FBI computer crime and security survey in the United States for the year 2004 [82], DoS attacks are the second most widely detected outsider attack type in computer networks, immediately after virus infections. A computer crime and security survey in Australia for the year 2004 [18] gives similar results.

It is currently not possible to prevent DoS attacks because many of these attacks are based on using ordinary protocols and services in an overwhelming manner. Specific security holes in the victim hosts or networks are thus not necessarily needed. For this reason we can only mitigate these attacks.

1.1 An Overview of Denial of Service Attacks

Denial of Service (DoS) attacks have proved to be a serious and permanent threat to users, organizations, and infrastructures of the Internet [14, 18, 34, 42, 74, 82, 89, 93, 145, 149, 151, 186, 193]. The primary goal of these attacks is to prevent access to a particular resource like a web server [41]. A large number of defenses against DoS attacks have been proposed in the literature, but none of them gives reliable protection. There will always be vulnerable hosts in the Internet to be used as sources of attack traffic. It is simply not feasible to expect all existing hosts in the Internet to be protected well enough (in July 2005 it was estimated that there were approximately 350 000 000 hosts in the Internet [106]). In addition, it is very difficult to reliably recognize and filter only attack traffic without causing any collateral damage to legitimate traffic.

A DoS attack can be carried out either as a flooding or a logic attack [151]. A *flooding DoS attack* is based on brute force. Real-looking but unnecessary data is sent as much as possible to a victim. As a result, network bandwidth is wasted, disk space is filled with unnecessary data (such as spam e-mail, junk files, and intentional error messages), fixed size data structures inside host software are filled with bogus information, or processing power is spent for unuseful purposes. To amplify the effects, DoS attacks can be run in a coordinated fashion from several sources at the same time (Distributed DoS, DDoS).

A *logic DoS attack* is based on an intelligent exploitation of vulnerabilities in the target. For example, a skillfully constructed fragmented Internet Protocol (IP) datagram may crash a system due to a serious fault in the operating system (OS) software. Another example of a logic attack is to exploit missing authentication requirements by injecting bogus routing information to prevent traffic from reaching a victim's network.

There are two major reasons that make DoS attacks attractive for attackers. The first reason is that there are effective automatic tools available for attacking any victim [44, 145], so expertise is not necessarily required. The second reason is that it is usually impossible to locate an attacker without extensive human interaction [50, 72, 201] or without new features in most routers of the Internet [22, 23, 48, 182, 195].

DoS attacks make use of vulnerabilities in end-hosts, routers, and other systems connected to a computer network. The size of a population having the same vulnerability can be large. In July 2003 a vulnerability was found from the whole population of Cisco routers and switches running any version of the Cisco IOS software and configured to process Internet Protocol version 4 (IPv4) packets [45]. This vulnerability made it possible to block an interface, which resulted in a DoS condition without any alarms being triggered. Another example of a large population is the Microsoft Windows Metafile (WMF) vulnerability which was found in December 2005 from all versions of Windows 98, 98SE, ME, 2000, and XP [205]. This vulnerability made it possible to install any malicious software on these hosts, for example, to send DoS attack traffic. User interaction was, however, required to exploit this vulnerability.

1.1.1 DoS Attacks in Real-Life

Real DoS incidents in the Internet between the years 1989 and 1995 were investigated in [95]. The three most typical effects were the following: 51% of these incidents filled a disk, 33% of the incidents degraded network service, and 26% of the incidents deleted some critical files. A single incident was able to cause several types of damages at the same time (the sum of percentages is more than 100%).

The first reported large-scale DDoS attack occurred in August, 1999, against a university [74]. This attack shut down the victim's network for more than two days.

In February 7, 2000, several high-profile Web sites were attacked, which caused them to go offline for several hours [74]. In some cases these DDoS attacks were able to produce about 1 Gbit/s of attack traffic against a single victim.

The backscatter analysis was used to assess the number, duration, and focus of DoS attacks in the Internet [151]. Backscatter is called the unsolicited response traffic which the victim sends in response to attack packets with spoofed IP source addresses. The results indicate more than 12 000 attacks against more than 5000 distinct victims during the 3-week period examined in February, 2001.

Packet fragmentation in real networks was studied in [193]. Bugs in the fragment handling software are exploited in many logic DoS attacks, and the results of this study still indicate the presence of these kinds of DoS attacks in the Internet.

The Coordination Center of the Computer Emergency Response Team (CERT) was attacked in May, 2001. A DDoS attack caused its web site to be available only intermittently for more than two days [108].

Internet service providers (ISP) in UK have been targets for DDoS attacks during 2002. Some customers experienced a downtime of 12 hours in one of these attacks [129].

The Domain Name System (DNS) is a continuous target for DoS attacks [34, 43]. In October, 2002, all root name servers experienced an exceptionally intensive DoS attack. Some DNS requests were not able to reach a root name server due to congestion caused by the DoS attack [39, 207]. Another major DoS attack on June 15, 2004 [6, 80, 97], against name servers in Akamai's Content Distribution Network (CDN) blocked nearly all access to many sites for more than two hours. The affected sites included Apple Computer, Google, Microsoft, and Yahoo. These companies have outsourced their DNS service to Akamai to enhance service performance [111, 120].

In UK online bookmaking, betting, and gambling sites have been extorted with DoS attacks at least during 2004 by unidentified attackers [138].

The Internet-based business service of Al Jazeera was brought down due to a DoS attack in January, 2005 [90]. Al Jazeera provides many Arabic-language news services.

The text-to-speech translation application running in the Sun Microsystem's Grid Computing system was disabled with a DoS attack in March, 2006 [73]. This attack was carried out during the opening day of this service.

1.2 Terminology

This section gives definitions for the most important terms used throughout this dissertation.

1.2.1 General Security Terminology

The subject of this dissertation is related to security in computer networks, that is network security. Generally the word security can be preceded by practically any asset to be protected, such as software security and computer security.

Terms related to security do not unfortunately have any single definition and are seldom defined even roughly [28]. One reason for this is that computer security is still in the early days of the discipline [12]. Dieter Gollmann has described well the problem with security terminology in [79]:

- There is no single definition of security.
- When reading a document, one must be careful not to confuse one's own notion of security with that used in a document.
- A lot of time is being spent (and wasted) in trying to define unambiguous notations for security.

It is not possible, however, to write scientific text without well-defined terms. The following security-related terms are based on existing standards and text books.

Security in general means protection of any kinds of assets [79]. Examples of assets are information, computers, computer networks, equipment, services (such as electricity, heating, air-conditioning), buildings, and humans. Security has been defined in the following way in the Telecom Glossary 2000 which is the standard T1.253-2000 published by the American National Standards Institute (ANSI).

> **Security (current widely accepted definition):** A condition that results from the establishment and maintenance of protective measures that ensure the inviolability from hostile acts or influences [11].

The current widely accepted definition for security means ensuring the inviolability from hostile acts. This is, however, not possible because one cannot protect oneself perfectly from risks, especially when very complex systems, such as the Internet, are involved. It is simply too expensive or even theoretically impossible. We can only prove the existence of security related problems, but not the lack of them. It seems that the term *secure* would better match the meaning of the current widely accepted definition of security.

In this dissertation the following definition for security is used. Parts of this definition are based on ideas from [79], [11], [76], [183], and [184].

> **Security (definition used in this dissertation):** A continuous process towards reasonable resistance against known risks.

In the literature there are slightly different opinions whether non-malicious (unintentional) risks should be included in the definition of security. For example, in [27] security is seen to be related only to attacks which are intentional, but in [11] security is seen to be related to both accidental and intentional risks. The difference is not essential here because this dissertation concentrates on malicious DoS attacks.

The term security is also widely used to refer to guarding, which is close to the definition of security many tens of years ago, before the wide existence of computers and the Internet. According to this point of view, companies in the security industry provide equipment and services, for example, in the area of guarding, 24 hour central monitoring, surveillance cameras, burglar alarms, and fire alarms.

The term information security is being used in the contemporary information society to emphasize the importance of information to nations, organizations, and individuals. A significant activity in an information society is to create, manipulate, and distribute information [214]. Information is thus seen to be the major result for labor. The definition of information security is based mostly on the ANSI's Telecom Glossary 2000. The notion of unauthorized control is taken from Matt Bishop's "Computer Security".

> **Information security:** A continuous process towards reasonable protection of information against unauthorized disclosure, transfer, modification, destruction, or control [11, 27].

Information security does not require information to be processed with computers or to be in electronic format. However, information is typically processed with networked computers. Any computer should have measures to protect information processed and contained in it. The definition of computer security is based mostly on "Computer Security" written by Dieter Gollmann.

> **Computer security:** A continuous process towards a reasonably good prevention and detection of unauthorized actions by users of a computer system [79].

Most existing computers are connected to networks, so network security is important. The definition of network security is based mostly on "Computer System and Network Security" written by White, Fisch, and Pooch.

> **Network security:** A continuous process towards meeting reasonable objectives of providing confidentiality, integrity, availability, and access for legitimate users of network resources ([212], p. 143).

The difference between computer security and network security has blurred as most computers (or hosts) are connected to networks, and practically any host can be accessed through a network in a similar way as sitting physically in front of it [183].

Defending against DoS attacks is mostly network security, as these attacks are carried out from a remote location through a network. Separate network security devices are typically required for defending against DoS attacks. Defending against DoS attacks can also be seen to be part of computer security, as DoS attacks ultimately exploit weaknesses in computer systems.

Information security and variations of it (such as computer and network security) have three fundamental objectives: confidentiality, integrity, and availability [27, 79, 212]. These are defined as follows in "State of the Practice of Intrusion Detection Technologies" written by Julia Allen et. al. from the Carnegie Mellon University:

> **Confidentiality:** The property that information is not made available or disclosed to unauthorized individuals, entities, or processes [9].

Integrity: The property that data has not been changed, destroyed, or lost in an unauthorized or accidental manner. For systems, integrity is defined as the quality that a system has when it can perform its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation [9].

Availability: The property of a system or a system resource being accessible and usable upon demand by an authorized entity, according to performance specifications for the system [9].

A good level of availability cannot be provided without reliable systems, especially in safety critical environments, such as aviation and intensive health care. Reliability can thus be seen as an aspect of availability, even though availability and reliability in principle mean different things. This difference is clearly indicated in the reliability-availability-serviceability (RAS) system design concept where availability is strictly separated from reliability [194]. In the RAS concept, a system can emphasize either reliability or availability, depending on the final end-user requirements. Safety critical environments cannot tolerate systems failing during their operation, so reliability is more important than average availability. Many telecommunications systems, however, tolerate failures, but the failures should be fixed very quickly. Here availability can be improved by either using more reliable systems or by fixing problems faster (serviceability). This kind of average availability has been defined in the ANSI's Telecom Glossary 2000 [11] as the ratio of the total time a functional unit is capable of being used during a given interval to the length of the interval. It is specified in decimal fractions, such as 0.9999 (four nines). An example of applying average availability on network services is in [61]. Availability as a decimal fraction, however, describes only part of the truth if used to quantify availability in the security context because it indicates a raw average value without any details about when a system is not available.

To overcome problems in overlapping notations, such as between availability and reliability, the generic term dependability is used to combine availability, reliability, safety, confidentiality, integrity, and maintainability [19]. Security, with its three fundamental objectives, is thus part of dependability.

Dependability: The ability to deliver service that can justifiably be trusted [19].

Finally some widely used definitions related to security will be defined. These definitions come from "Critical Infrastructure Glossary" created by the U.S. National Infrastructure Institute, from "State of the Practice of Intrusion Detection Technologies" written by Julia Allen et. al. at the Carnegie Mellon University, and from the ANSI's Telecom Glossary 2000.

Vulnerability: A flaw in security procedures, software, internal system controls, or implementation of an (information) system that may affect the integrity, confidentiality, accountability, and/or availability of data or services. Vulnerabilities include flaws that may be deliberately exploited and those that may cause failure due to inadvertent human interactions or natural disasters [40].

Exploit: A program, script, or technique that makes it possible to take advantage of a vulnerability in a system [9].

Threat: Any circumstance or event that could harm a critical asset through unauthorized access, compromise of data integrity, denial or disruption of service, or physical destruction or impairment [40].

Risk: The probability that a particular vulnerability is exploited by a particular threat weighted by the impact of that exploitation [40].

Risk management: The process concerned with the identification, measurement, control, and minimization of security risks in information systems to a level commensurate with the value of the assets protected [11].

More security related terms can be seen in [96] which defines a taxonomy for computer security incidents.

1.2.2 Denial of Service Terminology

International Organization for Standardization (ISO) has given the following definition for denial of service (DoS) in the standard ISO 7498-2:1989.

Denial of service: The prevention of authorized access to resources or the delaying of time-critical operations ([104] in [79]).

Examples of resources in this definition are network bandwidth, processing capacity, disk space, memory, and static memory structures [41].

An attack (which does not have to be successful) is defined in the ANSI's Telecom Glossary 2000 [11] to be an attempt to violate security. This will be used as the basis for defining a DoS attack.

Denial of service attack: An intentional attempt to prevent or degrade availability of any resources.

This thesis concentrates on malicious DoS attacks, but a DoS condition can also result from unintentional human errors, design faults, or software bugs. For example, at least 500 000 Netgear routers overloaded a single Internet time server due to an unintentional software design error [169].

It is not always possible to say exactly what in practice is a DoS attack. For example, spam e-mails constitute approximately 70% of incoming e-mails [142], but spam is generally not considered to be DoS, even though a large amount of spam induces delay for end users. Added delay as such, however, is one characteristic of DoS attacks.

The term intrusion is used to denote a successful attack:

Intrusion: Successful unauthorized usage or misuse of a network or computer system [9, 171].

As expressed in this thesis, it is not possible to prevent DoS attacks reliably. Instead, we talk about mitigating these attacks. According to the Webster's dictionary, the verb mitigate means the following.

To mitigate: To lessen in force or intensity, or to make less severe [13].

DoS attacks are one manifestation of computer crime [82, 128], other manifestations including malicious software, spam, spyware [86], fraud, phishing [144], abuse of networks, unauthorized access, and theft of proprietary information. The definition for computer crime in the ANSI's Telecom Glossary 2000 is the following.

Computer crime: A violation of law committed with the aid of, or directly involving, a data processing system or network [11].

DoS attacks can be classified based on the number of sources included in an attack [93]. In a basic DoS attack the attacker uses a single source host to send attack traffic

to a victim. A distributed DoS (DDoS) attack involves more than one sources of attack traffic:

Distributed denial of service attack: An attempt to prevent or degrade availability of any resources by using multiple source hosts at the same time to send attack traffic.

Typically the participants in a DDoS attack form a hierarchical DDoS network where an attacker controls a few masters (or handlers), which in turn control a much higher number of agents (or daemons or zombies or bots) to carry out a real attack against a victim. These are defined as follows.

Agent (or daemon or zombie or bot): A compromised host used to send attack traffic in a DoS attack.

Master (or handler): A compromised host used to control the operation of a large set of agents.

DDoS network: A hierarchically structured set of masters and agents to make it easier to control a DDoS attack by an attacker.

DoS attacks may be either destructive or degradative ([95], p. 160).

Destructive DoS attack: Prevents the availability of a resource completely.

Degradative (non-destructive) DoS attack: Reduces the performance of a resource.

A destructive DoS attack can, for example, crash a system or fill disk partitions. In these cases human intervention is typically needed for recovery. A degradative DoS attack will typically cause only temporary problems, and a system will recover automatically as soon as an attack terminates. An example of a degradative DoS attack is a flooding attack overloading a network link or a host central processing unit (CPU). A prolonged high-bandwidth flooding attack, however, may have unexpected results, such as system crashes.

A DoS attack can be seen to have two different directions.

Inward DoS attack: From the victim point of view a DoS attack consists of incoming attack packets.

Outward DoS attack: From the attack source point of view a DoS attack consists of outgoing packets.

DoS attacks consist of two major phases [93]. Both of these phases make use of deficiencies in the design or implementation of applications, protocols, and the Internet architecture [146].

Deployment phase: Installation of a malicious program on a set of compromised hosts to be used later as a source for DoS attack traffic.

Attack phase: Coordinated transmission of attack traffic against a victim.

1.3 Research Problem

Mitigating DoS attacks is difficult especially due to the following problems:

- Very little has been done to compare, contrast, and categorize the different ideas related to DoS attacks and defenses [156]. As a result it is difficult to understand what a computer network user needs to do and why to mitigate the threat from DoS attacks.

- There are no effective defense mechanisms against many important DoS attack types.
- There is no guidance on how to select defense mechanisms.
- Existing defense mechanisms have been evaluated according to a very limited criteria. Often relevant risks have been ignored (such as in [3]) or evaluations have been carried out under ideal conditions (such as in [199]). No research publications exist for giving a systematic list of issues related to defense evaluation.

1.4 Objective of the Research

The objective of this research is to help any network user in mitigating DoS attacks in IP-based networks. This dissertation concentrates especially on the following areas:

- One should understand existing attack mechanisms and available defense mechanisms, and have a rough idea about the benefits (best-case performance) of each defense mechanism.
- One should acknowledge possible situation dependency of defense mechanisms, and be able to choose the most suitable defense when more than one defense mechanisms are available against a specific attack type.
- One should evaluate defense mechanisms in a comprehensive way, including both benefits and disadvantages (worst-case performance), as an attacker can exploit any weakness in a defense mechanism.

Knowledge of all of these issues is necessary in successful mitigation of DoS attacks. Without knowing how a specific defense mechanism works under different possible conditions and what the real benefits and weaknesses are, it is not possible to assure the suitability of a defense mechanism against a certain type of a DoS attack.

1.5 Research Methodology

Research methodologies used in this dissertation are primarily based on simulating different attack scenarios, but measurements, mathematical modeling based on game theory, and requirement specification are also used in the publications. The used research methodologies are explained in detail later in this dissertation when describing each contribution.

1.6 Scope of the Research

This dissertation studies DoS attacks in computer networks using the Internet Protocol (IP), namely the Internet and mobile ad hoc networks. DoS attacks in the physical world are not studied here.

Majority of the publications in this dissertation concentrate on the fixed (wired) Internet, but most of the presented attack and defense mechanisms are applicable to wireless networks, too. Publications P3 and P4 concentrate only on specific attacks and defenses in wireless mobile ad hoc networks.

The emphasis of this research is on DoS attacks in general, and DDoS attacks are treated as a subset of DoS attacks. DDoS attacks are based on the same mechanisms

as basic DoS attacks, but there is one exception during the deployment phase. A DDoS tool needs to be installed on many vulnerable hosts. The installation of DoS software on a single vulnerable host is, however, a common prerequisite for most DoS attacks. Thus attack and defense mechanisms described in this dissertation are applicable to both DoS and DDoS attacks.

1.7 Contributions

The main contributions of this dissertation are the following.

1. A comprehensive and well-structured description is given about

   - what DoS attacks are,
   - how DoS attacks can be carried out in IP networks, and
   - how one can defend against DoS attacks in IP networks.

   A good understanding of existing attack mechanisms and available defense mechanisms is a prerequisite for succeeding in mitigating these attacks cost-effectively.

2. An overview of an organized approach for selecting a comprehensive set of defense mechanisms against DoS attacks is given and it emphasizes

   - the importance of basic security mechanisms at every host in the Internet,
   - the importance of risk management in choosing additional defenses when basic defenses are not enough, and
   - the necessity of implementing new defenses against such important DoS attacks for which there are no existing defenses.

3. A new defense mechanism for protecting organization-specific name servers is described and simulated.

4. A new DoS attack in mobile ad hoc networks is presented. This attack is called the range attack, and it is based on degrading the performance of ad-hoc routing by modifying the transmission range of a wireless node in a periodical fashion.

5. Cross-layer security is shown to be a promising defense approach, and two cross-layer designs are given against the range attack in ad hoc networks.

6. The resilience of three existing ad-hoc routing protocols against the range attack was studied. It was found out that this resilience is situation dependent. Depending on the quality of service requirements of the primary applications, different ad-hoc routing protocols were found to be resistant against the range attack.

7. Game theory is applied in some information warfare scenarios, and the example games show that game theory can be a useful tool for making it easier to understand the suitability of many defense strategies. Even though information warfare is not related only to network attacks, DoS attacks are an important tool for many subtypes of information warfare.

8. A new possible attack mechanism based on a game theory approach to perception management was suggested. Defense strategy optimization is based on the correct perception of involved payoffs. If these payoffs are not seen in a correct way by a defender, wrong defense strategies will be chosen which will make it easier for an attacker to succeed. This perception management is based on meta-strategies and applying the principles of the Observe-Orient-Decide-Act (OODA) loop.

9. A taxonomy of criteria is presented for evaluating defense mechanisms. Evaluation of most defense mechanisms has been based on very limited criteria, but the presented taxonomy brings out important issues to be considered during an evaluation process, such as the quality of service available for false positives and any possible collateral damage.

10. The effectiveness of rate limiting in mitigating flooding DoS attacks against servers was evaluated. It was found out that up to 50% of incoming Transmission Control Protocol (TCP) packets can be discarded without any clear problems in the quality of service of any legitimate flows downloading information from the victim site.

1.8 Structure of the Thesis

This thesis consists of the following chapters.

- *Chapter 2, Defense Mechanisms against Denial of Service Attacks:* This chapter describes existing attack and defense mechanisms. The basic defenses required by all network users are listed. The importance of risk management is pointed out for selecting a cost-effective set of defenses providing defense in depth.

- *Chapter 3, Situation Dependent Selection of Defense Mechanisms:* This chapter shows how the effectiveness of some defense mechanisms can be dependent on the QoS requirements of primary applications. Game theory is also applied in selecting defense strategies.

- *Chapter 4, Evaluation of Defense Mechanisms:* This chapter describes issues that should be considered when evaluating defense mechanisms. For example, possibility for misuse and any collateral damage must be carefully examined.

- *Chapter 5, Discussion:* This chapter discusses some problems in carrying out risk management, and analyzes shortly the credibility of results from simulations and game theoretic models.

- *Chapter 6, Conclusions.*

2 DEFENSE MECHANISMS AGAINST DENIAL OF SERVICE ATTACKS

In risk management one must understand the most important risks and decide how to mitigate them. Risks can be either accepted as such, mitigated by using one or more defense mechanisms, or transferred to third parties (such as with insurances). The primary goal is to ensure business continuity and, at the same time, keep the associated costs at a reasonable level.

Effective risk management, however, is not possible without a good knowledge in existing attack mechanisms and available defense mechanisms. A widely exploited attack mechanism can be associated a high risk requiring effective mitigation. Completely different actions should be taken in a risk management process when no defense mechanisms exist against a specific attack, and when effective defense mechanisms can be easily deployed.

Information about individual DoS attack and defense mechanisms exist but this information is scattered in the literature. The section 2.1 describes existing DoS attack mechanisms and available defense mechanisms. It emphasizes the importance of defense in depth for mitigating the most relevant risks. Risk management is seen here as the primary tool for achieving a cost-effective set of defenses. This section is based on the publication P1.

The section 2.2 describes a new defense mechanism for protecting authoritative name servers providing information about end-hosts in the Domain Name System (DNS). When there are no existing defense mechanisms against an important attack type, new defense mechanisms must be designed due to the very high risk of being attacked successfully. As the DNS is a prerequisite for accessing practically any service in the Internet, it is considered as one of the most critical services in the Internet. Many of the DoS attacks currently identified in the Internet are against the DNS, as described earlier in the section 1.1.1. This section is based on the publication P2.

The section 2.3 describes two different cross-layer designs for mitigating DoS attacks in mobile ad hoc networks. Cross-layering is a relatively new approach in implementing security. It is based on sharing important information between different protocol layers. Correlation of information at different protocol layers makes it easier to detect and defend against attacks. Regarding the two simulated cross-layer designs, cross-layer security is shown here to be a promising approach for implementing defense mechanisms. This section is based on the publication P3.

2.1 Mitigating Denial of Service Attacks

A comprehensive and structured description about existing DoS attack and defense mechanisms is given here. This section is divided in five parts. The first part describes the role of worms and viruses in creating programmable sets of source hosts for DoS attacks. The second part gives a structured view on existing generic DoS attack mechanisms. The third part describes how to handle DoS attacks in general at a victim site. The fourth part gives a structured list of many existing defense mechanisms against major attack types. The final part in this section discusses the importance of risk management in the process of actually selecting defense mechanisms.

2.1.1 Worms and Viruses as Deployment Tools

To get anonymity attackers use compromised hosts for sending DoS attack traffic. For this reason, any DoS attack can be seen to have two phases: the deployment phase, and the attack phase. During the deployment phase an attacker installs a DoS tool on as many compromised hosts as needed. The more there are source hosts for attack traffic, the more effective an attack is. Most real-life DoS attacks ending up in the news headlines are actually DDoS attacks.

Worms are typically used for the deployment phase, but viruses using social engineering are also possible tools. Worms are self-propagating malicious software which do not require any human interaction. Advanced worm technology is able to infect fast even small populations randomly located in the Internet. A sudden but very intensive attack is thus possible [196].

Worms can exploit vulnerabilities in any operating system (OS) platform. The Code Red I v2 worm infected more than 359 000 hosts within 24 hours by exploiting a vulnerability in the Microsoft IIS web server [149]. The Slammer worm infected 75 000 hosts (more than 90% of the whole population) in 10 minutes by exploiting a vulnerability in the Microsoft SQL server [148]. The Slapper worm infected approximately 6 000–16 000 hosts by exploiting a vulnerability in the OpenSSL software used by many Linux distributions and Apache Web servers [15]. The Witty worm infected 9 000 hosts (75% of the population) by exploiting a vulnerability in the ISS RealSecure firewall [192].

Botnets are nowadays available for any malicious purposes in the Internet. Botnets are groups of compromised hosts that can be controlled remotely by a malicious person, typically via Internet Relay Chat (IRC) channels. Any malicious software can be downloaded and executed in these hosts. Botnets can be hired for attack purposes such as DoS attacks [103, 178]. In a botnet-related study [71], 60 active botnets were identified in March 2005 with at least 300 000 compromised hosts belonging to these botnets. In real-life there are more botnets and compromised hosts than were identified. While hosts in botnets are disinfected and patched, new botnets with new hosts are continuously created, for example, by using viruses, worms, and existing backdoors. Botnets offer a continuous source for intensive DoS attacks [142].

2.1.2 Attack Mechanisms

Once DoS software has been deployed, an attacker is able to proceed to the final attack phase. An actual attack will consist of a flooding or a logic attack against a single victim. A structured overview of the contents in this subsection 2.1.2 is shown in the figure 2.1.

2.1.2.1 Coordination of DDoS Agents

In case of a DDoS attack an attacker must first coordinate all DDoS agents to attack in unison for effectiveness reasons [58]. This coordination requires attack commands to be transmitted to every agent through a *control channel*. There are several choices for transmitting this control channel information, usually in an encrypted form. In [103] it is stated that IRC channels are the most widely used control channel mechanism (see also [142]) but web-based channels are used in an increasing fashion by many botnets. The Slapper worm found in September 2002 contains the ability to execute DDoS attacks, and the coordination traffic is carried over a specific peer-to-peer (P2P) protocol where all commands and responses are transmitted through a random chain of agents to make the DDoS network more robust against tracing [15]. TCP or User Datagram Protocol (UDP) are used by the simplest DDoS tools. Many ordinary

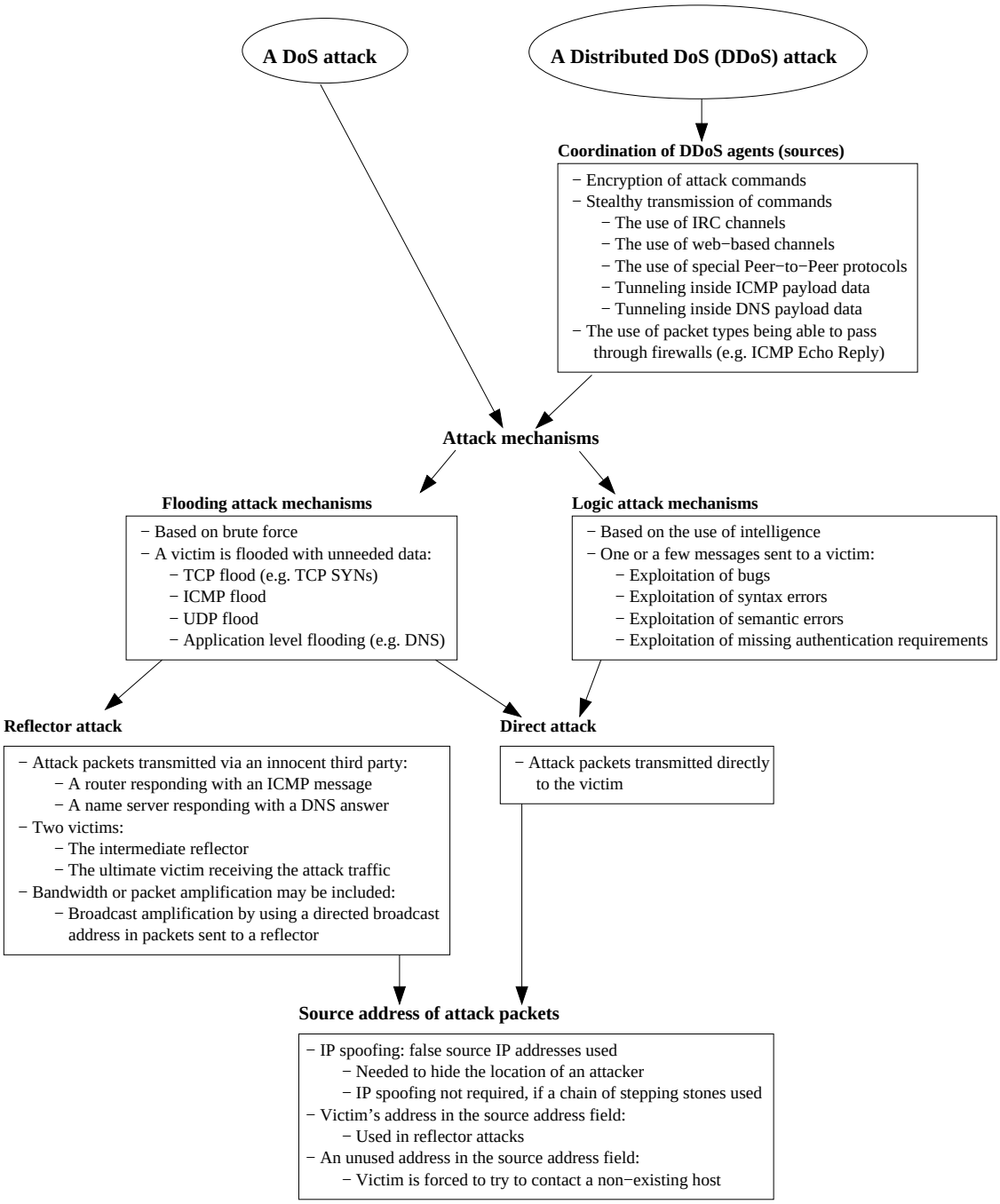

Figure 2.1: The major attack mechanisms used by DoS or DDoS attacks.

protocols provide a way to tunnel commands, for example inside the Internet Control Message Protocol (ICMP) or the Domain Name Service (DNS) payload data. Especially packets containing a reply (such as an ICMP *Echo Reply* or a DNS answer) have a higher probability of passing through firewalls.

2.1.2.2 IP Spoofing

A basic mechanism in all DoS attacks to hide the location of an attacker is *IP spoofing* which means sending packets with a false source IP address. A certain kind of a value in the source IP address field is also a prerequisite for some DoS attacks. Setting the victim's address in the source field makes it look like the packet was originally sent from the victim. Setting an unused IP address in the source field forces the victim to try to contact with a non-existing host.

It is possible to carry out DoS attacks without IP spoofing if an attacker has compromised enough hosts, or if a chain of compromised hosts is used. Tracing an attacker through a chain can be made difficult or impossible by selecting the compromised hosts of a chain from sites with poor security practices or from countries with suitable legislation. In this kind of a case IP spoofing is not necessarily required for protecting an attacker.

2.1.2.3 Flooding Attack Mechanisms

Flooding DoS attacks are generally divided into direct and reflector attacks [48]. In *direct attacks* spoofed packets are sent directly to the victim. In *reflector attacks* packets with the victim's address in the source IP address field are sent to an innocent third party, which in turn will send the reply to the victim. Examples of innocent third parties are web servers, DNS servers, and routers. Reflector attacks have thus at least two victims at the same time [67].

Basically any ordinary protocol behavior can be utilized as the underlying mechanism for flooding attacks. Any protocol layer is suitable for attack purposes.

Direct attacks use typically only few mechanisms, namely TCP SYN flooding (a flood of TCP packets having the SYNchronize bit set for initiating the three-way handshake), ICMP `Echo` flooding, or sometimes UDP data flooding [50]. In TCP SYN flooding the victim is sent SYN packets with an address of a non-existing host in the source IP address field, which will result in lots of half-open connections that fill static data structures and prevent legitimate connections. These half-open connections will timeout typically in 75 seconds [85]. In ICMP `Echo` flooding the victim is forced to handle a large number of ping-packets. In UDP data flooding one possible objective is to connect `chargen`- and `echo`-ports between two victims. These attacks do not consume resources permanently, so at least in theory, the victim should be able to continue serving legitimate users normally after the attack is over. Some flooding attacks may, however, have longstanding effects. For example, IP fragment flooding may consume all available memory for storing partial IP datagrams, after which hosts may crash due to unavailability of free memory.

Reflector attacks utilize any protocol behavior where an attack packet triggers a response packet to be sent to the ultimate victim [165]. Reflector attacks can also include the use of a technique called *bandwidth or packet amplification*. The innocent third party will either reply with a longer packet or with several packets to a single attack packet, respectively. Ultimately a reflector may be able to send a continuous stream of packets to a victim, such as in the bombing attack [17] where an attacking node uses mobility features to redirect a packet stream to a victim node. A form of packet amplification is the broadcast amplification where an attack packet is sent to a subnet directed broadcast address. All hosts receiving the attack packet will send

their own response to the ultimate victim. A well-known example of this is the smurf attack [157] where a single ICMP `Echo` is amplified into several ICMP `Echo Reply` packets.

Flooding attacks against routers can be effective because routers are usually optimized for forwarding traffic instead of handling data sent directly to them [44]. Flooding attacks against DNS can cause widespread Internet slowdowns or effective outages.

A high-bandwidth stream of packets is not necessarily required for a flooding attack, especially when attacking victims having a low-bandwidth connection to the Internet (such as users behind mobile data connections). Bandwidth less than a typical analog modem speed may be enough in exploiting deficiencies in the implementation of data structures [59]. These kinds of *algorithmic complexity attacks* can, for example, degenerate binary trees and hash tables into linked lists. Due to the requirement of intelligence in selecting the attack traffic, these algorithmic complexity attacks might as well be classified as logic attacks described in the next subsection.

2.1.2.4 Logic Attack Mechanisms

The objective of logic DoS attacks is to build a small number of specific packets exploiting vulnerabilities which cause the victim to do abnormal things. The packets are normally sent directly to a victim because special knowledge about a vulnerability is required for building the attack packets. There is a wide variety of logic attacks. Typically an attack is based on more than one of the following issues at the same time:

- Exploitation of bugs: All software contains bugs which can, for example, cause a host to crash due to errors in dynamic memory structure handling, like in the Teardrop-attack based on overlapping IP fragments ([157] pp. 54–55).
- Exploitation of syntax errors: Implementations are not always able to handle syntactically incorrect data, like in the Internet Group Management Protocol (IGMP) attack based on malformed headers [188].
- Exploitation of semantic errors: Implementations may process normally all syntactically correct messages, even if these messages are semantically incorrect. For example, in DNS cache poisoning a bogus mapping (a DNS answer) may be appended to an innocent looking query message [44].
- Exploitation of missing authentication requirements: Lack of authentication makes it possible to enter false information into many protocols (such as a dynamic routing protocols) and services (such as the DNS) [160].

Implementations of protocols may include non-standard or missing features which can be exploited. For example, some implementations of the TCP state machine are non-standard and include extraneous state transitions, or not all states have well-defined timeouts. In these kinds of hosts it is possible to force the state machine to enter a state which cannot be exited or has a very long timeout. These vulnerabilities in some implementations can be exploited by TCP SYN-FIN packet streams or by replying to a TCP SYN packet with a TCP SYN packet [85].

Logic attacks against routers and security devices can affect large parts of the Internet infrastructure and enable other kind of attacks, when part of the defenses are no more operational. DoS attacks against the Internet infrastructure are becoming more common. The routing infrastructure is considered to be an excellent target for DoS attacks because an attacker is able to cause severe network outages without a

significant effort by simply injecting false routing information [160]. DNS is another infrastructure service vulnerable to DoS attacks, for example when a domain is hijacked [44].

2.1.3 Handling DoS Attacks in General

In general, defending against DoS attacks consists of preparation, detection, and reaction phases [94]. The *preparation phase* includes, for example, creation of a security policy, selection of good security protocols and products, installation of required security devices, separation of the critical services from each other, overprovisioning of capacity, monitoring ongoing operations to learn to know what is normal behavior, training of analysis capabilities, creation of an incident response plan, and making a cooperation plan with an Internet Service Provider (ISP). It should be noticed that incident response should not depend on the proper operation of a system under attack [123]. For example, a local compromised host can give unreliable information about an incident, it may be impossible to reach the real administrators of a remote compromised host, or it can be difficult to contact participants through an overloaded network.

The *detection phase* should be automatic. To be able to react as fast as possible, an *early warning system* is required, which means detecting DoS attacks as early as possible [135]. The later an attack is detected the less administrators have time to react before clear damages are caused to legitimate traffic, for example, in the form of decreased availability of services.

The *reaction phase* consists of two subphases, namely characterization and mitigation. In the *characterization phase* the victim must verify if an attack is really going on, and the victim must also analyze the attack to be able to find distinguishing characteristics of the attack traffic. A good understanding of the nature of the attack is required for the *mitigation phase*, in which the victim installs the required defenses, such as filters to block attack traffic.

2.1.3.1 Detection of DoS Attacks

Intrusion Detection Systems (IDSs) are tools for detecting intrusive network or host activity, and announcing alerts [154]. These systems can be divided in two major classes. *Network Intrusion Detection Systems* (NIDSs) are passive nodes which have access to all traffic in a network link. *Host Intrusion Detection Systems* (HIDSs) are applications which analyze log files and other security related information and try to detect intrusive use of a single host. NIDSs and HIDSs do not have equal advantages and disadvantages, so an important site needs to employ a combination of them.

There are two distinct analysis methods to decide, whether an intrusion has been found or not. Signature-based *misuse detection* tries to locate known patterns from the incoming sensor data, much like the existing antivirus software does. The major problem with misuse detection is the requirement for exact signatures (fingerprints) of attacks, which makes these kinds of systems reactive and places strict requirements on the speed of signature updating [153]. This means inability to detect new or even slightly modified attacks. *Anomaly detection* is based on observing significant deviations from typical or expected behavior of systems or users [135]. The major problem with anomaly detection is the difficulty in defining what is typical or expected behavior and what is not [9]. Anomaly detection systems can detect some new or modified attacks.

2.1.3.2 Effectiveness of DoS Attack Detection

IDSs have proved to be necessary tools for detecting attacks [157]. An IDS can provide log files and traces of network traffic which can be used to get further information about the involved hosts and the amount of damages. Later this information can be used as a proof of an attack in lawsuits. IDSs are used in an increasing fashion to show the presence of attacks against corporate and even home networks.

Detection of DoS attacks is not simple because these attacks exploit features of ordinary protocol behavior. By choosing an attack method suitably an attacker has the possibility of escaping the detection by an IDS. In general there are three possibilities for this.

First, an IDS may not be able to collect all desired information. An IDS may simply be not effective enough to handle all available data, or implementation bugs may make it inclined to crashing. In [153] several signature-based NIDSs were compared. Many systems were not able to capture all network frames or were not robust enough crashing from time to time. An IDS can be overloaded intentionally by an attacker, so that at least part of the attack traffic is dropped by the IDS. This makes it possible to evade detection [164]. An IDS is also susceptible to a crash attack, in which the attacker knocks down whole or part of an IDS by utilizing some vulnerability [164]. Flooding and logic DoS attacks can thus be used against an IDS to prevent it from collecting information.

The second reason for an IDS being not able to detect all intrusions is the possibility for an *evasion* or *insertion* attack against an NIDS. These attacks exploit ambiguities in the payload of packets [88]. For example, an NIDS may reassemble overlapping IP fragments in a different fashion than an end-host. This can prevent an NIDS from seeing complete signatures broken down in multiple overlapping fragments [164]. In an insertion attack an NIDS accepts a packet that an end-system rejects or does not receive. In an evasion attack an NIDS rejects a packet that an end-system accepts. Both insertion and evasion attacks break signatures and thus prevent an NIDS from recognizing an attack. Insertion and evasion attacks can be implemented, for example,

- by using a low Time To Live (TTL) value not reaching the end-system,
- by using a packet longer than the MTU of the end-system network and setting the `Don't Fragment` flag in the IP header,
- by using source-routed packets discarded at the end-system,
- by exploiting different IP reassembly timeouts,
- by sending overlapping IP fragments,
- by sending overlapping TCP segments, or
- by using special combinations of TCP flags not accepted by every TCP/IP stack implementation [171].

The third reason for an IDS being not able to detect all intrusions is the inability to recognize intrusions correctly even from a complete and correct sensor data. This can happen if attack traffic resembles legitimate traffic too much. The frequency of false positives (false alerts) is important because they need to be checked by humans. Too many false positives during a specific time interval makes an IDS completely useless. The number of false positives can be reduced at the expense of the number of true positives (detected real intrusions). If an IDS is tuned to create fewer false positives,

it also gives less true positives. In other words, less true attacks are detected. The relation between false positives and true positives can be represented as a *Receiver Operating Characteristics (ROC) curve*. Some ROC curves of real IDSs can be seen in [131] and [64]. An ROC curve for an IDS clearly indicates what kind of effect reducing the number of false positives has on the number of detected true positives. In one study the average detection rate for known DoS attacks was about 80%, but for new or slightly modified DoS attacks the average detection rate was only about 20%, when the maximum number of false positives was set to approximately 10 a day [131].

Examples of two theoretical ROC curves are shown in the figure 2.2 where the continuous line shows an ROC curve for an IDS based on anomaly detection, and the dotted line shows an ROC curve for an IDS based on misuse detection. In this figure the two ROC curves are not based on any measurements, but they simply illustrate possible shapes of ROC curves for different kinds of IDSs. In the figure 2.2 the IDS based on misuse detection operates practically in a single point at (0.35,0.65) because this kind of an IDS either detects an attack or not. An IDS based on anomaly detection, on the other hand, assigns different warning values for suspected attacks. Weak signs of an attack are indicated with lower warning values and clear signs of an attack are indicated with higher warning values. One point of the ROC curve is calculated by selecting one of the possible warning values as a threshold value. The probability of a true positive is the number of those real attacks having a warning value greater or equal to this threshold value, divided by the total number of real attacks. The probability of a false positive is the number of those legitimate sessions/connections having a warning value greater or equal to this threshold value, divided by the total number of legitimate sessions/connections. By looping this threshold value over all warning values, it is possible to create a complete ROC curve ([213], pp. 26–34). The shape of an ROC curve depends on the test material, so one should not use this figure 2.2 to directly compare the effectiveness of misuse and anomaly detection. An important issue here is the reliability of signatures in misuse detection. In case of antivirus products, for example, malicious software typically contains long byte streams which can be used reliably for detecting viruses. DoS attacks, however, are based on exploiting ordinary features in existing protocols and services. As a result, it is difficult to find good signatures from short protocol packets, especially in case of flooding DoS attacks. This makes it difficult to detect DoS attacks reliably by using misuse detection. If the test material for creating an ROC curve emphasizes new DoS attacks (as is expected in the figure 2.2), the ROC curve for an anomaly based IDS most probably involves higher probabilities for true positives most of the time when compared to the ROC curve for a misuse based IDS. On the other hand, if only well-known DoS attacks with reliable signatures are used for creating an ROC curve, a misuse based IDS should perform better than an anomaly based IDS.

2.1.3.3 Reaction Against Detected DoS Attacks

As was shown in the previous subsection, detection of DoS attacks is not a simple task. An experienced attacker can hide DoS activity. This has implications on the reaction phase. Automatic reaction mechanisms are fast, but the problem with false positives must be tackled somehow. Typically human intervention is required at some moment of time.

A prerequisite for the mitigation of DoS attacks is a detailed knowledge of the details of an ongoing attack (the characterization subphase). If the exact signature of attack traffic is not known, such as in the case of a flooding DoS attack, mitigation mechanisms can easily cause damage for legitimate users.

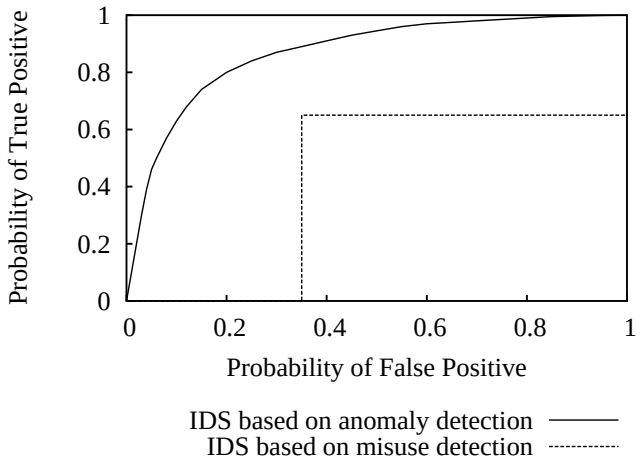

Figure 2.2: Example ROC curves for IDSs trying to detect new or modified DoS attacks. The performance difference between the two kinds of IDS mechanisms varies depending on the test material used for creating an ROC curve.

A widely used way to react against DoS attacks has been a labor-intensive manual procedure by network administrators, which means manual input debugging to locate routers on the path of the attack traffic step by step towards the attack source, and manual installation of packet filtering or rate-limiting rules in these routers handling attack traffic [199].

An automatic mechanism is needed for a quick early reaction. The implementation of a reaction mechanism can reside either in an end-host or a network security device. When comparing the two implementation locations, network security devices are better places for reacting against inward flooding and many logic DoS attacks because the attack must be mitigated as near the actual source as possible. Host implementations, however, have an advantage in reacting better to outward attacks and the DoS tool deployment phase.

Reaction mechanisms concentrate on mitigating the effects of a DoS attack. A reverse attack where a victim starts fighting back, is typically not feasible due to IP spoofing. A self-defense or a revenge would only hit another innocent site. A known self-defense mechanism can even be used as a logic reflector DoS attack against a final target.

2.1.4 Defense Mechanisms

It is claimed here that no single defense is enough against a DoS attack. A comprehensive set of defenses has to be utilized to get *defense in depth* ([9], pp. 96–97). If one layer of defense fails, the other defense layers still have the possibility to detect and mitigate an attack. A successful intrusion requires all defense layers to fail. Defense in depth is a widely used term also in human safety [176].

The Internet is as secure against DoS attacks as its weakest hosts. As there will always be exploitable hosts accessible from the Internet, DoS attacks can be launched even against a site with a comprehensive set of defenses. In this respect all hosts in the Internet are dependent on the protection of other hosts. Detection of compromised hosts (deployment phase) is thus as important as detection of DoS attacks (attack phase).

The earlier a DoS attack can be detected and mitigated, the better. Defenses for the deployment phase are important, because they can be used to prevent or detect installation of DoS tools.

An overview of some available defenses for both the deployment and the attack phase is given here. The list of defenses is definitely not exhaustive, but gives an understanding of the variety of defenses available. The order in which defense mechanisms have been listed is not relevant.

The figure 2.3 gives a structured overview of all of the defense mechanisms described in this subsection 2.1.4. This figure is organized as a simple tree. Defenses against DoS attacks can be selected by traversing this tree from the root towards the leaves. The root of this tree contains necessary defenses required by any network element, like an end-host, a router, or a name server. If basic defenses are not enough and DoS attacks need to be mitigated, one typically needs mechanisms to detect intrusions more effectively. Defense mechanisms against both the deployment and the attack phase are shown in their own boxes in the tree.

2.1.4.1 Basic Defenses

This section lists basic defense mechanisms that should be taken care of by any organization or individual having hosts connected to the Internet. All defense mechanisms listed here are effective also in preventing or making it more difficult to exploit logic DoS attacks.

All unnecessary services should be removed. The less there are applications and open ports in hosts, the less there are vulnerabilities to be exploited by an attacker. Default installations of operating systems often include many applications not needed by a user. Especially many home-users do not even know what services are running on their systems. A vulnerability scanner can be used to detect what network services (open ports) are available in a network.

A firewall (or a router with similar abilities) should be used to control access to a network. Even if there are many services available from local hosts, not all of these services need to be accessible from the public Internet. All possible connections to a network or a host should be protected equally. Fixed connections via Internet Service Providers (ISP) are typically taken care of, but there are several other access paths, such as the Public Switched Telephone Network (PSTN) and Wireless Local Area Networks (WLAN). Guidance for configuring a router to mitigate DDoS attacks is available from manufacturers, such as in [51].

All relevant security patches should be installed timely. The DDoS tool deployment phase and many logic DoS attacks are based on exploiting vulnerabilities in host software. Removing known security holes prevents re-exploitation of vulnerabilities for example with publicly available scripts. In practice, this important defense is often neglected which makes it possible for available exploits to have lifetimes up to several years [21, 77, 93, 177].

Attackers should not be able to get unauthorized access to hosts, for example, by exploiting weak passwords. A minimum requirement is to use passwords which are difficult to guess with or without existing password cracking tools.

The antivirus software should be using the most recent virus definition database. This helps detecting known worms and viruses. Antivirus software can thus be considered as an IDS.

All network users should use common sense in handling attached files or weblinks in e-mails. Social engineering is widely used in the Internet to lure people into installing malicious software voluntarily on their hosts [206].

Users should always read the end user license agreement (EULA) before installing software on their hosts. EULA is a legal contract between the user and the software

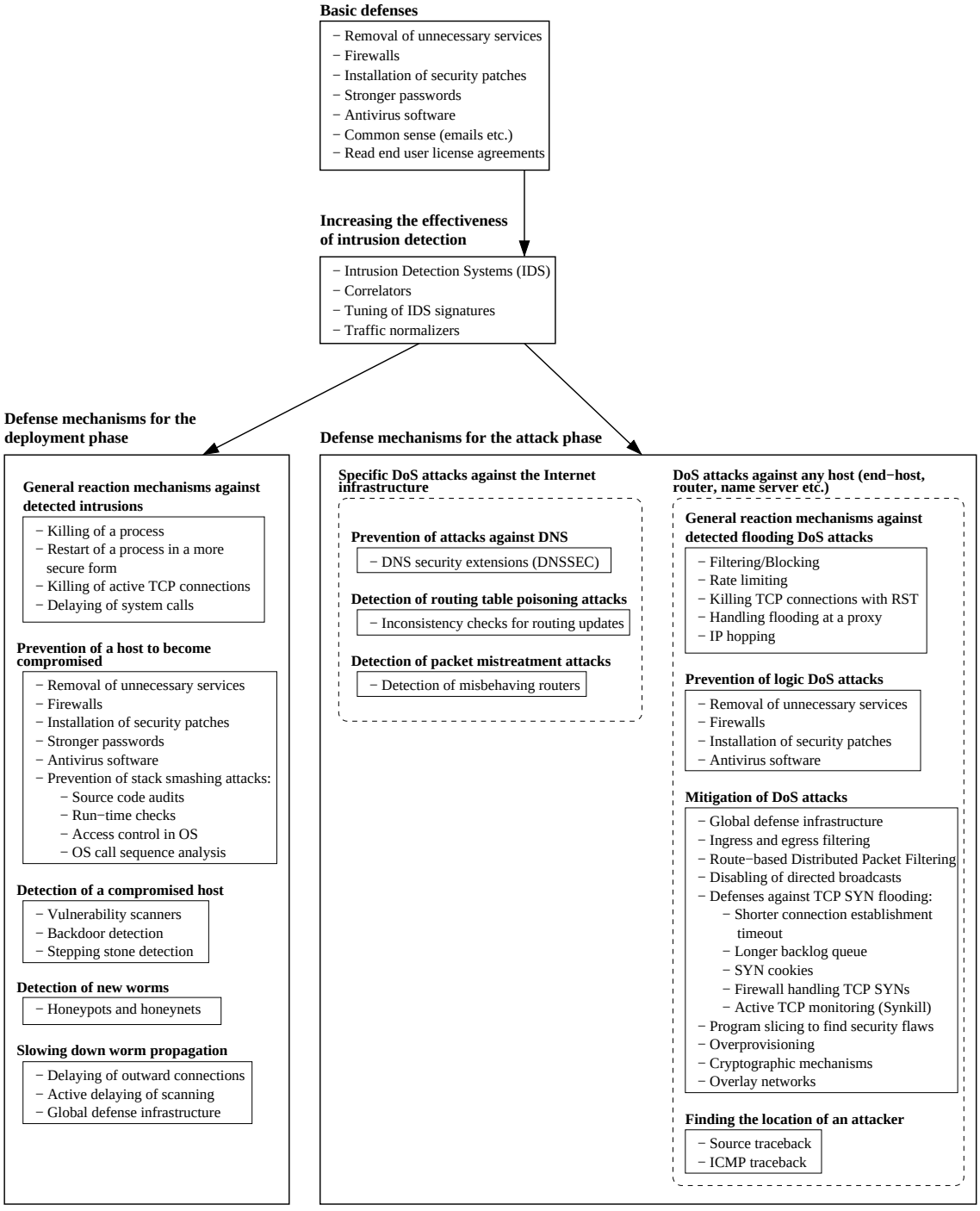

Figure 2.3: Different defense mechanisms available to mitigate DoS attacks.

publisher. A user typically does not read the EULA because it is often shown in a very small window, contains a large amount of text, and is difficult to understand. EULA can, however, allow third parties to use computing resources in a Trojan horse fashion [62].

2.1.4.2 Increasing the Effectiveness of Intrusion Detection

Mitigation of DoS attacks is not possible if these attacks are not detected. A combination of NIDSs and HIDSs are typically needed in the most important networks and hosts, respectively.

To make intrusion detection more effective, *correlators* can be used to prioritize alerts, group alerts related to different phases of an attack, and group alerts from several different IDS sensors or analyzers [87]. The objective is to combine the benefits of different kind of IDSs and reduce the amount of human interaction.

An IDS needs to be customized to a given environment to reduce the number of false positives [153]. Instead of technical effectiveness one should increase the cost/benefit trade-off, which means focusing the limited computer and human resources on the most damaging intrusions [126]. This is done by tuning the signatures or training the anomaly detection to detect those attacks, for which the environment is most vulnerable, and to prevent alerting on those attacks, for which environment is not vulnerable. An existing signature should be modified if it creates too many unnecessary alerts.

Attacks against an IDS should be made as difficult as possible. An NIDS should have defenses against insertion and evasion attacks (such as those described in [173]). If an NIDS itself cannot provide a decent protection against these attacks, a traffic normalizer [88] is one possible tool for this purpose. A normalizer eliminates some ambiguities in the traffic stream, for example by reassembling IP fragments.

Anomaly based IDSs are required for detecting new and many existing DoS attacks. Anomalies can be based on different features, such as the number of connections made to a host during a specific time interval and the number of connections made to a service during a specific time interval [155].

One should be careful in using existing evaluations as the basis for choosing an IDS. Evaluation of IDSs is not easy, as these results depend on many issues, such as the type of background traffic, amount of background traffic, distribution of attack traffic in the background traffic, correct training of anomaly detection, correct configuration of systems collecting audit data, types of attacks tested, and number of attacks tested [139]. Despite of these problems, it is necessary to have evaluations, as they will point out at least some characteristics of existing IDSs.

2.1.4.3 Defense Mechanisms for the Deployment Phase

The goal of the deployment phase is to compromise a host by installing malicious software (a DoS attack tool) on it. A host can be any node connected to the Internet, like an end-host or a router.

Some defense mechanisms described in this section only detect the deployment phase, but many defense mechanisms also include an automatic reaction mechanism. Basically it is possible to combine any detection mechanism with one or more reaction mechanisms. The following general reaction mechanisms have been proposed in the literature:

- killing of a process [56],
- restarting an application in a more secure form (such as with more effective and more expensive checks against buffer overflow attacks) [56],

- killing of active network connections, for example, by using a TCP packet with the reset (RST) bit set in the TCP header (a TCP RST packet) [164], and
- delaying of network connections to new destinations [215].

Worm propagation and DoS tool deployment are typically based on so called *stack smashing attacks* [55] where an attacker can get unauthorized access (like a root shell) to a victim host. These attacks are based on overwriting the contents of a stack [8,36]. Buffer overflow is the most usual software vulnerability used by these attacks. During the time period 1–11/2002 there were 31 CERT Advisories, from which 21 referred to buffer overflow attacks [24]. Another possibility to carry out a stack smashing attack is to exploit a format string vulnerability [29]. Yet another possibility is to exploit an integer error vulnerability [5].

Stack smashing attacks can be prevented or made more difficult by paying attention to *software security* which can be enhanced by three different mechanisms [53]:

- *Software auditing* can be used to search vulnerabilities from source code automatically or manually before these vulnerabilities are found and exploited by attackers [209].
- *Vulnerability mitigation* is based on compile-time installation of special checks that detect certain types of buffer overflow attacks and protect the system at run-time [54, 56].
- *Behavior management* is based on run-time features in an operating system to limit potential damage or block specific behavior known to be dangerous (access control).

Stack smashing attacks can also be prevented or made more difficult by looking at short sequences of operating system calls [70]. For example, the exploit of a buffer overflow vulnerability in a `sendmail`-program causes it to issue abnormal sequences of system calls, when a `sendmail`-process starts to execute a root shell.

If detection of stack smashing attacks is based on anomaly detection, an attacker may use a service in such an exceptional way that causes the reaction mechanism to halt the process after the detection of a false positive. In this case it should be noted that the misuse of a defense mechanism can result in another kind of a DoS condition, when an intact process is halted. The length of this kind of a DoS condition may vary depending on whether a process is automatically restarted or if human intervention is required. A service may also be restarted automatically in an enhanced mode which includes additional checks against stack smashing attacks. This can make a service run slower. Naturally it is more important to prevent a host to become compromised.

Compromised hosts can be detected with vulnerability scanners and with an intelligent analysis of network traffic. Malicious software often installs secret backdoors into an infected system. Backdoors are a basic method to provide a hidden access for an attacker into a compromised host. To make it more difficult to locate an attacker, a chain of compromised hosts (or stepping stones) can be used to hide the real source of interactive attack commands. It is possible to uniquely recognize flows containing interactive traffic because every interactive flow has distinctive packet size and timing characteristics. Encryption does not affect the detection because the actual content of the flows is not used as the discriminating feature. A backdoor can be detected by recognizing interactive traffic entering an unusual port [221]. A stepping stone can be detected by recognizing two interactive flows with identical characteristics, one entering a host and the other leaving the same host [222]. Even longer chains of compromised hosts can be detected in a similar fashion [197].

Worm propagation can be restricted by limiting (delaying) the rate of connections to new destinations [215]. An infected host will try to connect to as many different hosts as fast as possible, but an uninfected host typically makes connections mainly to locally correlated destinations and at a lower rate. The slower a worm propagates, the easier it is to prevent the worm from infecting most of the population because there is more time to react, for example by installing filtering rules in routers.

If a (global) defense infrastructure is available, the number of infected hosts can be reasonably restricted by installing filtering rules in the most important routers of the Internet. In [150] it was studied how the reaction time affects the size of the infected population in case of the Code Red I v2 worm. To keep the ratio of susceptibles infected within 24 hours below 10%, simple address blacklisting filters (filtering based on IP addresses of the infected hosts) should be installed within 20 minutes, but more generic content filtering (filtering based on the signature of the worm) allows almost three hours for installation. In [150] it was also estimated that in case of the Code Red I v2 worm the infected population could have been restricted to less than 20% of the population really infected. This would have required the installation of worm signatures within 2 hours in the 30 most important Autonomous Systems of the Internet.

An example of a more active defense mechanism for slowing down the worm propagation rate is the LaBrea Tarpit ([157], pp. 173–178). When an attacker scans an unused IP address, a router in the destination network will send an Address Resolution Protocol (ARP) request for the unused destination IP address. If the response for this ARP request is not found within a while, a specific server (the LaBrea host) will respond to it with its own hardware address. The LaBrea host will thus receive traffic sent to an unused destination IP address. The scanning application (or one thread of the attack application) may be delayed for quite a long time (possibly indefinitely) by replying appropriately to incoming TCP packets.

Honeypots and honeynets are hosts or networks, respectively, intended to be attacked or compromised [118, 170]. These systems are not used by any legitimate users, so any connection to them is expected to be malicious. These systems can be useful in detecting new worms and DoS attacks, for which there does not yet exist any exact signature. To succeed in identifying the attack characteristics, a honeypot should be difficult to differentiate from an ordinary end-host. Otherwise an attacker or a worm may be able to avoid using a false end-host [92, 119].

2.1.4.4 Defense Mechanisms for the Attack Phase

This section lists defense mechanisms presented in the literature for the attack phase. Many of these defense mechanisms are suitable against both flooding and logic DoS attacks, but not all. For instance, some defense mechanisms cannot prevent logic DoS attacks because even a single malicious IP packet is able to cause a DoS condition, such as a system crash.

The following general reaction mechanisms have been proposed in the literature for mitigating flooding DoS attacks [48]:

- Blocking: all packets matching a signature are discarded at an upstream router.
- Rate limiting: a fraction of packets matching a signature is discarded at an upstream router. A support for quality of service (QoS) features should be provided by the involved routers. Only incoming packets are rate limited, and outgoing packets can leave the network freely without any additional penalties. Rate limiting cannot discard too many packets because legitimate flows matching an attack signature must survive the one-way packet loss. This limits

the effectiveness of rate limiting in mitigating high-bandwidth flooding DoS attacks (see the publication P6).

- Connection tear-down: malicious TCP connections are torn down with a TCP RST packet.
- Flood processing in another place: a DoS flood can be handled in a place with better abilities. For example, a router can take the responsibility of handling (proxying) certain resource intensive tasks. This saves resources in a victim end-host.
- IP hopping: the IP address of a victim is changed in the DNS. In the Code Red I v2 worm the IP address of the victim was hard-coded, which made it easy to prevent the attack by changing the victim's IP address in the DNS.

Both end-hosts and network security devices should be able to control these reaction mechanisms. In [122] it was argued that end-hosts are in the best position to detect and react to attacks, but network security devices are also required for this, such as to detect distributed attacks against many different target hosts.

Blocking and rate limiting of DoS traffic at upstream routers requires a mechanism for distributing the attack description. The proposed mechanisms for an IDS to distribute attack identification information are the Pushback-messages [68] and the Intrusion Detection and Isolation Protocol [199]. These protocols must authenticate every message because otherwise an attacker can exploit this mechanism by sending spoofed messages, which cause routers to block or rate limit legitimate traffic and cause DoS.

The term *Internet firewall* [48] has been used to denote a global defense infrastructure where many routers in the Internet infrastructure detect and filter attack traffic in a coordinated way. At the moment it is mostly the responsibility of the owner of an end-host to protect it from malicious attacks, for example, by using conventional firewalls and antivirus software. The Internet infrastructure is more optimized for efficient transmission of IP packets than for implementing defense mechanisms against malicious attacks. An Internet firewall means giving the Internet infrastructure a more active role in mitigating the effect of malicious network behavior on end-users. A wide-scale Internet firewall will, however, have problems handling false positives. It will be difficult to restore the state of a legitimate stream if it has been accidentally classified as attack traffic. A defense infrastructure consists of systems from several different organizations, and this may slow down the cooperation during error cases. An Internet firewall, however, has been found to be a necessary defense mechanism against fast-spreading worms exploiting a newly found vulnerability [192].

IP spoofing can be restricted by using ingress [67] and egress filtering in a border router of a network, such as a network of an ISP or its customer [117]. An ingress filtering router in an ISP network will check that packets coming from a customer network have valid source IP addresses with the associated prefix of that customer. Egress filtering does the same check for packets going in the opposite direction, which prevents an ISP border router from forwarding packets that have a source IP address belonging to the same customer as the destination IP address. It should be noted, however, that ingress and egress filtering do not prevent DoS attacks because a well-chosen chain of stepping stones makes it unnecessary to use IP spoofing for hiding the attacker. Also, ingress and egress filtering do not prevent sending packets with addresses of non-existent hosts (and with the correct address prefix) because only the address prefix is checked. It is not feasible for an ISP to keep track of all IP addresses really used by the customers. Egress filtering is, however, suitable for implementing prioritizing of local accesses to a server under a DDoS attack. Here,

egress filtering is used to implement blocking which prevents all attack packets (and legitimate connections) from entering the target network from the rest of the Internet. As a result of this kind of prioritizing, clients in the same network as the victim server experience normal availability of this server, but clients in other networks are blocked.

In route-based Distributed Packet Filtering (DPF) a router will only accept a packet received through the same interface as the one through which packets would be sent back to the original sender [161]. A packet is discarded if these two interfaces do not match. DPF is not able to consider asymmetric routing or recent route changes [48].

Flooding DoS attacks based on broadcast amplification can be prevented by disabling directed broadcasts in routers [189] because this feature is normally not utilized, except in some implementations of Mobile IP.

TCP SYN flooding is a widely used flooding DoS attack mechanism. Most available DoS tools support this attack type and studies also indicate that most DoS attacks are TCP-based [151]. The effect of TCP SYN flooding attacks can be mitigated by applying the following defenses [187]:

- improve end-system configurations (reduction of the timeout period for half-open connections, increase in the backlog queue size),
- improve connection establishment to prevent storing half-open connections (storing the connection status in the initial sequence number as a SYN cookie),
- move the burden of handling half-open connections to a firewall (such as described in [49]), and
- monitor actively existing TCP connections (Synkill, sources classified as evil are prevented from making additional connections).

Security flaws in protocol software can be detected by using a technique called program slicing. Those parts of the full source code that have an effect on a value of a certain variable are extracted and carefully studied. For example, spurious (extraneous) state transitions in implementations of the TCP state machine can be identified with this technique [85].

An attacker does not need to attack a victim directly. The attacker can instead target the Internet routing infrastructure to make a DoS attack [20, 107, 160]. Attacks against the Internet infrastructure can be divided in four classes [46]:

- *Attacks against the DNS* can be mitigated by using DNS Security Extensions (DNSSEC) which provide end-to-end authenticity and integrity [107].
- *Routing table poisoning attacks* can be mitigated by using inconsistency checks to detect malicious updates [100]. For example, advertisements in the Border Gateway Protocol (BGP) can be compared to the information in the Internet Route Registry (IRR). Another proposal is to use DNS to verify the contents of routing advertisements. Digital signatures may also be required.
- *Packet mistreatment attacks* include misrouting of packets (for example, to heavily loaded links), dropping of packets, and delaying of packets. Packet mistreatment attacks can be detected by running a specific detection protocol inside an Autonomous System to locate and isolate routers dropping or misrouting valid packets [31]. Another possibility is to use the secure traceroute to locate the source of a routing misbehavior [159]. Secure traceroute is based on including the next-hop address in traceroute replies, and using such test packets, which can be recognized only by the intended recipient router.

- *Ordinary DoS attacks* are flooding or logic DoS attacks already described in this section. The same defense mechanisms can be used, regardless of the victim being an ordinary host or a node in the infrastructure. One simple mechanism to mitigate effects of flooding DoS attacks against DNS name servers is to increase the Time To Live (TTL) value of host IP addresses (the publication P2).

Source traceback is an important task in locating the source of attack traffic. The proposed traceback mechanisms are usually based either on recording information in routers about forwarded packets for later traceback requests [195] or on sending additional information about the route of the packet to the victim (like ICMP traceback) [23, 182]. It is, however, difficult to locate true origins of attack packets because reflector attacks are based on using innocent third-parties as sources, and network address translators can make it impossible to further trace the route of the attack packets. Also, the use of a chain of stepping stones can make it impossible to locate the real attacker.

Overprovisioning of resources like access bandwidth and processing power can increase the resistance against flooding DoS attacks [37]. Despite of the related costs this defense can be an effective way to protect important targets like DNS name servers.

Cryptographic mechanisms, like IP security extensions (IPsec), can be used to authenticate message sources and encrypt messages. Authentication mechanisms (like end-to-end authentication) are suitable for preventing many logic DoS attacks, but may have difficulties in mitigating the effect of flooding DDoS attacks because the number of compromised DDoS agents can be huge. Even if the work load on a client would be higher than that of a server (such as in the case of client puzzles [16, 112]), the large number of compromised hosts can still generate more traffic than a victim network or server can handle. It should also be noted that cryptographic mechanisms are often susceptible to resource exhaustion attacks, and have difficulties in mitigating insider attacks where an attacker succeeds in getting the access rights and privileges of a legitimate user, for example, by compromising a host.

The use of overlay networks has been proposed as a defense mechanism against DoS attacks. One example is the secure overlay structure (SOS) [116] which forces legitimate traffic from well-known sources to be validated before forwarding to a protected server is allowed through a prespecified secret route. Another example is the Secure-i3 (Secure Internet Indirection Infrastructure) [3], which decouples the act of sending a packet from the act of receiving it. Packets are sent with a logical destination identifier, and receivers express their interest in receiving packets by inserting a trigger for a specific logical identifier in the Secure-i3 overlay infrastructure. This kind of an overlay network is able to mitigate many flooding DoS attacks. In [3] it is argued that the complexity of the Secure-i3 overlay infrastructure does not introduce any new security vulnerabilities, but it is not reasonable to expect, for example, that an attacker would not be able to compromise any parts of an overlay system. Also, an overlay network cannot remove all vulnerabilities of the underlying IP infrastructure.

2.1.5 Selection of Defenses

Mitigation of DoS attacks requires a comprehensive set of defenses. Implementing and applying every possible defense is, however, not feasible. This would simply cost too much in terms of resources, like humans, equipment, money, and time [126]. It is not even possible to achieve perfect network security: new vulnerabilities in computer systems are continuously found, security of one site is dependent on the

security of other sites in the Internet, and some attacks are difficult or even impossible to distinguish from legitimate network traffic.

Risk management is the deliberate process of understanding the most important risks and deciding how to mitigate them [40, 91]. The goal of risk management is to decide whether to accept a risk, mitigate it to an acceptable level, or transfer it to someone else (or any combination of these) [157]. Some risks can be accepted as such if they are not very probable or if the impact is not too critical. Risks related to the most important assets of an organization must be reduced (mitigated) either partially or completely by applying reasonable defenses. Insurance can sometimes be used to transfer a (remainder) risk to another party. Even though information security risks are difficult to measure and predict ([183], p. 302), there are insurances available in the marketplace [83]. These insurances cover both first-party (damages to oneself) and third-party (damages to someone else) risks arising from Internet activities of an organization. In any case, the total cost of handling a risk must be commensurate with the value of assets being protected.

Combating DoS attacks is primarily an exercise in risk management which must consider both technical and business aspects [94]. Major risks must be avoided, but at the same time the consumption of finite resources on security must be optimized. The costs related to security expenditures must be traded against the acquired benefits. An organization which is able to avoid major security incidents and at the same time minimize security costs is better able to stay in business and make profit.

An organization-wide *security policy* is an important prerequisite for risk management. A security policy defines the main principles (goals) for protecting the most important assets [9]. It also explains why these assets should be protected from certain threats. With the help of a security policy it is possible to concentrate consistently on the major threats and implement a cost-effective set of defenses [126].

Legislation, standards, best current practices, and other documents may dictate parts of security policy and risk management. Legislation may specify requirements for availability of public services and protection of confidential information. There are also recommended security services and procedures for ISPs who are encouraged to become proactive in security issues. The set of recommendations for ISPs in [117] includes, for example, a reasonable resistance to known security vulnerabilities in the network infrastructure. These recommendations are incentives for ISPs to defend themselves against DoS attacks.

The selected set of defenses should be seen to be dynamic. Threats, risks, and attack mechanisms change as a function of time. The set of implemented defenses should be modified to match the current requirements. For example, at the moment the most prevalent attacks exploit code-level flaws such as buffer overflow vulnerabilities. The future malicious software will probably exploit in an increasing fashion vulnerabilities in system's design, architecture, and usability [216].

2.1.5.1 Factors Affecting the Selection Process

Choosing a cost-effective set of defenses is not a simple process because it is difficult to compare different defense mechanisms. There are several factors that need to be considered when assessing defense mechanisms:

- Effectiveness: How capable is a defense mechanism in mitigating DoS attacks?
- Reliability: Does a defense mechanism always mitigate DoS attacks as well, or is it sometimes less effective? Is there a possibility for false positives?
- Misusability: Can an attacker exploit a defense mechanism in an unexpected way as a tool for achieving a DoS condition?

- Collateral damage: Does a defense mechanism cause any negative side effects, like performance problems in routers or a requirement for extensive human intervention to solve false positives?
- Proactivity: Can a defense mechanism prevent attacks or does it only react to existing attacks?
- Completeness: What kind of other defense mechanisms are required? For example, a plain detection mechanism must be combined with a reaction mechanism.
- Reaction delay: How fast does a defense mechanism react to intrusions (for example, has a host already been compromised and trojaned/backdoored when an intrusion has been detected)?
- Ease of implementation: Is it feasible or possible to implement a defense mechanism (for example, the number of different organizations involved, access to source code, implementation cost worth the benefit)?
- Ease of use: Is the human interface easy to use? Does a defense mechanism fit with an already existing security infrastructure?
- Installation place: What is the optimal place to implement a defense mechanism (such as an ISP or a customer network)?

Answering these questions will help to understand the real benefit of using a certain defense mechanism to mitigate DoS attacks.

2.1.6 Summary

The section 2.1 has described what Denial of Service attacks are, how they can be carried out in IP networks, and how one can defend against them. It is not possible to completely prevent these attacks because there will always be vulnerable hosts in the Internet to be compromised for attack purposes, and many DoS attack mechanisms are based on using ordinary features of protocols or network services. Also, vulnerabilities in applications can be easily exploited by malicious DoS software. In practice this means that mitigation of DoS attacks requires a comprehensive set of defense mechanisms to get defense in depth.

Many different defense mechanisms are typically needed to mitigate DoS attacks. At minimum, any organization or individual should remove all unnecessary services, use a firewall, install relevant security patches timely, avoid using weak passwords, use antivirus software with the most recent virus definition database, and use common sense in executing e-mail attachments and navigating in the Web.

It is not cost-effective to blindly choose a large set of defense mechanisms against DoS attacks. Organizations differ in the way they do business, and this has an effect on what kind of defense mechanisms are needed. For example, a university and a web book store have very different requirements for DoS attack mitigation. Even two similar organizations will probably choose at least partially different defense mechanisms if one organization is willing to accept a higher risk and push down the associated short-term costs. Also, the size and reputation of an organization can make a difference in the defense strategy. Attacks against well-known organizations have a higher probability of getting publicity. Even though massive, large-scale DoS attacks are found rather seldom, the risk is evident. Small-scale DoS attacks, on the other hand, are part of every-day life in the Internet. The risk of a DoS attack should not be underestimated, but it should not be overestimated either.

2.2 Mitigating Flooding Attacks against the DNS

The Domain Name System (DNS) represents an effective target for Denial of Service (DoS) attacks [46]. In a flooding DoS attack a continuous flow of valid-looking DNS requests overloads a network link, a router, a firewall, or a name server. As a result, legitimate DNS requests have problems in reaching a name server and getting an answer. By disabling part of the DNS an attacker is able to prevent or delay access to many services in the Internet. Users typically have only the textual name of a server they are trying to connect to. If the DNS is not available for mapping a textual server name to a numerical IP address, the corresponding server cannot be contacted regardless of the availability of this server. A numerical IP address is always required before it is possible to create a connection to a server.

A new defense mechanism for protecting name servers against flooding DoS attacks is described and analyzed here. This new defense mechanism is based on increasing the Time To Live (TTL) value of IP addresses (DNS A records) during a flooding DoS attack. The main goal is to increase the cache hit rate at local name servers which reduces the number of legitimate DNS requests at an overloaded name server. Simulation results show that the mechanism is able to reduce both the average delay associated with the DNS lookup and the number of completely failed DNS lookups.

Protecting DNS is an important subject due to the necessary role of DNS in accessing services, due to the prevalence of flooding DoS attacks against name servers [34, 43, 151], and due to the lack of effective defense mechanisms against these attacks. At the moment there are no effective defense mechanisms that an organization could use to mitigate flooding DoS attacks against its name servers. Name servers can prevent DNS queries from specific source addresses, but for public services this kind of prevention is not possible. Ingress and egress filtering have been suggested for mitigating flooding DoS attacks using spoofed source IP addresses [43], but these defense mechanisms require extensive deployment in the Internet.

The scope here is limited to those name servers providing a final DNS A record (IP address) for a DNS lookup. These name servers are typically the responsibility of the owner of the corresponding *zone* ([7], p. 21). A zone is a non-overlapping part of the DNS. In February, 2003, approximately 68% of the zones in the *com.*-domain were found to be misconfigured [141]. Thus, the level of expertise in operating these name servers is not always high, and a flooding DoS attack against them can easily be successful. A successful flooding attack against name servers of a single zone prevents all accesses (without a numerical IP address or without an already cached address record) to any service or host in this zone, anywhere from the Internet. Root and Top Level Domain (TLD) name servers, on the other hand, have proved to be very resistant against flooding DoS attacks [207] due to required overprovisioning [37], so they are not considered here.

2.2.1 The Dynamic TTL Mechanism

This section describes the dynamic TTL mechanism which mitigates flooding DoS attacks against name servers. As a reaction mechanism [94] it is used after a DoS attack is detected manually or automatically (for example, by inspecting log files or by measuring DNS performance from a remote site). The detection mechanism, however, is not included in the dynamic TTL mechanism.

The dynamic TTL mechanism is based on using two different TTL values for each A record: a lower value for normal operation (*default TTL*) and a higher value during a detected DoS attack (*TTL during attack*). An optimal value for the TTL during

attack is the length of the attack, but as it is not possible to exactly estimate this length, one should make a reasonable guess. Different problems arise if the guessing algorithm chooses a too short or a too long TTL value during an attack. When the TTL during attack is too short, clients must again contact an overloaded name server and try to get a response. If a response is not received, access to the attacked zone is prevented. When the TTL during attack is too long, possible DNS-based load balancing is not possible after the attack is over until the name information with a long TTL value expires. In practice, different values can be used for the TTL during attack, as long as it has a higher value than the default TTL.

A longer TTL value makes it possible to have higher cache hit rates at remote name servers. When the IP address of a destination host is found from the cache, the overloaded name servers need not be contacted. If the attack is targeted only at the name servers, the final destination can be contacted without problems.

The dynamic TTL mechanism is supposed to be used primarily for A records containing IP addresses of end hosts. A Name Server (NS) record contains an IP address of another name server having more detailed information about a specific subdomain. NS records are fairly static, and they have a much longer average TTL value (up to several days) than A records.

A major benefit of the dynamic TTL mechanism is that it is easy to implement and does not depend on any third parties. Only local operations are required to mitigate an attack and increase the availability of local services to the public Internet.

2.2.2 The DNS Simulator

The effect of dynamic TTL values in mitigating DoS attacks against name servers was simulated with an OTcl program under the ns-2 network simulator. The setup of client groups, World Wide Web (WWW) servers, and name servers is shown in the figure 2.4 where 200 independent client groups initiate DNS lookups to 1000 different WWW servers each in a separate subdomain. All client group subdomains are completely separate from server subdomains. The WWW servers are ordered according to their Zipf-like popularity, the WWW server in subdomain 1 being the most popular. A flooding DoS attack is targeted against the name servers of subdomain 1 which is using the dynamic TTL mechanism. The simulator implements all the basic DNS functions.

The numbered arrows in the figure 2.4 indicate the order of messages required for initiating a connection to a web server when DNS caches are initially empty. In this case one needs to contact a local name server, a root name server, a generic Top Level Domain (gTLD) name server and finally an authoritative name server for the final subdomain. For more details, see the publication P2.

2.2.2.1 Clients, WWW Servers and Name Servers

In the simulator there are 200 independent groups of clients. Each client group would reflect, for example, the users of an organization or the customers of a small Internet Service Provider (ISP). Each client group is expected to have one local caching name server. The exact nature of the arrival process of DNS requests at a local name server is not known. Here it is expected that each client group is independently initiating DNS lookups to the local name server with an exponentially distributed inter-arrival time. The inter-arrival time means the average time between the arrival of two consecutive DNS requests from a specific client group. Each client group selects randomly the name to be looked up. The selection probabilities of domain names have a Zipf-like distribution. Two different average values for the exponentially distributed

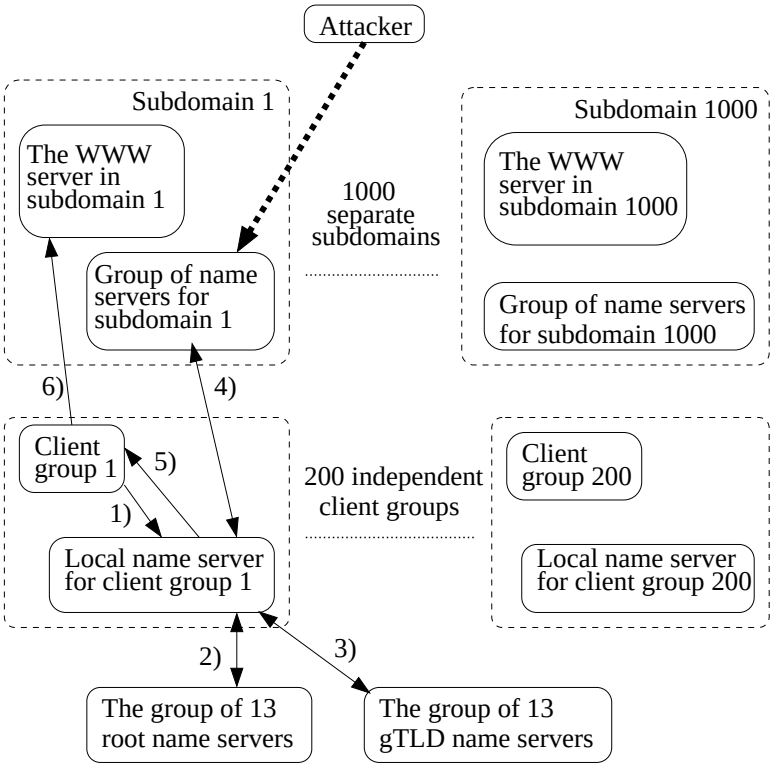

Figure 2.4: The setup of the DNS simulator.

inter-arrival times were used in the simulations: 120 and 7200 seconds. This makes it possible to study the approximate effect of inter-arrival time on the usefulness of the dynamic TTL mechanism.

The client groups are trying to resolve the name of a server, which is here expected to provide WWW services in a subdomain under the *com.*- or *net.*-domains. The number of WWW servers is 1000, each located in a different subdomain. There are thus 1000 different server subdomains with their own name servers. WWW servers and their corresponding subdomains are identified by their number from 1 to 1000. All client group subdomains are completely separate from server subdomains. Each client group chooses the destination WWW server using a Zipf-like distribution, with the parameter $\alpha = 0.8$. In one study the distribution of web requests was found to be Zipf-like with the value of α being approximately 0.8 [32]. This kind of a distribution means that the WWW server in subdomain 1 is the most popular site, and the WWW server in subdomain 1000 is the least popular site [2]. Vast majority of DNS lookups from any client group are for the WWW server in the subdomain 1.

Each client contains a stub resolver (see [7], p. 26), which is configured to always contact a single local name server. There is one local recursive name server for every client group. Every local name server saves any received resource record in its cache for the duration of the TTL.

Each WWW server subdomain is expected to have a set of four name servers providing authoritative resource records for the subdomain. The number of these name servers has an effect on the retransmission schedule. Namely, a local name server retransmits in a round-robin fashion to every member of a name server group with an initial timeout length of two seconds ([121], and [7], p. 109).

2.2.2.2 TTL Values

All information stored in the DNS, such as an IP address, is associated with a TTL value. This value indicates the time period for which the corresponding information can be stored in a cache. After an IP address has expired in a cache, this address must be refetched from an authoritative name server if the address is needed again.

All NS records in the DNS simulator are associated with a TTL value of 172800 seconds (2 days). This was a typical TTL value for NS records in the Internet in January, 2004.

All A records (IP addresses of WWW servers) in the DNS simulator use a default TTL of 600 seconds (10 minutes). In one study it was found that the median TTL of A records was approximately 15 minutes when measured from a TTL distribution weighted by access counts [113]. Due to the tendency to use shorter TTL values for A records, the DNS simulator uses 600 seconds as the TTL value for A records.

The dynamic TTL mechanism uses another higher TTL value for A records during DoS attacks. A TTL value of 7200 seconds (2 hours) was chosen for this purpose. This temporary TTL should be in the order of the expected attack length.

2.2.2.3 Delay Distribution for Request-Response Times

The delay between the transmission of a DNS request and the reception of the corresponding response (request-response time) is expected to be normally distributed with a mean value of 92 milliseconds and a standard deviation of 15 milliseconds. In one study [35] it was found that generally no single distribution appears to give a consistently good fit to measured delays of real DNS traffic. A normal distribution with the above mentioned parameter values was found to match reasonably well the request-response times of the L root name server during one measurement period. In the simulations all transmission delays are included in this request-response time distribution.

The simulator does not take into account the transmission delays between the resolver and the local name server. Delays of remote DNS requests in the simulations have the average value 92 ms. Therefore we may ignore some local delay components. It is expected here that a client group is close to its name servers (at most in the order of some tens of kilometers). Therefore local propagation delays are fractions of a millisecond and can be ignored. Transmission delays are also ignorable (< 0.2 ms) because the size for DNS reply packet is at most in the order of 200 bytes and the transmission speeds are assumed to be in the order of 1–100 Mbps. In the simulations the DNS lookup delay is assigned to 0 ms if the target name is already cached at the local name server, and no DNS request needs to be sent to a remote name server of a target zone.

2.2.2.4 Flooding DoS Attack

The victims of the DoS attack are the four name servers of the most popular WWW server subdomain having the number 1 in the Zipf-like distribution. The attacker is expected to flood all these name servers with excess traffic, like DNS, ICMP, UDP, or TCP SYN messages. All four name servers of the victim subdomain 1 are attacked in the same way. All remaining name servers (name servers for subdomains 2–1000, root name servers, gTLD name servers) experience no packet loss and respond always to every DNS request.

The DoS *attack intensity* is defined to be the percentage of lost incoming DNS requests due to the excessive load. For example, only 10% of the DNS requests will be answered if the attack intensity is 90%.

It is not possible to accurately model the load from DNS requests at name servers. In real-life the amount of pure attack traffic at the victim site depends, for example, on generic network load, number of attack sources, access line speeds of attack sources, the processing power of attack sources, properties or the used DDoS software, and the properties of the system coordinating DDoS agents affect. DNS requests from legitimate sources also affect the load of name servers. As an approximation a constant load for name servers (the attack intensity) was used in the simulations. The effect of different DNS request retransmission schemes on DNS traffic load is left as a possible future work item.

Random packet loss typically found in networks is not included in the simulator.

2.2.2.5 Retransmission of Lost DNS Requests

During a flooding DoS attack, only the name servers for subdomain 1 will experience packet loss. The simulator software includes support for the retransmission of lost DNS requests both at resolvers and local name servers.

The DNS resolver in a client is expected to have a retransmission mechanism similar to the Berkeley Internet Name Domain (BIND) resolver, version 8.2.1 or later ([7], p. 110). The resolver will retransmit only once after a timeout of 5 seconds. If the retransmission is not answered within 10 seconds, the resolver will return an error to the calling software. A resolver will thus spend a maximum of 15 seconds for a DNS lookup.

Local name servers in the simulator have a similar retransmission mechanism as in BIND version 9.2.3 with the following exceptions: Round-Trip Time (RTT) is not calculated and the set of redundant name servers are cycled through only twice. The timeout length is always 2 seconds during these two cycles. A local name server will cease retransmitting after 7 trials.

2.2.3 Results of the Simulations

The goal of the simulations is to see how the DNS performance depends on the TTL value of DNS A records during a flooding DoS attack. The simulator provides information about the delay of successful DNS lookups and the proportion of completely failed DNS lookups.

The relevant parameter values in the simulations are the following:

- the length of one simulation is 1 000 000 seconds,
- the DoS attacks starts at the time of 300 000 seconds,
- the attack is carried out at the intensity of 90% (some tests also with the intensity of 50%), and
- the average time (from exponential distribution) between consecutive DNS lookups from a single client group is either 120 or 7200 seconds (*inter-arrival time*).

The rationale for some of these parameter values is the following. The simulation length was chosen so that it is several times longer than the value of the TTL of NS records, that it is long enough to provide reasonably smooth result curves, and that it is short enough to provide a reasonable running time on the available workstation. The average inter-arrival time was chosen more or less randomly, but two different values were used to study the effect of this parameter on the results.

All simulation results are calculated from the DNS traffic from any client group (1–200) to subdomain 1 because only subdomain 1 is the target for a DoS attack.

In the simulations it is expected that the DoS attack is detected at the same time when the attack begins (detection is not the subject here). In practice, however, there is always some delay associated with the attack detection. The dynamic TTL mechanism cannot increase the DNS performance until a DoS attack against the name servers is detected and the new TTL values have reached the local name servers.

If the detection delay was considered in the simulations, it is expected that benefits from the dynamic TTL mechanism would begin at the moment when the longer TTL is taken into use. After an attack has started and before the new longer TTL is in use, the DNS performance is expected to be similar to the case where no dynamic TTL mechanism is used.

2.2.3.1 The Delay of Successful DNS Lookups

The average DNS lookup delay is shown in the figure 2.5. The X-axis indicates the time in seconds when a DNS lookup was initiated. The Y-axis indicates the delay of DNS lookups averaged over 10 000 second intervals. This figure shows the average delay when the dynamic TTL mechanism is used to protect the subdomain 1 (*TTL = 600/7200)* and when this mechanism is not used (*TTL = 600*). The results are shown for inter-arrival values of 7200 seconds (thick lines) and 120 seconds (thin lines).

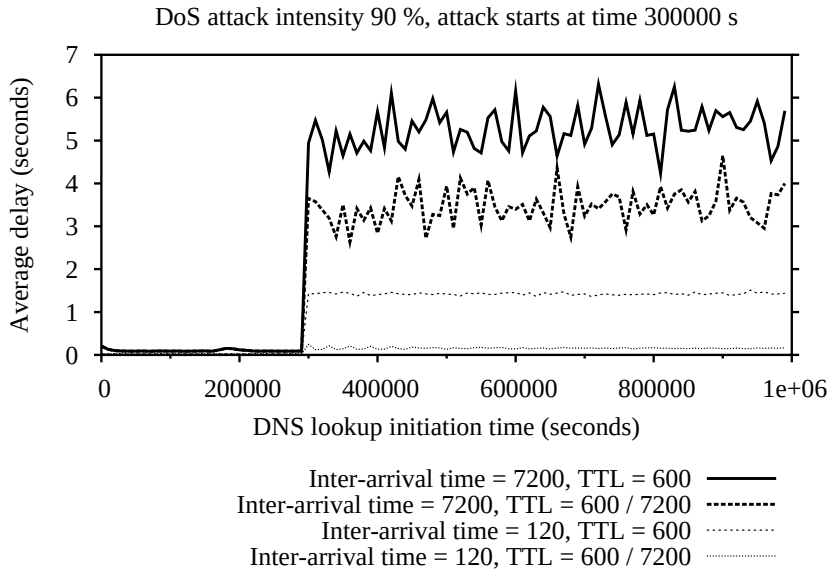

Figure 2.5: The delay of successful DNS lookups averaged over 10 000 second intervals.

If the inter-arrival time is 120 seconds, the average delay is reduced almost 90% from 1.5 seconds to approximately 0.16 seconds. When the inter-arrival time is 7200 seconds, the average DNS lookup delay is much longer than when inter-arrival time is 120 seconds. The reason for this is very logical because the more DNS lookups there are in a time unit, the more lookups will result in a cache hit at the local name server. This pulls down the average lookup delay.

The positive effect of the dynamic TTL mechanism is visible with both inter-arrival times. This effect is, however, stronger when inter-arrival time is shorter. The shorter the inter-arrival time is, the more cache hits will result at the local name server.

If a subdomain is visited very seldom, the cached A record will time out before the next visit to it.

As expected, the dynamic TTL mechanism provides the best benefit for those client groups which have many clients referencing a similar set of popular destinations. This increases the possibility for a cache hit at the local name server.

Averaging DNS lookup delays over 10 000 second intervals hides some details of shorter time scale. For this reason the time range from 290 000 seconds to 350 000 seconds of the figure 2.5 is magnified in the figure 2.6 which shows the delay of DNS lookups averaged over 200 second intervals. Only results for the inter-arrival time of 120 seconds are shown in this figure. The figure 2.6 shows that all client groups are practically synchronized due to the short TTL value (600 seconds) after the DoS attack starts at the time of 300 000 seconds. The client groups gradually desynchronize due to the randomness in both the DNS lookup initiation process and the request-response times. The figure 2.6 also shows the delay until the dynamic TTL mechanism begins to enhance the DNS performance after the attack is detected (the high peak at the time of 300 000 seconds).

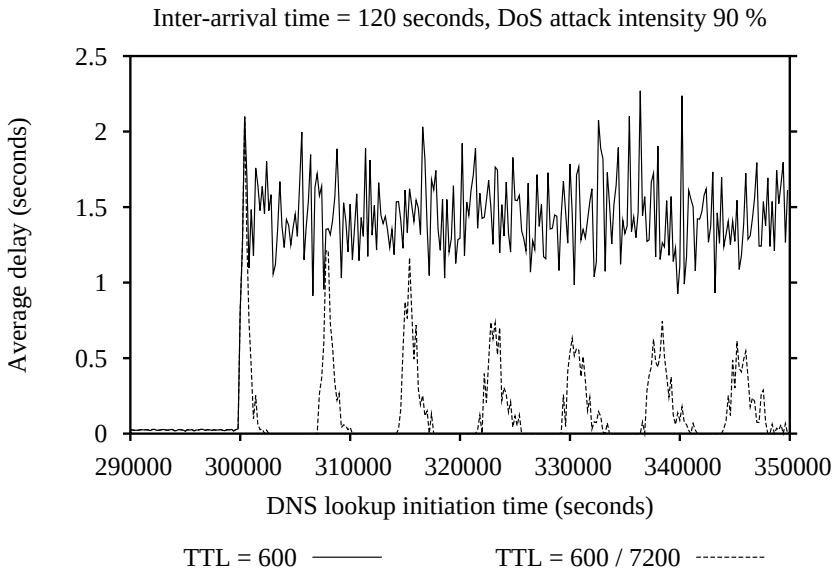

Figure 2.6: The delay of successful DNS lookups averaged over 200 second intervals during the time range from 290 000 to 350 000 seconds.

If the detection delay was considered in the simulations, the transient behavior would most probably look similar to that in the figure 2.6, but that the transient behavior would begin at the time when the attack is detected and the TTL is increased.

2.2.3.2 The Percentage of Failed DNS Lookups

The average percentage of failed DNS lookups is shown in the figure 2.7. The X-axis indicates the time in seconds when a DNS lookup was initiated. The Y-axis indicates the percentage of failed DNS lookups averaged over 10 000 second intervals. A DNS lookup fails if it times out completely at a resolver without an answer. This figure shows the average percentage of failed DNS lookups when the dynamic TTL mechanism is used to protect the subdomain 1 (*TTL = 600/7200)* and when this mechanism is not used (*TTL = 600*). The results are shown for inter-arrival values of 7200 seconds (thick lines) and 120 seconds (thin lines).

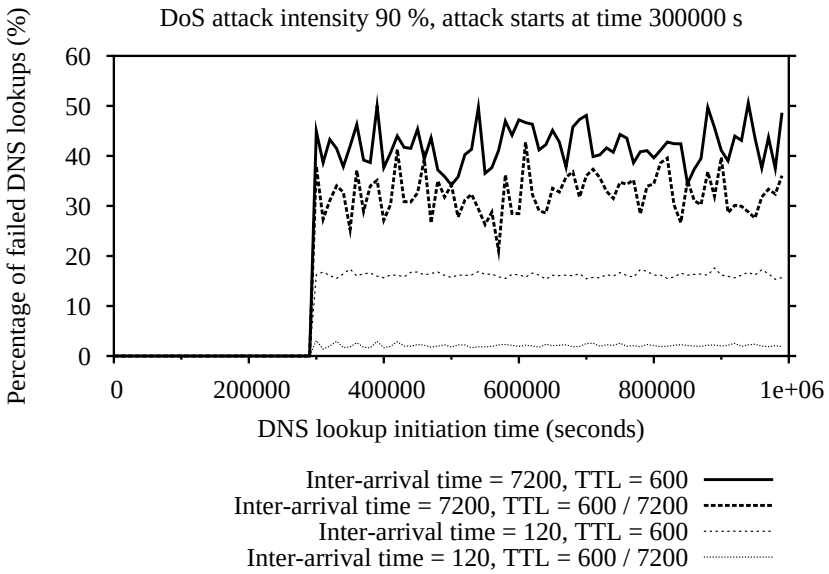

Figure 2.7: The percentage of failed DNS lookups averaged over 10 000 second intervals.

The positive effect of the dynamic TTL mechanism is visible with both inter-arrival times. The shorter inter-arrival time (120 seconds) results in more cache hits at the local name server, which decreases the need to send any DNS requests to the overloaded name servers. This reduces the average DNS lookup failure percentage. Increasing the TTL value of A records from 10 minutes to 2 hours decreases the average percentage of failed DNS lookups from 16% to less than 3%, when 90% of the DNS requests are lost due to a DoS attack (inter-arrival time being 120 seconds).

2.2.3.3 Cumulative Distribution Functions for the DNS Lookup Delay

The Cumulative Distribution Functions (CDF) were calculated for several cases with different parameter combinations. CDF for the DNS lookup delay is defined as follows: CDF(x) = Probability (delay $\leq x$). These CDFs are shown in the figure 2.8 where the inter-arrival time is 7200 seconds. Only successful DNS lookups are included here. Because a resolver will timeout completely after 15 seconds, the delay for all successful DNS lookups is less than 15 seconds.

As can be seen from the figure 2.8 the dynamic TTL mechanism increases the probability of low delays, and decreases the probability of longer delays.

The DNS retransmission policy is visible in these curves. The local name server will retransmit at times of 2 and 4 seconds. At the time of 5 seconds the resolver will timeout and retransmit. After that the local name server will again retransmit with a 2 second interval until the DNS lookup completely times out at the resolver at the time of 15 seconds.

2.2.3.4 The Effect of the Dynamic TTL Mechanism on the DNS Performance

The performance of the DNS during a flooding DoS attack depends on the TTL value. The longer the TTL value, the better the performance. The DNS performance as the function of the TTL value during an attack is shown in the figure 2.9. The default TTL is 10 seconds when no DoS attack is present. The thick lines indicate the average delay of successful DNS lookups as a function of the TTL (left Y-axis). The thin lines indicate the percentage of failed DNS lookups as a function of the TTL (right Y-axis).

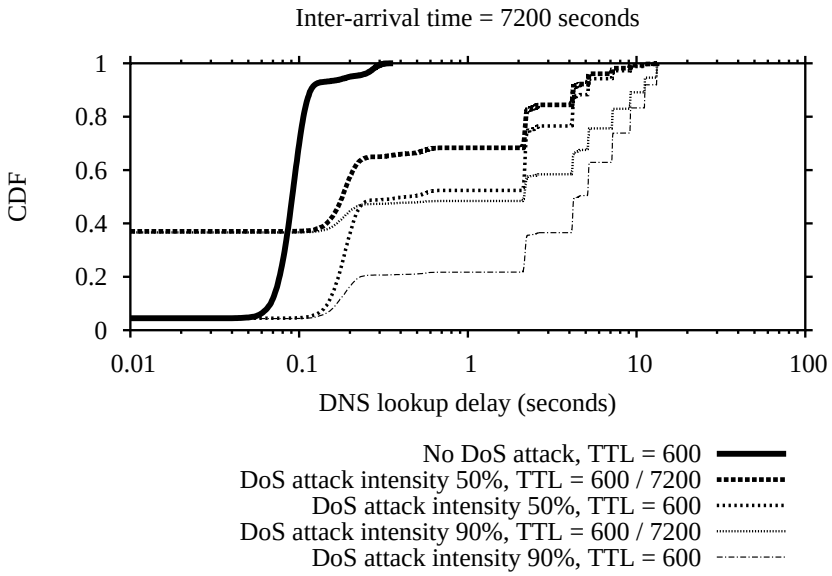

Figure 2.8: The Cumulative Distribution Functions (CDF) for the delay of successful DNS lookups, when the inter-arrival time of DNS requests at every local DNS server is 7200 seconds.

Inter-arrival times of 120 and 7200 seconds were used in these simulations. The DoS attack intensity was 90%.

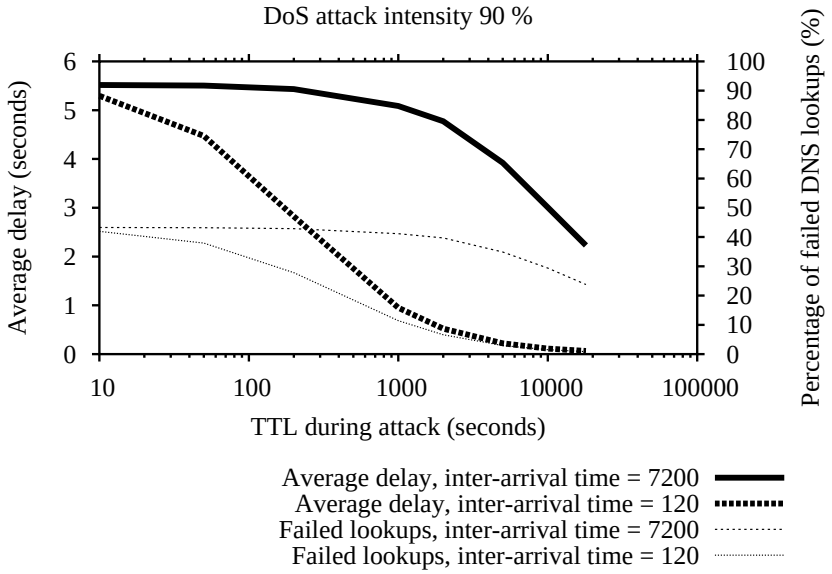

Figure 2.9: The effect of the dynamic TTL mechanism on DNS performance. Thick lines indicate average delay of successful DNS lookups (left Y-axis). Thin lines indicate percentage of failed DNS lookups (right Y-axis). Attack intensity is 90%.

As can be seen from the figure 2.9 the shorter the inter-arrival time, the higher the performance gain from increasing the TTL. When the inter-arrival time at client groups is 120 seconds, a 50 second TTL (default TTL multiplied by 5) will increase the performance by approximately 10% at the most popular destination domain, and a 400 second TTL (default TTL multiplied by 40) will increase the performance by approximately 50%.

DNS-based load balancing depends on a small TTL value for A records. Even though the dynamic TTL mechanism requires relatively long TTL values during an attack, this mechanism can increase the availability of load balanced services. Without any TTL modification many requests for a popular service would fail at the DNS lookup phase, and even a perfect load balancing has no possibility for increasing the DNS performance. The dynamic TTL mechanism will increase the availability at the price of less effective load balancing. This should be seen as a good trade-off. In IPv6 networks one possibility to solve this problem with DNS-based load balancing (application-level anycasting) is to use network-level anycasting [211].

2.2.4 Summary

These simulation results clearly show the benefits of this new simple defense mechanism. For example, when the average inter-arrival time of DNS requests is 120 seconds and the attack intensity is 90%, increasing the TTL from 600 seconds to 7200 seconds during an attack reduces the average DNS lookup delay by 90% from approximately 1.5 seconds to 0.16 seconds. The average percentage of failed DNS lookups was also reduced approximately from 16% down to less than 3%. According to the simulation results the modification of the TTL of A records is a useful mechanism for mitigating flooding DoS attacks against the DNS.

DNS-based load balancing [113, 191] and DoS attack resistance require opposite kinds of changes to TTL values. This problem manifests itself only during DoS attacks when accurate load balancing must be traded off for the increased availability of a service.

2.3 Cross-Layer Security

Trade-offs between layered and cross-layered designs have been discussed for a long time in the area of protocol optimization. Applying cross-layer designs in security, however, is a relatively new approach for creating defense mechanisms [162, 174, 220]. This approach is based on combining security-related information from several protocol layers. The primary goal is to detect possible inconsistencies and malicious conditions related to security in a more reliable fashion. Even though cross-layering can have unintended consequences, such as making it more difficult to understand various relationships between different layers [114], many benefits have been found from using cross-layering. The suitability of cross-layering for increasing the resistance against DoS attacks is shown here in mobile ad hoc networks.

2.3.1 Introduction

Routing in ad hoc networks is significantly different from routing in ordinary wired networks [167]. The wireless medium makes it possible to transmit a message to any node within the transmission range of a sender, and dynamic network topologies place heavy requirements on the convergence characteristics of routing protocols. As the name implies, ad hoc networks can be used to construct temporary networks without administrative intervention or any specific infrastructure devices. Correct behavior of all nodes is required, but as expressed in [219], cooperation is assumed but not enforced in mobile ad hoc networks. Even a single malicious node can thus harm routing in a whole ad hoc network. This makes ad hoc routing an attractive target for denial of service (DoS) attacks.

The range attack described below is a new DoS attack against ad-hoc routing. It is based on changing periodically the transmission range of a wireless node. There is no need to compromise any node because an attacker only has to get close enough to the antenna of a node to be used for the range attack. The goal of this attack is to cause frequent topology changes.

It is analyzed below how the range attack affects application level delays when end-users are downloading web pages from a server node. The research methodology is based on using the ns-2 network simulator to measure delays of web page downloads in a small mobile ad hoc network. These delays are compared when the following three ad-hoc routing protocols are used: The Ad hoc On-demand Distance-Vector (AODV) [166], the Destination Sequenced Distance-Vector (DSDV) [168], and the Dynamic Source Routing (DSR) [110] protocols. The goal of this analysis is to find out how vulnerable these ad-hoc routing protocols are against the range attack.

Two cross-layer designs are presented for mitigating the range attack. A cross-layer design introduces protocol layer interdependencies to optimize overall network performance [52]. Traditionally protocol layers are strictly separated and cannot share network status information between layers.

The first cross-layer design allows the ad-hoc routing protocol layer to share information with the Medium Access Control (MAC) protocol layer. The goal here is to assure that unidirectional links are not used by an ad-hoc routing protocol if the underlying MAC layer requires bidirectional links, such as with the IEEE 802.11 MAC protocol. In the simulations both DSDV and DSR encountered transmission breaks when they tried to use unidirectional links caused by the range attack.

The second cross-layer design allows the application layer and the transport layer to share information. If an application is sending infrequent updates to time-sensitive information which should be transmitted within a specific maximum delay, an application should wait for the previous transmission to be completely acknowledged before sending a next message. According to the simulation results, the usage of this cross-layer design can even double the number of web page transmissions fulfilling a delay requirement during a range attack.

2.3.2 The Range Attack

The range attack is based on modifying the transmission range of a wireless node periodically. The goal of this attack is to cause frequent changes in an ad hoc network topology. Two kinds of range attacks are studied here: attenuating and amplifying range attacks. In both cases the transmission range of a node is changed periodically between the default range and an attack range. The range attack length defines the time how often the range is changed.

In a range attack an attacker must get very close to an internal node of an ad hoc network. The node does not need to be compromised. This attack can be carried out by a malicious insider or by an outsider having access to the networked area.

The range attack causes two kinds of effects. First, it overloads an ad-hoc routing protocol by enforcing it to process continuous network topology changes. Second, it creates unidirectional links when the transmission range of the attacking node is increased. Unidirectional links can degrade the performance of some ad-hoc routing protocols if these links are accepted by an ad-hoc routing protocol. Unidirectional links cannot be used over the IEEE 802.11 MAC.

2.3.2.1 The Attenuating Range Attack

The attenuating range attack was presented in the literature for the first time in the publication P4. In an attenuating range attack the transmission range of the attacking node is decreased, for example, by shielding the antenna with some material which prevents or degrades the propagation of the radio signal.

Isolating an internal node periodically from an ad hoc network will cause regular breaks in multi-hop connections traversing through the attacking node. Serious penalties on application level performance will result.

2.3.2.2 The Amplifying Range Attack

The amplifying range attack against ad-hoc routing has not been described or analyzed in the literature before. In an amplifying range attack the transmission range of the attacking node is increased, for example, by installing an attachment which modifies an omni-directional antenna into a directional antenna. This kind of range extension has been reported, for example, for Bluetooth interfaces where the default maximum range of 100 meters for a bidirectional connection can be extended up to 500 meters by using a high gain directional antenna [25]. Here it is expected that the range extension works only for the transmitted signal, not for the received signal. An extended range can thus only be used for unidirectional links to transmit information to very remote hosts.

The amplifying range attack should theoretically not cause any transmission breaks because no links disappear. Some new links are created, but they are all unidirectional. Of course, throughput can decrease due to added interference from amplified radio signals.

In practice, however, the amplifying range attack can cause severe problems in connectivity, especially in ad hoc networks based on MAC protocols requiring bidirectional links for all unicast messages. An example of this kind of a MAC protocol is the IEEE 802.11, in which all unicast transmissions are preceded with the Distributed Coordinated Function (DCF) consisting of Request-To-Send (RTS) and Clear-To-Send signals (CTS). All unicast messages are also acknowledged at the MAC level. Only bidirectional links must thus be selected by an ad hoc routing protocol in networks using this kind of a MAC.

Ad-hoc routing entities surrounding the attacking node receive route management messages from the attacking node through a unidirectional link. If a routing entity believes that a shorter route is available through the attacking node, it will accept a unidirectional link which cannot forward any messages towards the attacking node. As a result, forwarding of all messages to this link will fail. Depending on the ad hoc routing protocol and the MAC layer, the problem can persist even for the complete duration of the amplified range.

The reason for this vulnerability against the amplifying range attack seems to be the implicit trust for all links being bidirectional. The usage of a bidirectional MAC can increase this false belief. It should be emphasized that all implementations of ad-hoc routing protocols do not necessarily have this vulnerability against the amplifying range attack. To prevent this vulnerability, an ad-hoc routing protocol should make its own checks for bidirectionality, especially over MAC protocols, such as the IEEE 802.11.

2.3.3 Two Cross-Layer Designs for Mitigating the Range Attack

It is very difficult to detect the range attack because it is perfectly normal for the transmission range to vary due to terrain, moving obstacles, and weather. For this reason new mitigation mechanisms are needed. This section describes two cross-layer designs for this purpose.

2.3.3.1 Routing Level Cross-Layer Design

The routing layer and the MAC layer have at least the following overlapping features regarding the range attack:

- Both layers may have different requirements for bidirectionality.
- Both layers can implement tests for bidirectionality.
- Both layers can implement acknowledgement of transmitted messages.
- Both layers can detect disconnected links

All these features should be coordinated, and a cross-layer design is one possibility for this. These two layers should do not make any false assumptions about each other.

The main goal for the proposed routing level cross-layer design is to assure that an ad-hoc routing protocol does not accept any unidirectional links, when the ad hoc network is based on a MAC requiring bidirectional links. This is necessary for mitigating or even preventing the amplifying range attack. When necessary, an ad-hoc routing agent should implement itself additional features to prevent unidirectional links from entering route tables. Also, co-operation between these two layers make it possible to implement acknowledgements and link status detection in the most effective place. Cross-layer designs can make it easier to detect inconsistencies between the properties of these two layers.

2.3.3.2 Application Level Cross-Layer Design

When there is no end-to-end path available to a destination (such as temporarily due to the range attack), all packets to that destination will be queued until a path is again available. The associated delay can be problematic especially for an application sending time-sensitive data, such as updates for information on a web-page. The proposed application level cross-layer design will reduce the possibility for a message being delayed excessively.

The main goal of the application level cross-layer design is to prevent an application from sending new time-sensitive data when previous messages have not yet been acknowledged at the TCP level. Otherwise a new message would just remain in a send buffer waiting for a usable end-to-end path, and during this time the message would gradually lose its timeliness and waste network resources unnecessarily. This cross-layer design thus involves sharing TCP acknowledgement status with an application. When transmitting messages relatively infrequently, a sign of an unacknowledged previous message tells about an unavailable end-to-end path.

The proposed application level cross-layer design is not suitable for every application. The major application area here is expected to be the transmission of information that is refreshed or updated relatively seldom, but that has to be transmitted reliably (over TCP) and in a time-sensitive manner to a receiving host. The information

is resent only when relevant changes occur. Fast transmission of infrequently changing information can be necessary, for example, for situation awareness in military networks. Depending on the delay requirements, an application may require the information to be downloaded very quickly or it may tolerate longer download delays. TCP congestion control issues are thus not relevant for achieving high throughput when the amount of information transmitted at a time is small, such as a simple web page. An application requiring reliable transmission with a short end-to-end delay does not here refer to real-time applications.

An alternative to providing TCP acknowledgement status to an application is the use of the Stream Control Transmission Protocol (SCTP) [200] instead of TCP. SCTP can reduce problems from the head-of-line blocking where a lost packet prevents packets with higher sequence numbers from being delivered to an application, even if these packets would have been received correctly.

2.3.4 The Simulated Ad Hoc Network

The ns-2.28 network simulator was used to investigate the application level performance during range attacks. Two modifications were made to the basic ns-2.28 simulator: nodes were allowed to have different transmission ranges, and the infinite loop problem of the DSDV was patched.

The structure of the simulated ad hoc network is shown in the figure 2.10. This network consists of six nodes, numbered from 0 to 5. The x- and y-coordinates for a node are indicated in parenthesis below each node. The IEEE 802.11 MAC layer is used in the network. All messages are transmitted with the bandwidth of 1 Mbps.

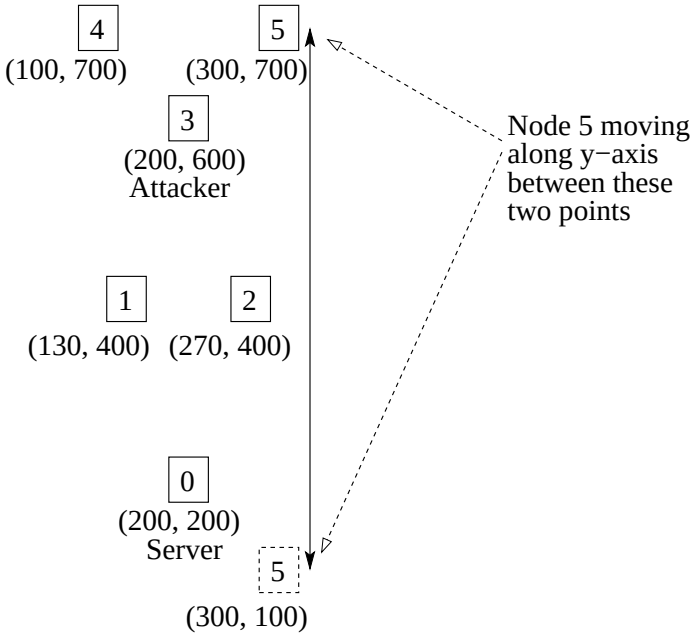

Figure 2.10: Structure of the simulated ad hoc network.

Nodes 0 to 4 are static. The node 5 is moving vertically along the y-axis back and forth between the points (300,700) and (300,100). At the beginning of a simulation it starts moving downwards with the speed of 3 m/s. At the time of 400 seconds it starts moving upwards. The node 5 initiates a movement every 400 seconds.

The node 3 is used for the range attack. The default transmission range for all nodes is 250 meters. In the attenuation range attack the range of the node 3 is reduced

periodically to 40 meters. In the amplifying attack this range is periodically increased to 550 meters.

Client nodes are downloading web pages from the server node 0 with an exponentially distributed inter-page time, the average value being 30 seconds. These pages are downloaded automatically over the TCP protocol. Each web page contains 2920 bytes, which results in two full-size TCP segments. It is expected that persistent TCP connections are used, so the three-way handshake is not required for initiating a download. It should be noticed that the downloaded information does not necessarily have to be a web page because an application is only expected to use TCP for its transmission purposes.

The transmission delay for a download is the complete time to transmit and acknowledge a single web page. This delay is thus the time from the transmission of the first TCP segment to the reception of the acknowledgement of the second TCP segment at the server node 0.

2.3.4.1 Simulation Parameters

The length of each simulation was 60 000 seconds. Increasing the simulation time beyond this value did not cause any relevant changes in the result curves, and the running time of simulations on the available workstation was still tolerable. Due to the memory requirements of DSR, every simulation was divided into 15 independent sub-simulations, each of length 4 000 seconds. All sub-simulations used different random number sequences.

Simulations were repeated with the following parameter combinations:

- The routing protocol was AODV, DSDV, or DSR.
- The range attack length (time between range modifications) was 1, 5, 10, 20, 30, 40, 50, or 60 seconds.

2.3.5 Simulation Results

In all of the simulations, the transmission delay of a web page was measured only to the node 4. The node 4 suffers the most from the range attack, as it uses frequently a multi-hop connection through the attacking node 3. A multi-hop connection through the mobile node 5 is also possible, depending on the location of the node 5.

All graphs have three different curves. The continuous line represents the AODV protocol, the narrow dotted line represents the DSDV protocol, and the broad dotted line represents the DSR protocol.

2.3.5.1 Normal Delay

The figure 2.11 shows the cumulative distribution function (*CDF*) for the delay of transmissions to the node 4. This figure shows the normal delay distribution when there is no attack. $CDF(x)$ is the probability that the transmission delay is less or equal to x.

As can be seen from the figure 2.11, the complete transmission delay is mostly below 0.1 seconds. In case of the DSDV protocol, approximately 5% of downloads experience a relatively long delay of more than 10 seconds.

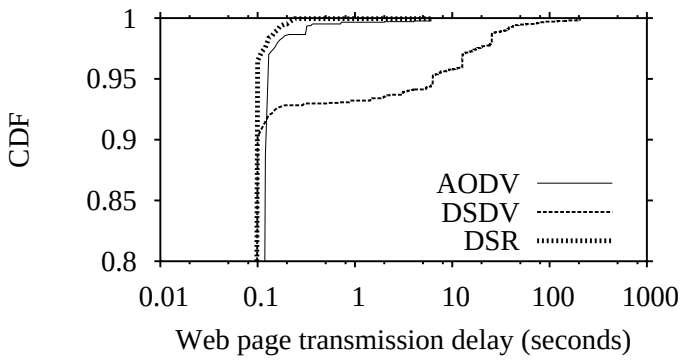

Figure 2.11: Cumulative Distribution Function for the delay of the node 4 when there is no range attack.

2.3.5.2 Delay During an Amplifying Range Attack

The figure 2.12 shows the fraction of transmissions having a delay less or equal to 0.3 seconds during an amplifying range attack. In this case it is expected that the application is transmitting very time-sensitive information. The x-axis in this and all the following figures shows the range attack length.

The amplifying range attack should have no effect on the delay distribution because no links are broken due to this attack. All end-to-end paths remain available regardless of this attack. These results indicate, however, that the implementations of both DSDV and DSR in the ns-2 network simulator are vulnerable to this attack. The AODV protocol is completely insensitive to the amplifying range attack.

The figure 2.13 shows the fraction of transmissions having a delay less or equal to 2 seconds. When comparing figures 2.12 and 2.13 we can see the following:

- AODV is completely insensitive to the amplifying range attack.
- DSR tries to use unidirectional links for short periods of time. It takes approximately 0.6–1.3 seconds to recover from this situation. The effect from the amplifying range attack is visible only in the figure 2.12 where applications require a very short delay. If applications tolerate a delay less or equal to 2 seconds, DSR is practically insensitive to this attack.
- The application level performance collapses with DSDV in ns-2. This is due to an implementation error in the DSDV, and this error causes DSDV to accept and use unidirectional links for as long as these unidirectional links are available. Transmission of unicast messages, however, is not possible over unidirectional links due to the RTS-CTS signaling on the IEEE 802.11 MAC layer.

These results show that ad-hoc routing protocols can be vulnerable to the amplifying range attack, even if it should not be possible according to the specifications. This finding in the ns-2 network simulator should be treated as a proof of concept, that the amplifying range attack can cause denial of service. The use of the proposed routing level cross-layer design would prevent this kind of a DoS attack completely if unidirectional links are rejected immediately.

2.3.5.3 Delay During an Attenuating Range Attack

Figures 2.14 and 2.15 show the fraction of transmissions having a delay less or equal to 0.3 or 2 seconds, respectively. An attenuating range attack is carried out from the node 3.

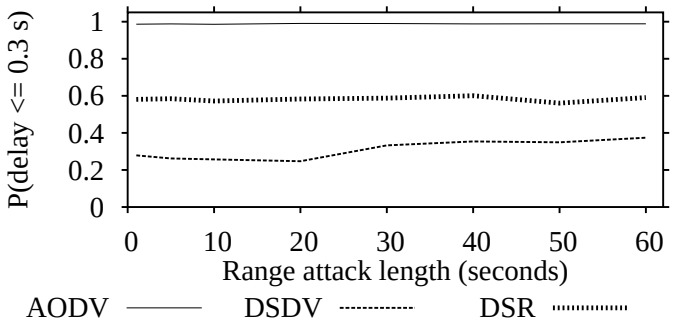

Figure 2.12: Fraction of transmissions having a delay $\leq$ 0.3 s for the node 4 during an amplifying range attack.

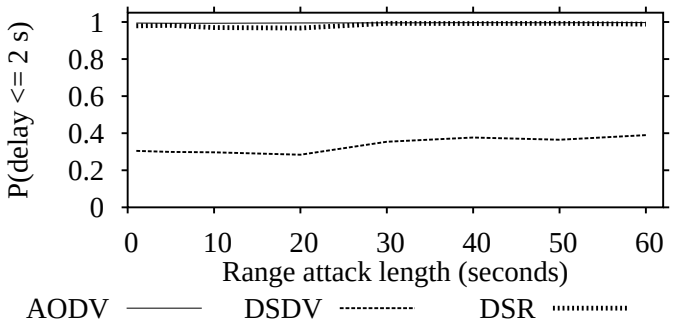

Figure 2.13: Fraction of transmissions having a delay $\leq$ 2 s for the node 4 during an amplifying range attack.

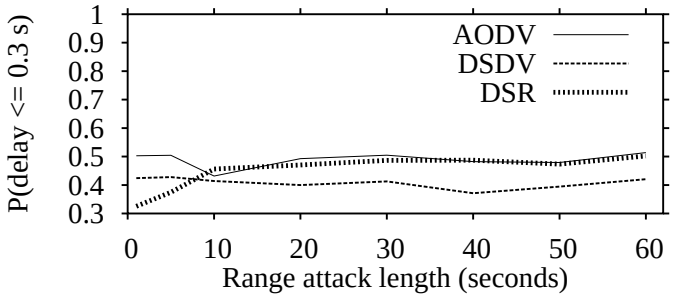

Figure 2.14: Fraction of transmissions having a delay $\leq$ 0.3 s for the node 4 during an attenuating range attack.

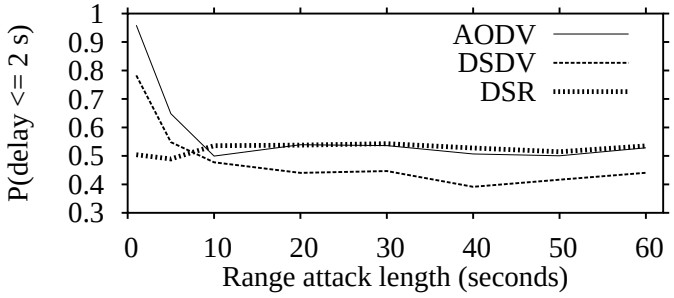

Figure 2.15: Fraction of transmissions having a delay $\leq$ 2 s for the node 4 during an attenuating range attack.

As can be seen from these two figures, long delays result from the attenuating range attack. Approximately 50% of the transmissions fail to have a delay less or equal to 2 seconds. This result is not a surprise because losing connectivity in an ad hoc network will certainly increase delays.

2.3.5.4 Benefits from the Application Level Cross-Layer Design

All previous simulation results were achieved without using the proposed application level cross-layer design. This section shows how delay properties change, when an application waits for the previous transmission to finish completely before initiating a new one. The higher the download frequency, the better the effect from using this design. In the remaining figures the average inter-page time is 5 seconds.

Figures 2.16 and 2.17 show the fraction of transmissions having a delay less or equal to 2 seconds when the application level cross-layer design is not included and when it is included, respectively.

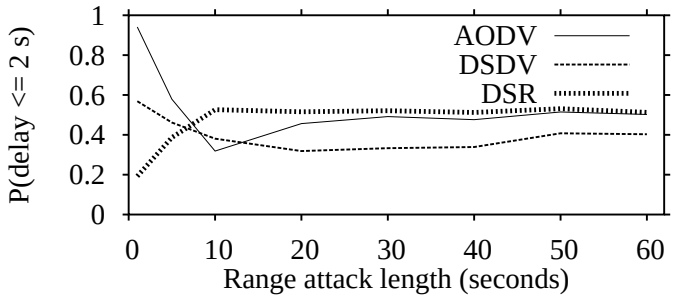

Figure 2.16: Fraction of transmissions having a delay $\leq$ 2 s for the node 4 during an attenuating range attack. Average inter-page time is 5 s. There is no interaction between application and transport layers.

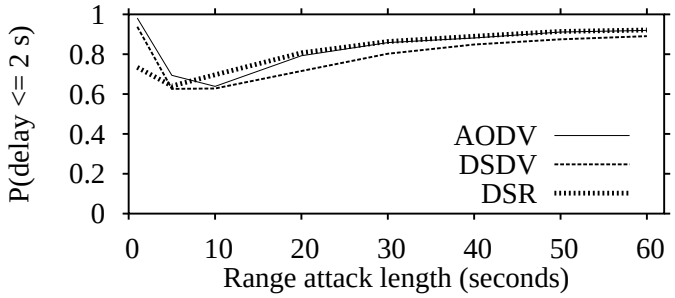

Figure 2.17: Fraction of transmissions having a delay $\leq$ 2 s for the node 4 during an attenuating range attack. Average inter-page time is 5 s. Application waits before previous transmission completely finished.

When the application level cross-layer design is not used, at least 20–30% of the transmissions fulfill the delay requirement. When the application waits for the previous transmission to be acknowledged, at least 60% of the transmissions fulfill the delay requirement. As a result, the number of transmissions fulfilling the delay requirement has been doubled by using this cross-layer design between application and transport layers.

2.3.5.5 Results When All Client Nodes Moving

All previous simulations include only one moving node. To see whether results change when all client nodes are moving, a second simulation setup shown in the figure 2.18 was used. In this second simulation setup all client nodes 1–5 are moving vertically. During movement the y-coordinate of a node changes back and forth between the initial position of the node and the value of 100. All client nodes are moving with the speed of 3 m/s. Nodes 1 and 2 initiate a movement every 150 seconds. Nodes 3–5 initiate a movement every 400 seconds. The server node 0 is static.

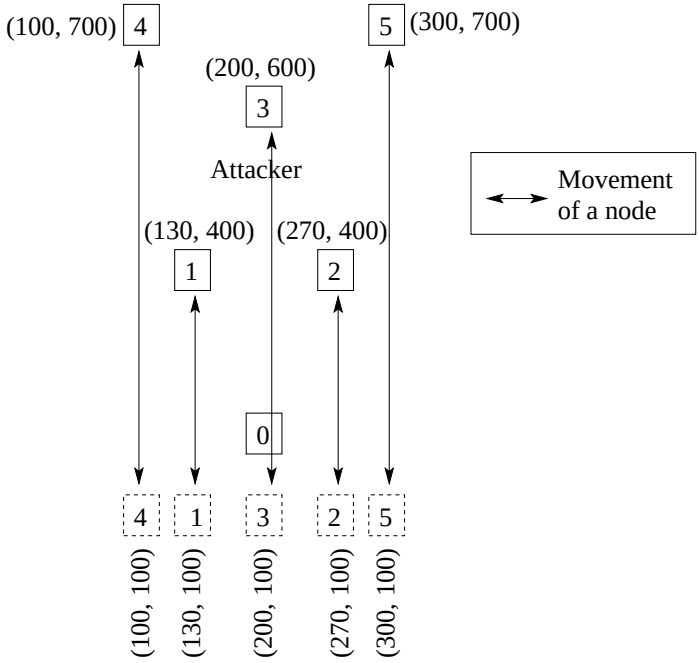

Figure 2.18: Structure of the simulated ad hoc network when all client nodes moving.

In this scenario the higher mobility increases the probability of long delays for the node 4. Approximately 80% of downloads have a delay less or equal to 10 seconds when there is no attack. The node 4 is here more tolerant against the range attack because it has a single-hop connection to the server node part of the time. The application level cross-layer design improved the performance here by 20–30 percent units, when average inter-page time was 5 s.

2.3.6 Related Work

Studies about DoS attacks in ad hoc networks have mostly concentrated on the misuse of routing protocols, such as injecting false routing messages [98, 210, 219]. Few papers have studied other types of DoS attacks in ad hoc networks. For example, the jelly fish attack forces TCP flows to have almost zero throughput by simply reordering, dropping, or causing variable delay to forwarded TCP segments [1]. Mitigation of route request floods in ad hoc networks was studied in [63].

Several cross-layer designs have been proposed for ad hoc networks. Most of these designs concentrate on improving throughput through cross-layer power control, such as in [115]. An architecture for a cross-layer design covering all possible layers is described in [52].

Directional antennas have the possibility of improving the performance significantly when compared to omni-directional antennas. Directional antennas can also

be used to overcome some DoS attacks, such as jamming. A system solution for using directional antennas in ad hoc networks is presented in [175].

2.3.7 Summary

Simulations with the ns-2 network simulator revealed that ad-hoc routing protocols can accidentally accept unidirectional links when an attacking node increases its transmission range. Only bidirectional links should be used over the IEEE 802.11 MAC protocol. This vulnerability is clearly visible with the implementation of DSDV which suffers from a large degradation in performance during an amplifying range attack. The implementation of DSR protocol also tried to use unidirectional links for 0.6–1.3 seconds, which increased the average delay of application level message transmissions. A cross-layer design was proposed here to enforce checks for this important issue and prevent this attack completely. This design makes it possible for the ad-hoc routing and MAC protocol layers to share information about each other to limit the possibility for a mismatch.

Another cross-layer design was proposed here for the application and transport layers. If an application is sending infrequent updates to time-sensitive information which loses its meaning gradually in time, a new message should only be sent when previous messages have been completely acknowledged at the transport layer. This design was tested over the TCP protocol. By using this cross-layer design it was possible to even double the number of transmissions fulfilling a delay requirement of 2 seconds.

According to these simulation results, cross-layering is a promising approach for implementing new defense mechanisms against DoS attacks especially in ad hoc networks.

3 SITUATION DEPENDENT SELECTION OF DEFENSE MECHANISMS

The previous chapter studied individual defense mechanisms specifically designed against different kinds of DoS attacks. The primary goal of that chapter was to present a set of tools (a kind of a tool box) for mitigating DoS attacks. The approach was attack driven, and the most important attack mechanisms were first identified before presenting defense mechanisms. The actual selection of defense mechanisms was simply left for the risk management process.

This chapter concentrates on some important issues that should be considered when selecting defenses against DoS attacks. Defense mechanisms are understood in a liberal way in this section. Any mechanism increasing the resilience against DoS attacks is considered to be a defense mechanism in this section. One can be able to provide higher resilience against specific attacks simply by changing some ordinary building blocks, such as a routing protocol in an ad hoc network. Here, defense mechanisms are expected to be mutually exclusive, so it is possible to select only one defense mechanism at a time.

It is shown here that effectiveness of a defense mechanism in mitigating a DoS attack can be situation dependent. Depending on the characteristics of the environment, different defense mechanisms can turn out to be effective in mitigating an attack. Examples of environmental characteristics are the quality of service (QoS) requirements of legitimate applications.

This chapter consists of two sections. The section 3.1 studies situation dependent selection of defense mechanisms against a new DoS attack in mobile ad hoc networks. The research methodology is based on simulating a mobile ad hoc network and measuring the round-trip delay of transmitted data. The resilience of different ad hoc routing protocols was found to be dependent on the QoS characteristics of the most important applications used in the network. This section is based on the publication P4.

The section 3.2 studies the usefulness of mixed defense strategies against network attacks. The research methodology here is based on using game theory to analyze a simple game where both the attacker and the defender have two mutually exclusive strategies available. This section is based on the publication P7.

3.1 Situation Dependent Resilience against DoS Attacks

This section studies how suitability of a defense mechanism against a specific DoS attack can depend on the QoS requirements of the most important application.

3.1.1 Background

Denial of Service (DoS) attacks are a more serious threat in mobile ad hoc networks than in wired networks due to the complexity, resource constraints, dynamic network topology, open network architecture, and shared transmission media [219]. The higher the complexity of a system, the more possibilities there are to be exploited for attack purposes. Resource constraints restrict the ability to handle and withstand attacks due to limited processing power, transmission bandwidth, and lifetime of batteries. Dynamic network topology places a burden on routing protocols when trying to achieve short reaction and convergence times. Open network architecture and

shared transmission media make it possible to join a network without a physical connection. Any of these vulnerabilities can be exploited in a DoS attack to prevent or delay legitimate access to services.

The primary contribution here is to investigate resilience of three ad-hoc routing protocols against the attenuating range attack which is a new DoS attack against ad-hoc routing (see section 2.3). The routing protocols are the Destination-Sequenced Distance-Vector (DSDV), the Ad hoc On-demand Distance-Vector (AODV), and the Dynamic Source Routing (DSR) protocols. The research methodology is based on using the ns-2 network simulator for analyzing the transmission delay in a small ad hoc network. One node of this ad hoc network is used by an enemy to carry out the range attack.

It is argued in this paper that effectiveness of DoS defense mechanisms is situation dependent, that is, different defense mechanisms are useful for different applications. The simulation results indicate that DSDV provides the highest resilience against the range attack when applications require a very short transmission delay less or equal to 0.1 seconds. When applications tolerate a longer delay up to 2 seconds, AODV was found to provide the highest resilience against the range attack.

3.1.2 Structure of the Simulated Network

For the convenience of the reader, this section 3.1.2 and the figure 3.1 are repeated from the section 2.3.4 and the figure 2.10, respectively. The ns-2.28 network simulator was used to study the resilience of different ad-hoc routing protocols against the range attack. The following two modifications were made to the ns-2.28 network simulator: nodes were allowed to have different transmission ranges, and the DSDV infinite loop problem was patched. The structure of the simulated network is shown in the figure 3.1.

The simulated ad hoc network consists of six nodes, numbered from 0 to 5. The x- and y-coordinates of the nodes are shown in parenthesis below the node. Nodes 0 to 4 are static. The node 5 is moving along the y-axis, and its initial point is (300,

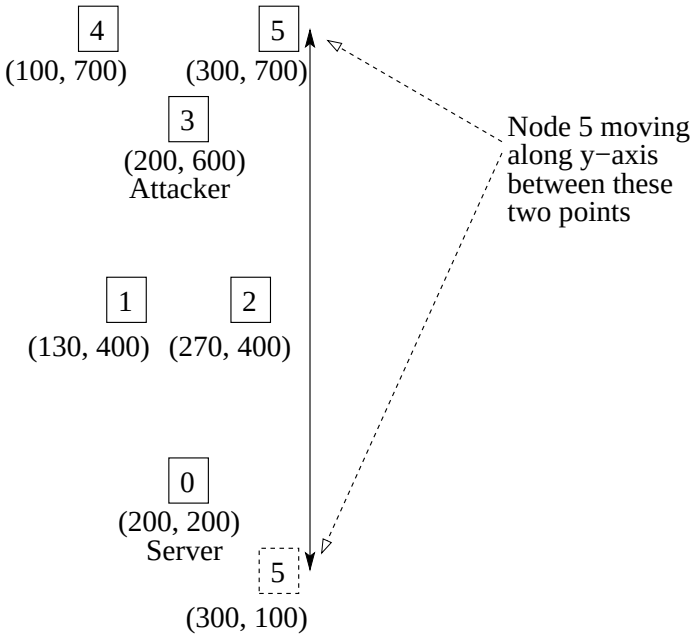

Figure 3.1: The structure of the simulated ad hoc network. The node 3 is used for the range attack.

700). The node 5 is moving back and forth between this initial point and the point (300, 100) at the speed of 3 m/s. The node 5 initiates a movement between these two points every 400 seconds. This scenario reflects a situation where the node 5 is attached to a slowly moving vehicle.

The node 3 is used for the range attack. This reflects a situation where an attacker is able to get close to a node which is used for forwarding traffic from other nodes, but which is still located near the edge of a network for easier access. The range attack length defines the time how often the transmission range of the node 3 is changed. In the simulations the range of the node 3 is alternated between 250 and 40 meters every x seconds where x is the range attack length. The transmission range of the node 3 is thus constant for a period of x seconds.

All five client nodes are automatically downloading web pages from the server node 0 over the Transmission Control Protocol (TCP). Each web page is 2920 bytes in length, which results in two full-size TCP segments. Maximum Segment Size (MSS) for TCP is here thus 1460 bytes. It is expected that persistent TCP connections are used. Each download transaction consists of the transmission of the two TCP segments (required here by one web page) from the node 0. The download transaction is finished when the TCP acknowledgements for both of the TCP segments are received by the node 0. The *transmission delay* of a single web page is thus the time from the transmission of the first TCP segment until the acknowledgement for the second TCP segment has been received at the node 0. After the transaction, the connection is completely ready for the next download. Web pages are downloaded with an exponentially distributed inter-page time. The average inter-page time between two consecutive downloads for each legitimate client is 30 seconds.

It is expected here that legitimate nodes do not initiate new downloads before the previous download is completely finished. A new inter-page time is waited for as many times as needed to finish the previous download. This feature implements the second cross-layer design described in the section 2.3.

All nodes are using the 802.11 MAC with Distributed Coordinated Function (DCF). All unicast messages are preceded with a Request-To-Send (RTS) and a Clear-To-Send (CTS) control messages, and all unicast messages are acknowledged at the MAC level. All broadcast messages are transmitted without any MAC level message overhead. All messages are transmitted with the bandwidth of 1 Mbps.

3.1.3 Simulation Parameters

The length of one simulation was 60 000 seconds because result curves seemed to have stabilized and the running time of simulations with all parameter combinations was still tolerable (2–3 days with the available workstation). Due to the continuous need for allocating new memory in the DSR implementation, every simulation was divided into 15 independent sub-simulations, each of length 4 000 seconds. Half a gigabyte of main memory would not be sufficient for the whole duration of 60 000 seconds if DSR was used as the ad-hoc routing protocol. Each of these independent sub-simulations was using different random number sequences. Simulations were repeated with all the following parameter combinations:

- Attacker's average inter-page time was 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 8, 9, 10, 12, 15, 17, 20, 22, 25, 27, or 30 seconds. This value defines the average time interval how often web pages are downloaded by the node 3. The value of this parameter was varied to see how useful it is for an attacker to create additional traffic during a range attack. Using a value other than

30 seconds (the default inter-page time for legitimate nodes) requires that the attacker has access to applications running in the node 3.

- The range attack length was 1, 2, 5, 10, 15, 20, 25, or 30 seconds. Different values for the range attack length were used to see how the network performance depends on this value.
- The routing protocol was DSDV, AODV, or DSR.

In all the following figures the transmission delays to all legitimate nodes 1, 2, 4, and 5 are combined and shown as a single curve. This makes it possible to get a comprehensive view of the network performance at all legitimate nodes. All graphs have three different curves, one for each ad-hoc routing protocol.

3.1.4 Simulation Results for the Distribution of the Transmission Delay

The figure 3.2 shows the Cumulative Distribution Function (CDF) for the delay to all legitimate client nodes when there is no range attack. Figures 3.3 and 3.4 show the CDF for the transmission delay when the range attack length is 30 seconds and 1 second, respectively.

The x-axis of figures 3.2, 3.3, and 3.4 shows the transmission delay as seconds on a logarithmic scale. The y-axis shows the CDF. CDF(x) is the probability that the transmission delay is less or equal to x. The average inter-page time for the node 3 is 30 seconds in all of these three figures.

Figures 3.3 and 3.4 indicate that the resulting CDF depends on the attack characteristics. DSDV is the most resilient routing protocol against the range attack, when short transmission delays are required by the primary application in an ad hoc network. AODV provides the best overall resilience against the range attack when longer delays are tolerated. DSR cannot provide a consistently good resilience against the range attack.

When comparing figures 3.3 and 3.4 to the figure 3.2 we can see that approximately 10% of the web page transmissions suffer from the range attack. The transmission delay for this 10% of web pages is at least 10 or 100 times the delay without an attack. Practically all of the deterioration in the transmission delay affects only nodes 4 and 5, which are dependent on multi-hop connections through the attacker's node. Nodes 1 and 2 have a direct connection to the node 0.

In the figure 3.3 all CDF curves are relatively flat on the range from 0.2 to 10 seconds. In the figure 3.4 all CDF curves are relatively flat on the range from 0.2 to 1 second. The length of the flat interval depends on the length of the range attack. When the transmission range is small it is not possible to have multi-hop connections through the node 3. This forces TCP to wait until the longer transmission range reenables an end-to-end connection. The longer the range attack length, the longer the flat interval of a CDF curve. Despite of the long simulation times all CDF curves have clear steps. This is mostly due to TCP-timeouts. After a timeout, TCP doubles the next timeout length.

3.1.5 Simulation Results for an Application Requiring Short Delays

This section describes how resilient the three ad-hoc routing protocols are against the range attack when the network is used to transfer information with a very strict

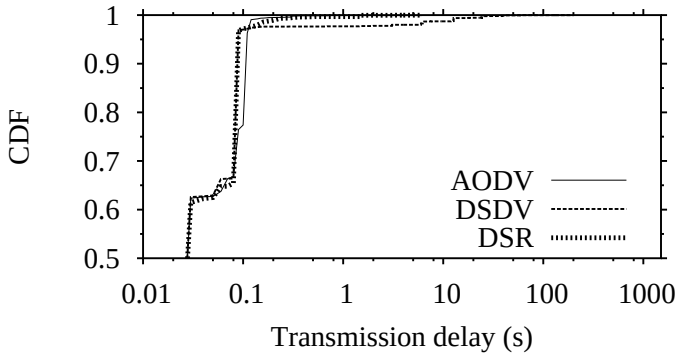

Figure 3.2: CDF for delay of all legitimate client nodes. No range attack.

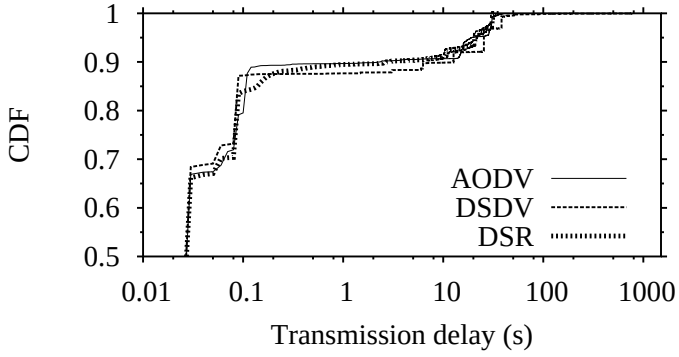

Figure 3.3: CDF for delay of all legitimate client nodes. Range attack length is 30 s.

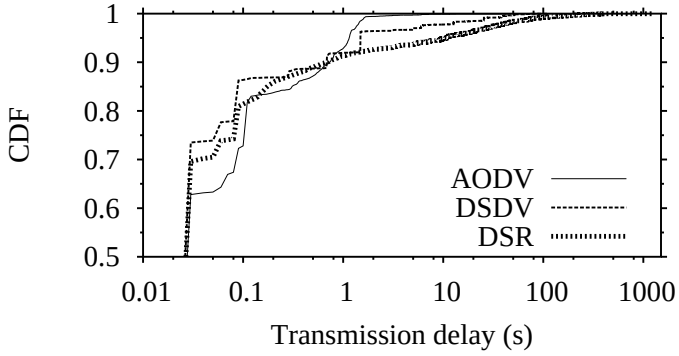

Figure 3.4: CDF for delay of all legitimate client nodes. Range attack length is 1 s.

delay requirement. It is expected here that the information must be transmitted within 0.1 seconds to an automated application requiring no human intervention. If the transmission delay is longer than 0.1 seconds, the information gets useless due to it becoming old. An example of this kind of an application can be the air defense.

Figures 3.5, 3.6, 3.7, and 3.8 indicate the fraction of transmissions having a delay less or equal to 0.1 seconds. In figures 3.5 and 3.6 this fraction is shown as a function of the range attack length, when the attacker's inter-page time is 30 s and 5 s, respectively. In figures 3.7 and 3.8 the fractions are shown as a function of the attacker's inter-page time, when the range attack length is 30 s and 1 s, respectively.

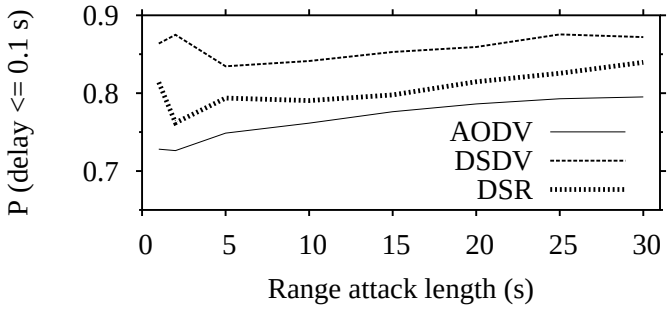

Figure 3.5: Fraction of transmissions having a delay $\leq$ 0.1 s for all legitimate client nodes. Attacker's inter-page time is 30 s.

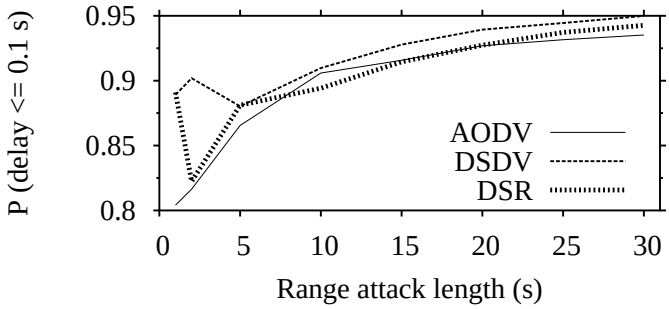

Figure 3.6: Fraction of transmissions having a delay $\leq$ 0.1 s for all legitimate client nodes. Attacker's inter-page time is 5 s.

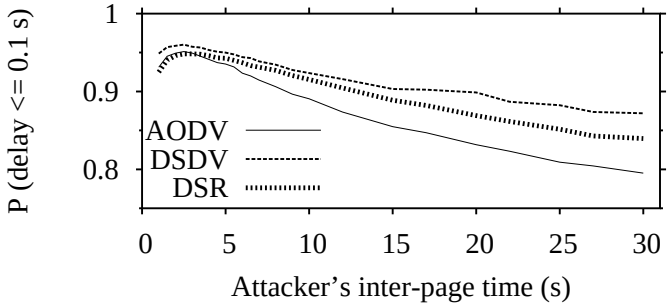

Figure 3.7: Fraction of transmissions having a delay $\leq$ 0.1 s for all legitimate client nodes. Length of the range attack is 30 s.

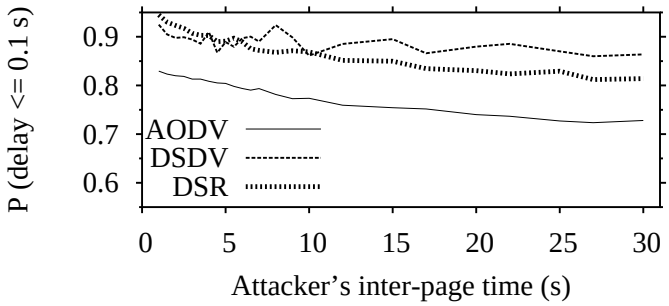

Figure 3.8: Fraction of transmissions having a delay $\leq$ 0.1 s for all legitimate client nodes. Length of the range attack is 1 s.

As can be seen from these figures, DSDV provides the best resilience against the range attack when short transmission delay is required. Here, AODV provides the worst overall resilience against the range attack. One reason for the good performance of DSDV here is that it is proactive and a high proportion of web-page downloads does not have to wait for a route setup.

3.1.6 Simulation Results for an Application Tolerating Relatively Long Delays

This section describes how resilient the three ad-hoc routing protocols are against the range attack when the network is used to transfer information for humans with a flexible delay requirement. The maximum transmission delay allowed is here 2 seconds. An example of an application with this kind of a delay requirement is a situational awareness application providing information for humans. Figures 3.9, 3.10, 3.11, and 3.12 indicate the fraction of transmissions having a delay less or equal to 2 s.

According to these results, AODV is the most resilient protocol against the range attack when applications allow a relatively long transmission delay. AODV is better than DSDV in all these four figures. One reason for the good performance of AODV here is that it is faster in finding a new route than DSDV. With DSDV the delay for setting up a new route after a topology change is at least approximately 12 seconds in ns-2. The performance of DSR varies. Sometimes it provides as good resilience as AODV. In the figure 3.12, however, DSR lags clearly behind of both AODV and DSDV.

Generally, the fraction of transmissions fulfilling the delay requirement increases as the function of the range attack length, as can be seen clearly in figures 3.5, 3.6, and 3.10. The reason for this seems to be the synchronization of the sources with the range attack frequency. Sources will synchronize better when the range attack length has a similar length as the average inter-page time. A web page download is omitted if the previous download is not yet finished, and a new download time is chosen.

In the figure 3.10 both the DSDV and the AODV curves have the lowest performance when the range attack length is 5 s. The fraction of transmissions fulfilling the delay requirement is high when the range attack length is 1 or 2 seconds because applications can tolerate up to one complete period of range attack and still achieve the delay requirement. Also, short topology changes may not even be noticed by a routing protocol. Thus, to make an effective attack, the range attack length should not be too short and not too long.

Surprisingly, the curves in figures 3.7, 3.8, 3.11, and 3.12 are mostly decreasing, which means that ad hoc routing protocols perform better when there is more attack traffic in a lightly loaded network (like the one simulated here). Even though this might seem to be counter-intuitive at first sight, the explanation is logical. As the network experiences frequent topology changes, the first packet trying to use a non-existing link will experience relatively long delays as this packet must wait until the routing tables converge and correct end-to-end routes are available again. The more there is unimportant traffic being sent to the attacker node 3, the more probable it is that an unimportant packet triggers the route setup, for example, through a triggered update or route request. As the ratio between the number of packets to legitimate nodes and the number of packets to the attacker node decreases, a higher fraction of legitimate packets will avoid expensive route setups and experience a low end-to-end delay. If complete flooding is not the goal for an attacker (for example, to make an attack more difficult to detect), the amount of additional attack traffic in lightly loaded networks should be minimized to make a range attack more effective.

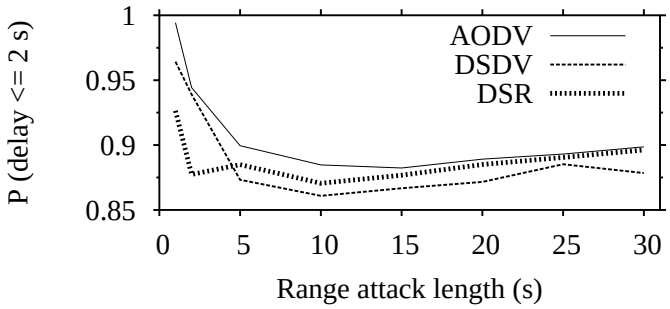

Figure 3.9: Fraction of transmissions having a delay $\leq$ 2 s for all legitimate client nodes. Attacker's inter-page time is 30 s.

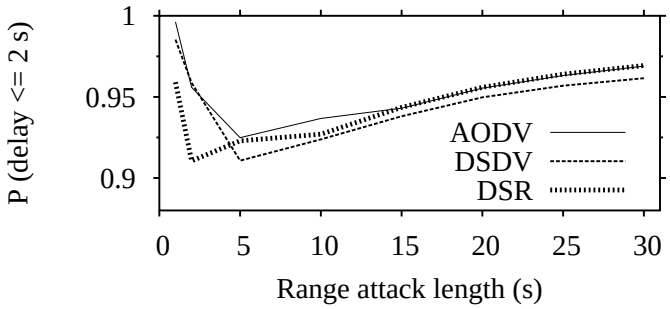

Figure 3.10: Fraction of transmissions having a delay $\leq$ 2 s for all legitimate client nodes. Attacker's inter-page time is 5 s.

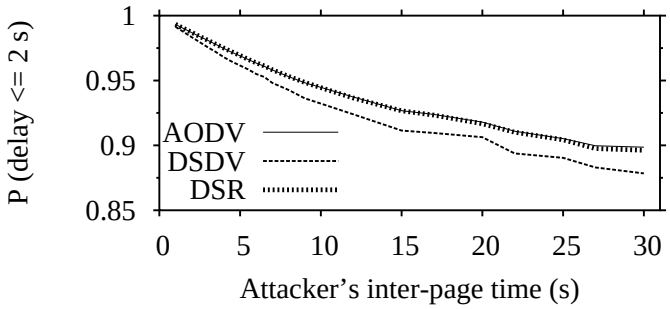

Figure 3.11: Fraction of transmissions having a delay $\leq$ 2 s for all legitimate client nodes. Length of the range attack is 30 s.

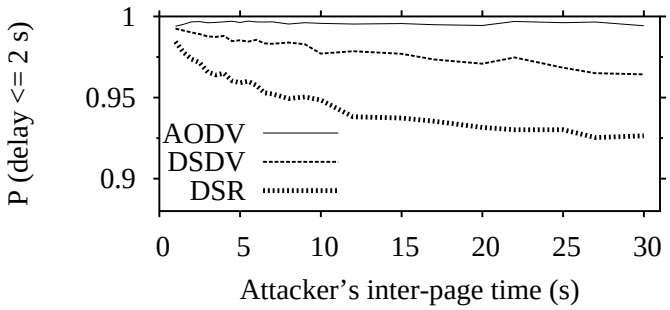

Figure 3.12: Fraction of transmissions having a delay $\leq$ 2 s for all legitimate client nodes. Length of the range attack is 1 s.

3.1.7 Increasing the Resilience against DoS Attacks

The previous results indicate that effectiveness of DoS defense mechanisms is situation dependent. If an ad hoc network is used by an application requiring a very short transmission delay (less or equal to 0.1 seconds), DSDV provides the highest resilience against the range attack and AODV the lowest resilience. This result turns upside down when the ad hoc network is used by an application accepting a longer transmission delay (less or equal to 2 seconds). In this case AODV provides the highest resilience against the range attack. The resilience of ad-hoc routing protocols against the range attack depends thus on the requirements of the primary application used in an ad hoc network.

In general, the characteristics of applications should be considered when evaluating and choosing defense mechanisms against DoS attacks. The following list gives some characteristics important in this process:

- Transmission delay: Does the primary application used in the network have a strict delay requirement, that is, how dependent is the application on a short transmission delay? What is the maximum delay accepted?
- Variability in transmission delay: How much jitter in delay is tolerated?
- Throughput: How much throughput is required for minimum, decent, or good application performance?
- Variability in throughput: How dependent are applications on even throughput?
- Transmission breaks: How does the application react to transmission breaks? What is the maximum length of a transmission break still tolerated by the application?
- Reliability of data transfer: Should all data be transmitted reliably, or is some data loss acceptable? Packet loss can make TCP retransmission timers to have long values, so it can take a relatively long time until a TCP connection is again able to transmit data. Tearing down a passive TCP connection and creating a new one may be a faster way to restore the flow of messages after a topology change.
- Connection initiation delay. If connections are shutdown and rebuilt on demand, how fast should a new connection be created?
- Reaction to network service degradation: If network service degrades heavily, should a connection be rather terminated than continued in a low-quality mode?
- Adaptation capabilities: Is the application able to adapt to varying quality of network service, for example, by varying the compression ratio of a lossy compression algorithm? Is it possible to aggregate data or otherwise decrease the frequency transmissions?
- Application structure: Is an end-to-end connection always required, or is it possible to use intermediate nodes as proxies or caches for nodes with intermittent connectivity? In other words, is it possible to move important data closer to destinations for which an end-to-end connection is not available?

The selection of defense mechanisms against DoS attacks requires first the assessment of application requirements, for example, based on the characteristics in the list above. When the requirements of the primary applications are known, then the properties of available defense mechanisms can be evaluated. The defense mechanisms providing the best match with the requirements should be selected.

Especially in military networks the requirements of applications and even the primary application may change as a function of time. For example, during a tactical attack there is no need to limit the transmission power, and data transmission should be as reliable and fast as possible. When there is no tactical attack going on, however, one should limit the transmission power even up to complete radio silence, and longer delays are tolerated to make interception of messages more difficult. As a result, applications used during a tactical attack may have completely different requirements than those applications used during a non-attack time. For this reason, military networks should be able to change the defense mechanisms according to the primary application. A control system is thus required to change the defense mechanisms to suit the requirements of applications. The following list gives some requirements for a control system:

- If a defense mechanism is used by all individual nodes (like the routing protocol in ad hoc networks), all nodes should start using the new defense mechanism approximately at the same time.
- As an ad hoc network can be partitioned, the control system should be able initiate the change of a defense mechanism at a prespecified time in the future. All nodes should receive this message in time.
- The reception of a control message to change a defense mechanism should be acknowledged system-wide. Depending on the defense mechanism it may not be feasible to start using a defense mechanism in only a subset of the nodes. All nodes should know if all nodes are able to change the defense mechanism.

In the context of this section 3.1, ad-hoc routing protocols are liberally considered to be defense mechanisms because a routing protocol has an effect on the tolerance against the range attack. Here, the control system is necessary for selecting the used routing protocol because only one routing protocol must be used in a network at a time.

Any defense mechanism should be stable enough so that too frequent changes can be avoided. If a defense mechanism fits only a very narrow class of application requirements, a change is needed more often than with defense mechanisms fitting a broader class of requirements.

3.1.8 Related Work

The misuse of routing protocols seems to be the most widely studied subtype of DoS attacks in ad hoc networks, for example in [98], [210], and [219]. Fewer papers have studied other types of DoS attacks in ad hoc networks. For example, [217] describes many possible DoS attacks on different protocol layers in sensor networks, but most of it is applicable to ad hoc networks, too. The jelly fish attack described in [1] forces TCP flows to have almost zero throughput by simply reordering, dropping, or causing variable delay to TCP packets.

The performance of ad-hoc routing protocols has been studied, for example, in [33] and [198]. In [190] the effect of mobility on ad-hoc routing protocol performance was studied. DSDV provided the lowest route acquisition time, packet-delay, and control-packet overhead. AODV provided the best throughput.

The impact of the traffic pattern [172] and the mobility model [109] is significant on the performance of ad-hoc routing protocols. Not only should the application requirements for the network performance be analyzed, but also the traffic pattern and mobility model in general.

3.1.9 Summary

The resilience of different ad-hoc routing protocols against the range attack was described here. The simulation results indicate that this resilience is situation dependent. If an ad hoc network is primarily used by an application requiring a transmission delay less or equal to 0.1 seconds, DSDV provided the best and AODV the lowest resilience against the range attack. If the primary application tolerated a longer transmission delay less or equal to 2 seconds, however, AODV provided the best resilience against the range attack. The results are almost completely opposite in these two cases.

According to the results, it is not possible to provide decent resilience against the range attack with only a single defense mechanism independently of the type of application. It must be possible to change a defense mechanism when the application requirements change. A control system is required for this purpose. This holds especially for military tactical networks where there can be many different QoS classes of traffic, each with multiple levels of precedence [65].

3.2 Modeling Defense Selection as a Game

Game theory is one of the possible ways for modeling network attacks. The publication P7 applies game theory in four example cases. One of these games gives an example of applying game theory in modeling ordinary hacking activity in computer networks where an attacker tries to compromise a victim host. Compromised hosts are a prerequisite for all DoS attacks.

Trying to compromise a host can also be seen as a relevant subgoal for many forms of information warfare, especially for command and control, hacker, and cyber warfare [130]. In information warfare the fundamental weapon and target is information, while the main goal is information superiority [102].

3.2.1 Background

Network security is a new application area for game theory. Some of the first game theoretic publications in network security include [133] from the year 2002, [132] and [10] from the year 2003, and [180] from the year 2004.

Close to network security are also game theoretic studies of misbehavior and selfishness (such as [143, 204]), which can have a clear influence on the performance of ad hoc networks. Game theory has also been applied to analyzing the performance effects of different backoff algorithms in the IEEE 802.11 DCF [218].

Game theory treats multi-person decision problems as games where each player makes decisions which result in the best possible payoff for himself/herself, regarding the most probable decisions of the other players. All players are expected to be rational, that is, they want to maximize their payoffs. An extensive introduction to game theory can be found, for example, in [78].

The normal-form representation of an n-player game G specifies the players of the game, the strategies for each player, and the payoff functions for each player:

$$G = \{S_1, \ldots, S_n; u_1, \ldots, u_n\}.$$

Here, S_i denotes the set of strategies available to the player i, $1 \leq i \leq n$. A strategy $s_i \in S_i$ is one complete plan of action for a game. The payoff for the player i is defined by the function $u_i(s_1, \ldots, s_n)$ where s_j, $1 \leq j \leq n$, is the strategy chosen by the player j.

Games are generally divided into static and dynamic games. A static game is a simultaneous-move game, and all players choose their strategies simultaneously without knowing what strategies other players have chosen. A dynamic game is a sequential-move game, and players choose their strategies in sequence. Static games can be combined in a dynamic fashion, and a repeated game is one example of this.

Games can have different properties. In a cooperative game players try to maximize the joint payoff, but in a non-cooperative game all players concentrate only on maximizing their own payoff. In network attacks, however, there is no cooperation. In a game of complete information the payoff function of all players is common knowledge, but in a game of incomplete information at least one player is not sure about the payoff function of another player.

The strategies in S_i are the pure strategies for the player i. If the player i has M different pure strategies ($S_i = \{s_{i1}, \ldots, s_{iM}\}$) then a mixed strategy for player i is a probability distribution $p_i = (p_{i1}, \ldots, p_{iM})$ where $0 \leq p_{im} \leq 1$ for $m = 1, \ldots, M$ and $p_{i1} + \cdots + p_{iM} = 1$. A mixed strategy indicates one player's uncertainty about what another player will do. A pure strategy s_{im} can be represented as a mixed strategy by setting p_{im} to 1, and the remaining terms in the probability distribution being 0.

Expected payoff v_i for player i in a static game $G = \{S_1, \ldots, S_n; u_1, \ldots, u_n\}$, when player j, $1 \leq j \leq n$, plays the mixed strategy p_j, is the following [78]:

$$v_i(p_1, \ldots, p_n) = \sum_{m_1=1}^{M_1} \cdots \sum_{m_n=1}^{M_n} \left[\prod_{k=1}^{n} p_{km_k} \right] u_i\left(s_{1m_1}, \ldots, s_{nm_n}\right), \tag{3.1}$$

where M_j is the number of pure strategies available to the player j. For a two-player case (3.1) is

$$v_i(p_1, p_2) = \sum_{m_1=1}^{M_1} \sum_{m_2=1}^{M_2} p_{1m_1} p_{2m_2} u_i\left(s_{1m_1}, s_{2m_2}\right). \tag{3.2}$$

When modeling decision problems with game theory it is often possible to identify one or more equilibrium conditions which define a strategically stable prediction of the outcome of a play. In an equilibrium condition no player has interest to change his decision because this would only result in a lower benefit for the player changing his decision. In an n-player static game $G = \{S_1, \ldots, S_n; u_1, \ldots, u_n\}$ of complete information the mixed strategies $(p_1^*, \ldots, p_n^*)$ are a Nash equilibrium if, for each player i, p_i^* is player i's best response to the mixed strategies specified for the other players. In other words, p_i^* solves

$$\max_{p_i} v_i\left(p_1^*, \ldots, p_{i-1}^*, p_i, p_{i+1}^*, \ldots, p_n^*\right)$$

for any player i, $1 \leq i \leq n$. A Nash equilibrium provides a local maximum value for v_i, $1 \leq i \leq n$. Single-player strategy changes are thus not useful at a Nash equilibrium because any two Nash equilibriums (provided they exist) must differ in the strategies of at least two players. So, to change the outcome of a game from one Nash equilibrium to another, at least two players must change their strategies at the same time.

It has been shown that there exists at least one Nash equilibrium, possibly involving mixed strategies, for any normal-form static game with a finite number of players and strategies. If a game has exactly one Nash equilibrium, it is the unique solution to the game. Game theory cannot necessarily predict the outcome of a game if there are more than one Nash equilibriums for the game. Especially, when a game has multiple Nash equilibriums with conflicting payoffs, this game can have an outcome which is not a Nash equilibrium.

Credibility is a central issue in dynamic, sequential-move games. Only credible threats or promises can have an effect on how a game proceeds. In the context of network attacks we will concentrate on threats instead of promises. In a two-player game, one player can threaten to change his/her strategy if the other player does not act as required by the stronger player. If the threat is not accepted, the play will end up in another Nash equilibrium which provides a worse outcome for the player not accepting the threat. Credible threats about future behavior can thus have an influence on current behavior.

3.2.2 Usefulness of Mixed Defense Strategies

This section describes and analyzes one example game for showing the usefulness of mixed defense strategies. The main contribution of this game is to show that using mixed defense strategies can result in better tolerance against dominative attack strategies, when no generic defense strategy is available. Both attack and defense strategies are thus expected to be mutually exclusive, that is, both attacker and defender can choose and use only one strategy at a time.

The studied game is a static game with complete information. There are two players, an attacker A and a victim B. Both players are rational. The goal of the attacker is to compromise a host to be later used for DoS attack purposes (for example, as part of a botnet). To achieve the goal, the attacker must find a primary attack which cannot be directly detected (for example, a zero-day exploit).

The attacker A has two strategies π_{A1} and π_{A2} where π_{A1} means overloading the victim with many secondary attacks at the same time with the primary attack. The large number of different kinds of secondary attacks makes it more difficult to identify any signs of the primary attack. The strategy π_{A2} means trying only the primary attack. The ultimate goal of the attacker can be, for example, the creation of a botnet to be used later in a DDoS attack.

The victim B has two strategies to be used in an intrusion detection system, π_{B1} and π_{B2}. π_{B1} means detecting and alerting on all suspicious network traffic. As all suspicious traffic cannot be prevented from entering the victim site (one possible example being network scans) due to a high number of false positives, at least part of suspicious traffic reaches a victim host. π_{B2} means detecting and alerting on only the most important attacks.

The primary attack will succeed initially because it is based on an unknown vulnerability for which there is not fingerprint. The victim may, however, be able to detect the attack after a delay if he pays enough attention to the available alerts. At least some of these alerts may indicate a malicious action. If there is enough reasons to suspect an intrusion, the victim can look at other traces of network traffic or host computer log files.

If there are only rare bursts of alerts, the above mentioned procedure can succeed in identifying compromised hosts within a reasonable time. On the other hand, if there is a continuous stream of alerts, this procedure cannot work due to the high amount of human intervention required.

The payoffs u_A and u_B for the attacker and the victim, respectively, are the following:

- $u_A(\pi_{A1}, \pi_{B1}) = 4$ and $u_B(\pi_{A1}, \pi_{B1}) = -5$: the victim does not detect the primary attack at all because the victim is overloaded by the large number of alerts caused by secondary attacks.

- $u_A(\pi_{A1}, \pi_{B2}) = 3$ and $u_B(\pi_{A1}, \pi_{B2}) = -2$: the victim does not detect the primary attack, but some of the secondary attacks may cause alerts. These alerts can be used as an indication of a larger attack, and signs of intrusions into important hosts can be searched more carefully. There is a low probability of identifying the successful primary attack.

- $u_A(\pi_{A2}, \pi_{B1}) = 1$ and $u_B(\pi_{A2}, \pi_{B1}) = -1$: the victim detects all suspicious network traffic. As there is only a low volume of alerts, the victim has time to analyze all data carefully, and the intrusion is detected after a delay.

- $u_A(\pi_{A2}, \pi_{B2}) = 5$ and $u_B(\pi_{A2}, \pi_{B2}) = -3$: the victim does not detect the primary attack.

Let us assume that the attacker A plays the mixed strategy $(p, 1-p)$ where $0 \leq p \leq 1$, and that the victim B plays the mixed strategy $(q, 1-q)$ where $0 \leq q \leq 1$. The expected payoff v_A for the attacker according to (3.2) is

$$\begin{aligned} v_A &= 4pq + 3p(1-q) + 1(1-p)q + 5(1-p)(1-q) \\ &= p(5q-2) + 5 - 4q, \end{aligned}$$

and the expected payoff v_B for the victim is

$$\begin{aligned} v_B &= -5pq - 2p(1-q) - (1-p)q - 3(1-p)(1-q) \\ &= q(2-5p) - 3 + p. \end{aligned}$$

The Nash equilibrium points (q, p) for this game are calculated by analyzing the best-response correspondences $p^*(q)$ (the value of p which maximizes $v_A(q)$) and $q^*(p)$ (the value of q which maximizes $v_B(p)$). These correspondences describe how the own optimal mixed strategy selection probability depends on the opponents probability. The figure 3.13 shows these best-response correspondences $p^*(q)$ and $q^*(p)$ which intersect only at one point $\left(\frac{2}{5}, \frac{2}{5}\right)$ which is the Nash equilibrium and the solution for this game.

In this specific game the attacker should overload the victim in 40% of the attacks, and use only a single primary attack mechanism in 60% of the attacks. The victim, on the other hand, should try to detect all possible suspicious network behavior 40% of the time, and concentrate only on detecting the major attacks 60% of the time. The mixed strategy Nash equilibrium reflects the mutual uncertainty about the other player's strategy.

This result has the following effect on the selection of mutually exclusive attack and defense strategies. When defense strategies work well only against different subsets of expected attacks, then changing the defense strategy more or less randomly results in better tolerance against network attacks. The attacker has thus more difficulties in succeeding when a victim is using mixed defense strategies.

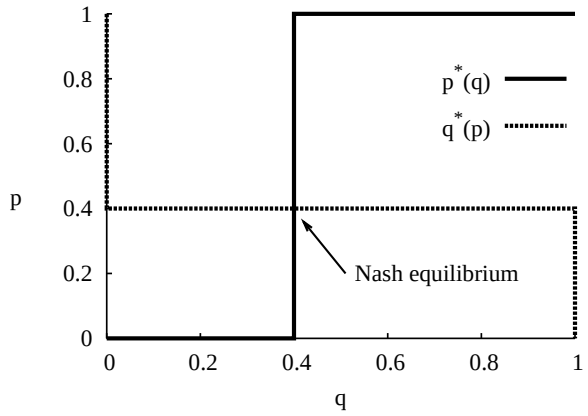

Figure 3.13: The best-response correspondences $p^*(q)$ and $q^*(p)$ have one intersection at $\left(\frac{2}{5}, \frac{2}{5}\right)$ which is the Nash equilibrium and the solution for this game.

3.2.3 Meta-Strategies and Perception Management

Game theory emphasizes correct understanding of costs and strategies, as this is necessary for succeeding in the decision process. A widely used model for describing decision processes in warfare is the Observe-Orient-Decide-Act (OODA) loop [30]. The party that is able to get inside the opponent's OODA-loop will control how fast the opponent is able to react to new situations, and also, what decisions the opponent is making. This is information superiority, and the essential goal for information warfare [102].

The publication P7 describes shortly how it is possible to model the modification of the opponent's OODA-loop. All complete-knowledge games rely on accurate knowledge of the payoff functions. In real-life any player must observe and make as realistic assumptions about these payoffs (costs) as possible. If the observations about an opponent's costs are unrealistic, a player can end up choosing a non-optimal strategy. In the OODA-loop language this is expressed as making a wrong decision.

By combining game theory and the OODA-loop we can end up in a meta-strategy which defines how to influence the strategy selection process (decision process) of the enemy. The primary goal of this kind of a meta-strategy is to make threats look credible, that is, lure the opponent into making wrong observations about the costs of the attacker. If an opponent believes some threats from an attacker, this opponent will play according to these false observations, and the attacker is allowed to enter at least partially inside the OODA-loop of the opponent. As a result, the quality and speed of the opponent's decisions will be decreased.

The meta-strategies described here are closely related to perception management where the goal is to force an opponent to perceive involved costs in an unrealistic way. Skillful use of perception management is, however, difficult because perception and its interpretation cannot be anticipated reliably, and manufactured perceptions may even turn out to be counter-productive [99].

There are, however, real-life cases of perception management, such as using the Internet to distribute misinformation in order to influence stock prices. The term covert cognitive hacking is used in [60] to mean the malicious usage of legitimate-looking information to change someone's behavior.

3.2.4 Summary

A game theoretic approach was used here for studying the selection of defense strategies. A mixed defense strategy was here found to be the most effective approach for mitigating an attack. This means using different defense strategies with a certain probability.

Game theory is very dependent on the correct perception of payoffs and strategies. The goal of meta-strategies is to change the way how existing strategies and payoffs are perceived. If a victim perceives some important issues in an erroneous way, he will make a wrong decision about how to defend against the expected attacks.

4 EVALUATION OF DEFENSE MECHANISMS

Managing the risk created by DoS attacks requires good knowledge in the strengths and weaknesses of defense mechanisms. Before making a decision to implement a specific defense mechanism, one should evaluate the following characteristics.

- How well does the defense mechanisms mitigate the risk from the primary (expected) attack type?
- What is the residual risk related to the primary (expected) attack type?
- What kind of new risks arise? Defense mechanisms increase the complexity of a system which can result in more vulnerabilities to be exploited by attackers.

Evaluation of defense mechanisms is typically based on very limited criteria. Sometimes evaluation is carried out under ideal conditions where there are no false positives and no collateral damage, such as when studying rate limiting in [199]. Three existing preventive defense mechanisms were compared in [156], and it was found out that they were originally evaluated according to restricted criteria, ignoring the risk of attacks that they cannot solve. In [134] rate limiting is carried out against any excessive traffic aggregate, whether malicious or benign, without estimating the possible damage to legitimate traffic. It is, however, emphasized in [134] that it is difficult to differentiate between innocent and attack traffic. A similar limitation is in the system where the end-system resource usage is regulated at a network router [75]. In [3] it is argued that a relatively complex Internet Indirection Infrastructure (*i3*) overlay structure to mitigate DoS attacks does not introduce any new vulnerabilities. This overlay structure requires new nodes and new protocol mechanisms, and there is no guarantee that they are free from vulnerabilities. For example, [3] assumes that an *i3* node cannot be compromised but does not explain how this could be guaranteed. [187] describes a defense against TCP SYN attacks and claims that the presented Synkill mechanism “can detect the conditions of a SYN flooding attack and react appropriately”. But earlier in that paper it is said that an attacker can force Synkill to accept addresses that are actually spoofed (false negative), or to deny legitimate connections (false positive). The experiments in [187] included test cases in which Synkill was able to perform well (which is good). It would have been interesting, however, to also see some experiments about the cases with false positives or false negatives, as this would help to identify in more detail both the advantages and limitations of this defense mechanism. In general, some commercial products promise to “prevent DoS attacks”, which implies that difficult issues in mitigating flooding DoS attacks are forgotten. All these examples show that it is difficult to make an extensive evaluation of a defense mechanism.

It should be emphasized here that all well-specified defenses have definitely their application areas regardless of a restricted evaluation. The point here is that an extensive evaluation will help to identify the limitations and the best application areas, and to get the best benefit out of a defense mechanism.

An organized approach for evaluating defense mechanisms is missing. As expressed in [156], very little has been done to compare, contrast, and categorize the different ideas related to DoS attacks and defenses. A user of a defense mechanism should be able to consider both advantages and disadvantages in an objective way. A taxonomy of DDoS attack and DDoS defense mechanisms is presented in [146]. This taxonomy, however, lacks all evaluation issues.

Only few organized approaches for analyzing denial of service attacks and defense mechanisms have been published in addition to the taxonomy presented in [146]. A framework of criteria for evaluating proactive DoS defense mechanisms is presented in [156] which compares three existing preventive solutions according to the following requirements: incremental deployment, resistance to traffic analysis, resistance to compromised infrastructure routers, and resistance to DoS attack on the infrastructure. A cost-based framework for analyzing the resistance of cryptographic protocols against DoS attacks is presented in [140]. It provides a formalized mechanism for comparing the costs of a DoS attack for both an attacker and a victim. A framework for classifying DoS attacks as either single- or multi-source is presented in [101], and it is based on analyzing packet headers, the ramp-up behavior of attack traffic intensity, and the spectral content analysis of the inter-message time. The number of attacking source hosts (DDoS agents) is important for deciding whether it is useful to initiate a source traceback for locating the real source of attack traffic.

This chapter concentrates on evaluating defense mechanisms against flooding DoS attacks. Pure logic DoS attacks are not considered here because they can be prevented by removing the associated vulnerabilities with existing patches or other mechanisms.

The section 4.1 describes a taxonomy of criteria for evaluating defense mechanisms against flooding DoS attacks. This taxonomy gives a list of important issues to be considered during an evaluation process. There are no existing papers providing any framework or taxonomy for this important subject. This section is based on the publication P5.

The section 4.2 evaluates the effectiveness of rate limiting in mitigating flooding DoS attacks. This section emphasizes the QoS requirements of all legitimate traffic flows, including false positives, as it is not possible to reliably avoid them. This section is based on the publication P6.

4.1 A Taxonomy of Evaluation Criteria

This section presents a taxonomy of criteria for evaluating defense mechanisms against flooding DoS attacks. This taxonomy will point out important issues in evaluating a defense mechanism, which will make it easier to carry out a more realistic evaluation. The taxonomy is shown in the figure 4.1 to give a structured overview of this section.

This taxonomy can be considered as an extension to the taxonomy defined in [146]. Together these two taxonomies will help to understand how attacks and defenses operate, and what issues have an effect on the effectiveness and usefulness of defense mechanisms.

4.1.1 Effectiveness

A defense mechanism should be effective in mitigating a flooding DoS attack. This attack time effectiveness, however, is not enough because a defense mechanism should also be effective when there is no attack, that is, during normal time a defense should disturb legitimate traffic as little as possible.

4.1.1.1 Normal Time

Some defense mechanisms are active continuously both during normal and attack time. In the taxonomy of DDoS defense mechanisms specified in [146] these are called preventive mechanisms. Normal time effectiveness of a defense mechanism is very important because attacks against a specific target are rather rare after all.

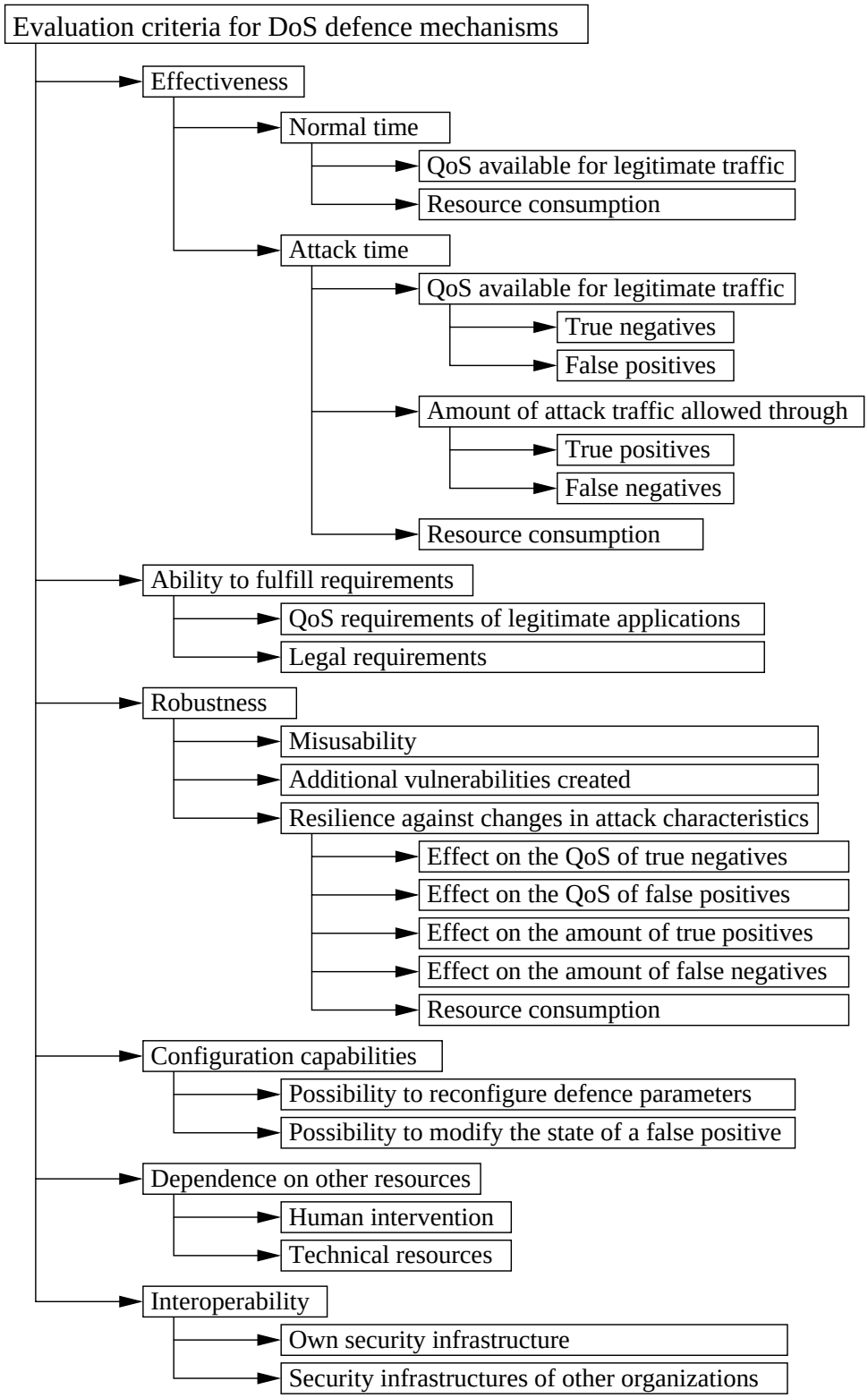

Figure 4.1: A taxonomy of criteria for evaluating defense mechanisms against flooding DoS attacks.

Quality of service (QoS) Available for Legitimate Traffic. A preventive defense mechanism can have an effect on the QoS available for legitimate traffic even if there is no attack due to the overhead from running the defense. The term QoS is understood here in a technical way (intrinsic QoS in [84]). QoS experienced by legitimate flows can deteriorate, for example, if a mechanism introduces additional network security devices increasing the transmission latency, new security procedures requiring additional steps for accessing services, or new security information increasing the length of legitimate packets. An opposite example providing better QoS during normal time is a preventive defense based on resource multiplication which will enhance the QoS experienced by legitimate flows. Resource multiplication is a term used in the DDoS taxonomy defined in [146] but another term for it is overprovisioning.

QoS degradation during normal time has been analyzed, for example, in [152] which describes a proactive secure overlay service structure (WebSOS) to prevent DDoS attacks against web servers. On average the basic WebSOS increases the end-to-end communication latency between a browser and a web server by a factor of 7 when compared to normal routing used in the current Internet. In the worst-case this latency is increased by a factor of 11 when compared to normal routing. In addition to this permanent increase in latency there is an additional security procedure in WebSOS when initiating an access to a web server. The goal of this initial security procedure is to verify that a human is trying to use the web server. This Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) [208] increases the time before a user is granted access to a web server.

QoS degradation due to different authentication and confidentiality mechanisms was studied in [4]. This experiment concentrated on Wired Equivalent Privacy (WEP), IPsec, Remote Authentication Dial In User Service (RADIUS), and Secure Sockets Layer (SSL). These results show, for example, that authentication of a roaming node can take up to 6 seconds, when mechanisms on many different protocol layers are used.

Resource Consumption. Preventive defense mechanisms will consume resources, such as processing power, memory, and transmission capacity of a network.

For example, if IP packets are encrypted/authenticated, each packet will consume more processing power and memory in nodes initiating or terminating a secured path. Also, the increased packet size will consume more network bandwidth.

4.1.1.2 Attack Time

The primary goal for any flooding DoS defense mechanism is to limit the volume of malicious traffic during an attack. In addition to preventive mechanisms, also reactive defense mechanisms can be used, but they require a separate detection mechanism. Reactive mechanisms do not generally degrade QoS of legitimate flows during normal time, that is, when there is no attack. According to the taxonomy of DDoS defenses defined in [146], there are four different basic types of reactive defense mechanisms: agent identification (source traceback), rate limiting, filtering (blocking), and reconfiguration.

QoS Available for Legitimate Traffic. Legitimate traffic is divided in two groups. True negatives are legitimate flows that are classified as legitimate. False positives are those legitimate flows that are classified accidentally as attack flows. Depending on the requirements of the applications used in a network, these two different traffic groups may have similar or different requirements for the QoS. In any case, these both types of legitimate traffic must be considered when evaluating the effectiveness of a defense mechanism.

The QoS available for true negatives is enhanced by preventing part or all of the detected attack traffic from entering a victim network or host. This results in more resources (such as network bandwidth or processing power at a server) available for true negatives.

In case of reactive defenses, true negatives will experience a short period of time of low QoS at the beginning of an attack. There is an inherent delay between the time when an attack is detected and when the corresponding reactive defense begins to mitigate the attack. For example, if the core nodes of the WebSOS overlay network [152] are attacked, the system will heal itself within 10 seconds. This reaction delay should be included in the evaluation of the available QoS during an attack.

The QoS available for false positives, on the other hand, is generally not very high because these flows cannot be differentiated from attack flows. In other words, both false positives and true positives (attack traffic classified as malicious) are associated with the same QoS level.

If all legitimate traffic, including false positives, require a reasonable QoS even during a flooding DoS attack, both preventive defenses (such as resource multiplication [146]) and some of the reactive defenses (such as rate limiting and reconfiguration) can be useful. For example, flooding DoS attacks have mostly failed against the root servers of the Domain Name System (DNS) due to required overprovisioning [37, 207]. Reactive defense mechanisms based on filtering are not suitable if the QoS of false positives is important.

Amount of Attack Traffic Allowed Through. Attack traffic is divided in two groups. True positives are those attack flows classified as malicious. False negatives are those attack flows classified accidentally as legitimate flows. At least all false negatives are allowed to pass through, and possibly part of the true positives (such as in the case of rate limiting).

The probabilities for a true positive and a false positive are inter-related [213]. If true positives must be detected with a high probability, then also the probability for a false positive grows. The same holds also between false negatives and true negatives. If the probability for a false negative must be lowered, then also the probability for a true negative will get lower. In practice this means that the less we allow undetected attack traffic to get in (false negatives), the less we allow legitimate traffic to get in (true negatives). It is not possible to adjust the attack detection so that exactly only legitimate traffic would get in. This fact has been recognized in existing DDoS toolkits which generate attack traffic looking very similar to legitimate traffic [199].

Due to these realities, both true positives and false negatives must be considered in the evaluation of a defense mechanism against flooding DoS attacks. Especially reactive defenses do not reduce the volume of false negatives.

Resource Consumption. Mitigating a detected flooding DoS attack is resource consuming. Limiting the volume of attack traffic may require, for example, several filters for classifying attack flows at routers. There are practical limits to the number of different filters in routers. If attack traffic is highly variable, it may not be possible to install filters for all detected types of attack traffic due to resource limitations.

In general, a defense mechanism can cause performance degradation in involved hosts, such as routers. Enough processing power and memory should be available for mitigating an attack at routers and other network security devices.

4.1.2 Ability to Fulfill Requirements

Different applications have different requirements for QoS. Legislation, standards, specifications, recommendations about best current practices, and other documents may also dictate requirements for the QoS of important applications. All these requirements should be fulfilled as well as possible, even during attack.

4.1.2.1 QoS Requirements of Legitimate Applications

Using a defense mechanism against a flooding DoS attack increases the level of QoS during attack. To see whether the available QoS is sufficient, one must compare this available QoS with the requirements of the most important application (or applications) used in a network. There should be a reasonable match between the available and required level of QoS.

4.1.2.2 Legal Requirements

Legislation or other official rules may require organizations to provide a reasonable resistance to known security vulnerabilities. For example, [117] describes recommended security services and procedures for Internet service providers. DoS attacks found frequently in real-life may have to be considered as known security vulnerabilities. Real-life DoS attacks have been investigated, for example, in [151] and [101].

4.1.3 Robustness

Any defense mechanism should not open any new possibilities for carrying out DoS attacks.

4.1.3.1 Misusability

It is possible to use some defense mechanisms as the ultimate tool for the DoS attack. Such a defense mechanism results in more damage than the original attack itself. Using intelligence in selecting the contents of DoS attack traffic, an attacker may be able to force a defense mechanism to cause more harm than benefits.

4.1.3.2 Additional Vulnerabilities Created

Defense mechanisms increase the complexity of a system and thus result in new vulnerabilities to be exploited by attackers. As attacks can exploit any weaknesses in protocols and services, any additional security protocol, security service, or network security device may provide a possible avenue for carrying out an attack. It is not possible to know all vulnerabilities in a new defense mechanism in advance but there are most likely to be vulnerabilities.

4.1.3.3 Resilience against Changes in Attack Characteristics

Many attacks have varying attack characteristics. Source address validity (how source address spoofing is used), attack rate, possibility of characterization (how easy it is to identify attack packets), persistence of attack sources, and victim type (application, host, resource, network) can be modified on the fly [146].

A defense mechanism should be able to adapt to changes in attack properties. One should also evaluate how varying attack properties affect the QoS of legitimate traffic and the amount of attack traffic allowed to reach a victim.

Frequent changes in attack characteristics may cause excessive resource consumption, for example, by overloading a router when it receives frequent descriptions of attack traffic.

4.1.4 Configuration Capabilities

Any defense mechanism should incorporate reasonable reconfiguration capabilities. This is required when attack characteristics change, or when a critical false positive is identified.

4.1.5 Dependence on Other Resources

To operate effectively, a defense mechanism may require extensive human intervention and several other security devices, such as intrusion detection systems. All these dependencies affect the cost-effectiveness of attack detection and response [126].

4.1.5.1 Human Intervention

If a defense mechanism is dependent on human interaction, this will increase the delays in operating the mechanism. Autonomous defenses should be preferred when primary applications cannot tolerate interruptions in the availability of services.

4.1.5.2 Technical Resources

A defense mechanism is more prone to malfunction if it depends on availability of other security devices. For example, a reactive defense mechanism is useless without a correctly operating detection system.

A requirement for a large number of other security devices or for a wide-spread deployment of a distributed defense mechanism has implications on implementation issues. Incremental deployment is needed for these more complex defense systems.

4.1.6 Interoperability

Defense mechanisms cannot operate in isolation, but instead, they must interact with many different entities. A prerequisite for any defense mechanism is that it must fit with the existing security infrastructure of an organization and be able to co-operate with other existing defense mechanisms, such as intrusion detection systems and security management tools. When global or distributed defense infrastructures are deployed, even higher demands for interoperability exist, such as described in [147].

4.1.7 Summary

Evaluation of defense mechanisms against flooding DoS attacks has often concentrated on easy or simple test scenarios where the possibility to circumvent or defeat a defense mechanism has been either underestimated or completely forgotten.

A taxonomy was presented here to classify evaluation criteria for DoS defense mechanisms. The presented taxonomy emphasizes effectiveness when there is no attack, effectiveness during an attack, ability to fulfill requirements on application QoS, robustness against misuse, resilience against changes in attack characteristics, possibility for dynamic configuration especially for removing critical false positives, dependence on technical resources and human interaction, and interoperability with existing security infrastructure.

4.2 Effectiveness of Rate Limiting

This section analyzes the attack time effectiveness of rate limiting as an automatic reaction mechanism against flooding DoS attacks when the usability of legitimate connections must be preserved. The research methodology is based on analyzing the throughput of a legitimate TCP flow as a function of one-way packet loss rate in a simulated network. Maximum packet loss allowed by legitimate flows also defines how much a DoS attack can be mitigated. The results from simulations are verified with empirical measurements in a small test network.

In this section 4.2 all flows are expected to be TCP-based because the vast majority (83% of packets [137]) of existing traffic and most of DoS attacks (94% of all recognized attacks [151]) are TCP-based. Also, congestion control of TCP is sensitive to packet loss. The effect of packet loss on TCP throughput has been studied for example in [136] and [158], but these studies consider only the loss of TCP data segments. There does not seem to be any studies about TCP throughput when packet loss is one-way, and either TCP data segments or TCP acknowledgements are lost.

4.2.1 Background

Intrusion Detection Systems (IDS) [154] can be used to detect DoS attacks. Reliable detection, however, is not always possible [131, 153, 171]. A well-managed IDS is able to detect many real attack flows (true positives), but it will also mis-detect some legitimate flows as attack flows (false positives). Regardless of this, the first reaction against detected DoS attacks must be automatic because human intervention is slow and attack characteristics can change rapidly in a distributed attack. An automatic reaction mechanism must at the same time try to avoid damages from attack traffic and restrict damages to legitimate traffic.

There are two widely known reaction mechanisms against flooding DoS attacks: filtering and rate limiting [94]. In filtering, all incoming packets of a flow are discarded, and in rate limiting an incoming packet in a flow is discarded with a certain probability. As DoS traffic cannot be easily distinguished from legitimate traffic [199], filtering (blocking) can cause more damage to legitimate user traffic than rate limiting because blocking will completely prevent availability of services to those users whose traffic matches the characteristics of attack traffic. Even though rate limiting is well-known and referenced in many papers, its effectiveness in mitigating flooding DoS attacks has not been analyzed.

4.2.2 Related Work

TCP throughput has been studied in [136], which gives a relatively simple model for the bandwidth BW of a sustained TCP connection, when packet-loss probability p is relatively small:

$$BW = \frac{MSS}{RTT}\frac{C}{\sqrt{p}} \tag{4.1}$$

where MSS denotes the Maximum Segment Size, RTT denotes the Round-Trip Time, and C denotes a constant. This model assumes that TCP avoids retransmission timeouts and always has a sufficient receiver window size. According to the measurements in [136], a TCP connection can withstand a packet-loss rate between 1–10%, depending on the parameters. A more accurate model for the TCP throughput is derived in [158].

Both of these models expect that only TCP data segments are lost. One-way packet loss where either TCP data segments or TCP acknowledgements are lost, is not considered by these models.

Rate limiting as an automatic reaction mechanism against flooding DoS attacks has been studied in [199], which specifies an infrastructure called the Cooperative Intrusion Traceback and Response Architecture (CITRA). The test in [199] demonstrated the suitability of rate limiting in a test environment where legitimate and attack traffic were completely distinguishable, that is, only attack traffic was rate limited. Legitimate traffic passed through without bandwidth or packet-loss penalties.

There do not seem to be any papers analyzing the resistance of TCP flows against inadvertent rate limiting, when a legitimate flow is mis-classified as DoS attack traffic. This is the goal of this section.

4.2.3 Application Areas for Rate Limiting

Effective mitigation of a DoS attack without any damage to legitimate traffic is difficult. As stated in [199], DoS traffic cannot be easily distinguished from legitimate traffic because sophisticated DoS tools generate a packet stream that resembles legitimate traffic. An attacker can also intentionally choose such traffic that maximizes damage to legitimate traffic. This can be seen as an indirect DoS attack based on the side-effect of an automatic reaction mechanism. In this case a countermeasure intended to protect a system from DoS attacks can turn out to be the vehicle for carrying out the DoS attack itself, which in practice means that an automatic reaction mechanism causes more damage than the attack traffic itself. Even in the case of true attacks (true positives) it may be difficult to build an efficient and reliable filter (identification information based on packet data) that matches only the detected attack traffic. Selection of an automatic reaction mechanism is thus a trade-off between the effective mitigation of DoS attacks and the damage to legitimate traffic.

To minimize damage to legitimate traffic and to make it more difficult to turn a countermeasure into an attack mechanism, the use of rate limiting is preferred to blocking as an automatic early-reaction mechanism against DoS attacks.

The key parameter R in rate limiting is the probability of packets being discarded, that is, $(1-R)\times 100\%$ of the identified attack packets are passed through a router. The value of R has to be chosen so that legitimate traffic can withstand the packet loss and that real attack traffic is reasonably dampened. Especially TCP-connections easily close if R is too high.

The major application area of rate limiting is a degradative (non-destructive) flooding attack where a victim is overloaded with incoming packets. The attack traffic is expected to be TCP-based here. Degradative flooding DoS attacks consume resources, such as network bandwidth, processing power, or disk space. Destructive flooding DoS attacks, which cause a permanent DoS condition (for example, by filling disks or crashing several target hosts), can possibly also be delayed enough, so that human intervention has time to prevent a total DoS condition. Rate limiting can also slow down worm propagation, which is important in restricting the effect of fast spreading worms [150, 215].

There are two classes of DoS attacks against which rate limiting is not an effective defense mechanism. First, logic DoS attacks [151] having a single target cannot be rate limited because even one packet can do harm, for example, by crashing, infecting, or compromising a host. Second, very high-bandwidth flooding attacks would require rate limiting with a high value of R, which in practice will approach complete blocking with $R = 1$.

4.2.4 A Suggested Structure and Requirements for a Rate-Limiting System

This section describes a possible structure and requirements for a Rate-Limiting System (RLS). An RLS is an early-reaction system, which automatically reacts to detected DoS attacks.

The Specification and Description Language (SDL) system diagram for an RLS is depicted in the figure 4.2. An RLS consists of an IDS, a control mechanism (RLS controller), a distribution mechanism, and quality of service (QoS) support in RLS-compatible routers. One or more IDSs can reside either in access network links or in end-hosts. IDSs send their DoS alerts to an RLS controller, which creates a filter matching the detected attack traffic. The filter may match both legitimate and real attack traffic due to problems in reliable detection and creation of exact filters. The filter data has to be distributed to upward routers nearer the attack source, for example, by using the Pushback messages [68].

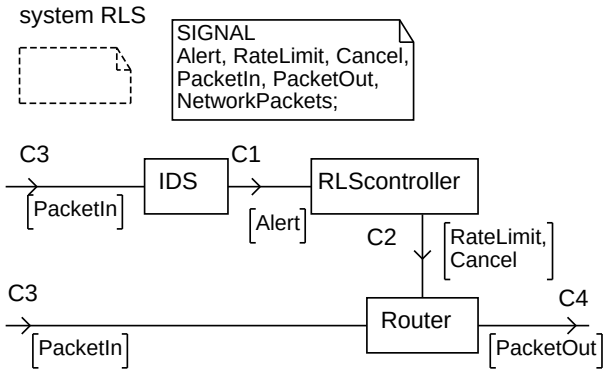

Figure 4.2: The SDL system diagram for a Rate-Limiting System (RLS) consisting of an IDS, an RLS controller, and a router with QoS-support.

The SDL block diagram for a router in an RLS (RLS router) is shown in the figure 4.3. The main parts of an RLS router are an RLS agent, a classifier, a legitimate queue, attack queues, and a scheduler. An RLS agent controls rate limiting in a router by using standard QoS building blocks: a classifier, different queues, and a scheduler. An RLS agent receives messages from an RLS controller. These received messages include the filter data identifying the attack traffic. An RLS agent first creates a new queue for the attack aggregate and then installs the received filter in the classifier. All packets traversing an attack queue are discarded with probability R. All legitimate packets go through the legitimate queue without any added packet loss. The scheduler will transmit all legitimate packets and those attack packets that survive the rate limiting. The maximum number of attack aggregates has to be limited to prevent routers from being overwhelmed with the number of queues to be handled. A message sequence chart (MSC) for a router is shown in the figure 4.4. Error conditions are omitted from these diagrams.

4.2.4.1 Requirements for Actual Rate Limiting in Routers

The RLS requires a basic QoS support from routers, which are expected to classify incoming packets, direct them in different queues, use Active Queue Management (AQM) [47], and finally schedule packets to be sent to an outgoing link. This is shown in the figure 4.5.

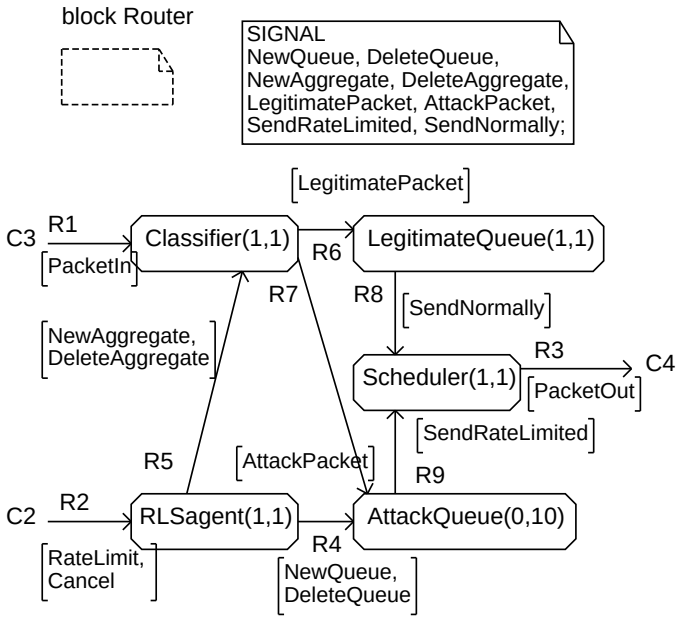

Figure 4.3: The SDL block diagram for a router in an RLS.

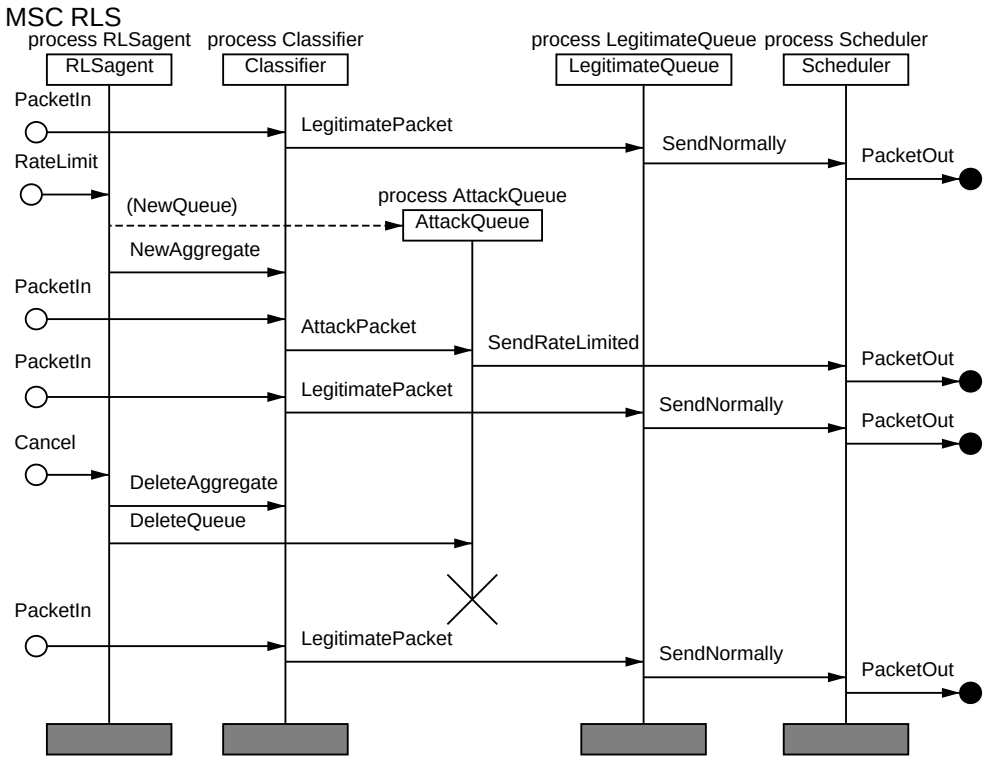

Figure 4.4: An MSC message diagram for a router in an RLS.

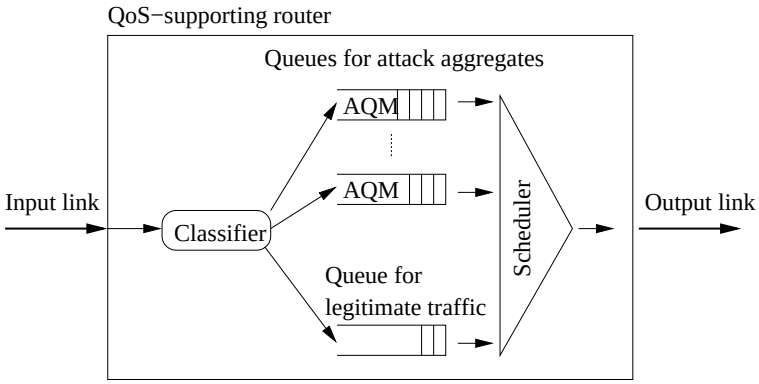

Figure 4.5: Packet flow through a rate-limiting router.

Each attack aggregate is directed to its own queue according to the filter data (like IP addresses, port numbers, and protocol numbers) describing the main characteristics of a specific aggregate of attack traffic. Because the objective is to discard packets from an attack aggregate with probability R, an AQM mechanism can be used. The scheduler must give each queue a fair share of bandwidth because attack queues may also contain legitimate packets. If transmission capacity (link bandwidth) is not the bottleneck, then the scheduler is not a critical point, and there is freedom in selecting a scheduler algorithm and setting its parameters.

The use of an AQM mechanism for rate limiting is preferred here instead of a bandwidth-allocating scheduler because an AQM mechanism can discard fairly accurately a certain proportion of packets traversing a queue. An AQM mechanism does not even need to know the bandwidth of an attack aggregate. A scheduler, on the contrary, needs a reliable estimate of the bandwidth of the attack aggregate, which is not feasible considering the properties of real, fast varying DoS attacks. A scheduler can limit the bandwidth allocated to a queue, but this kind of rate limiting may not even mitigate an attack at all if the bandwidth of an attack aggregate falls below the initially allocated bandwidth. The AQM mechanism chosen should share the bandwidth as fairly as possible, so that the non-responsive attack traffic does not steal bandwidth from the responsive legitimate traffic also belonging to the attack aggregate.

4.2.5 Simulation Results

The effect of one-way packet loss on TCP throughput was investigated by simulating a transmission of a large file with the File Transfer Protocol (FTP). The simulated network included one RLS router to mitigate flooding DoS attacks against the server. The ns-2 network simulator was used in these simulations.

4.2.5.1 The Setup of the Simulator

The topology of the simulated network is shown in the figure 4.6. The legitimate FTP traffic is sent between the FTP client and the FTP server which are attached to the Client router and the Server router, respectively. The RLS router in the middle implements the rate limiting and the related one-way packet loss as an ns-2 error model, which uniformly discards a specific fraction (R) of packets being sent to the Server router. The underlying TCP for FTP applications is of type Reno (TCP/Reno) with a packet size of 1460 bytes. The links between the Client router and the Server router have a bandwidth of 2 Mbps and a delay of 10 ms. The Attack traffic router is connected to the RLS router through a 500 kbps link with a delay of 20 ms.

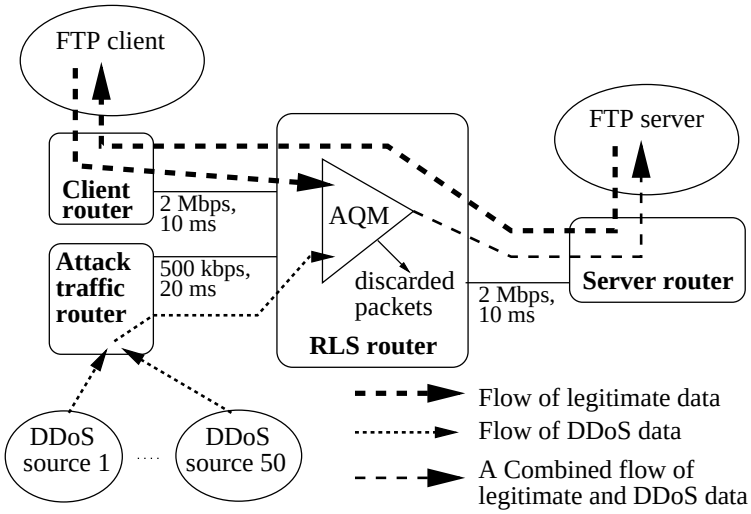

Figure 4.6: The topology of the simulated network. The dotted lines indicate the flow of data. The AQM in the RLS router discards a specific fraction of packets being sent to the FTP server. No packets are discarded by RLS in the reverse direction.

A Distributed DoS (DDoS) attack is simulated in the network with a group of 50 DDoS sources. Each DDoS source sends a large file with the FTP protocol to the FTP server. Attack traffic is thus sent over the TCP protocol (TCP/Reno). These DDoS sources are able to create at most 500 kbps of background traffic due the link bandwidth at the Attack traffic router.

The flow of data packets is shown with dotted lines in the figure 4.6 (the flow of TCP acknowledgements from the FTP server to DDoS sources is not shown in this figure). The FTP client either downloads a large file from the FTP server or uploads a large file to the FTP server. Both legitimate and DDoS FTP packets being forwarded to the Server router are discarded with probability R at the RLS router by an AQM mechanism. The reverse direction for FTP traffic does not encounter any packet loss by the RLS.

4.2.5.2 The Effect of One-Way Packet Loss on TCP Throughput

The simulations consisted of the transmission of a very large file for 100 000 seconds. This simulation time was chosen because it provided reasonably smooth result curves. The amount of data transmitted during this time was calculated from the final TCP acknowledgement received by the sender.

The figure 4.7 shows the simulation results for file upload and download tests when no background DDoS traffic was present. Simulation results in the figure 4.8 show the results of the file transfer tests during a DDoS attack.

The x-axis of these figures shows the packet discard probability R. The y-axis shows the average throughput during the whole 100 000 second simulation as bits per second (bps). The solid thick line indicates the throughput of file downloading, and the dotted thick line indicates the throughput of file uploading. The thin dotted line indicates the theoretical TCP throughput according to equation (4.1) (MSS=1460 bytes, RTT=40 ms, and C=0.45). Even though the theoretical curve is shown for the whole x-axis range, it is valid only with relatively small values of R.

These simulation results indicate that for file upload the one-way packet discard probability R must be below 0.1 for TCP to have a reasonable average throughput. File download, however, is able to withstand a packet discard probability up to 0.5 before the average throughput starts to decline seriously.

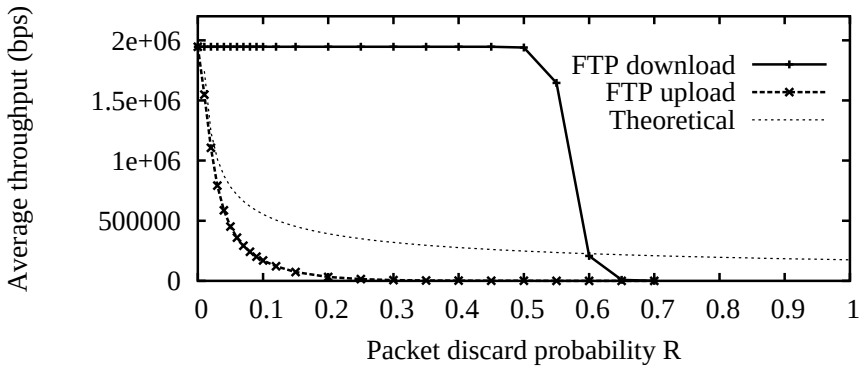

Figure 4.7: The average TCP throughput in the simulator. No background traffic was present.

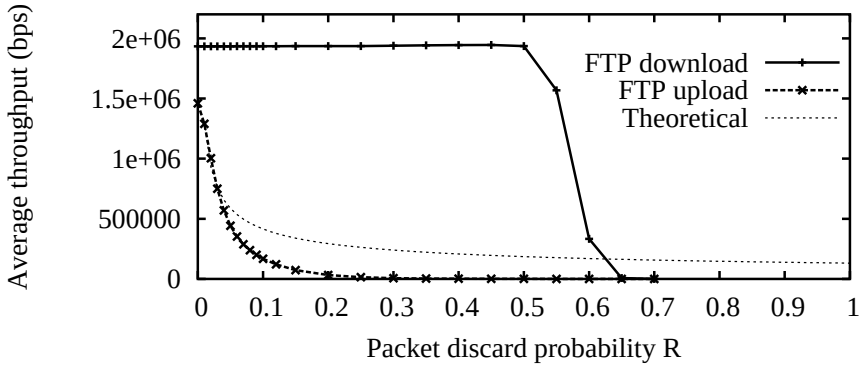

Figure 4.8: The average TCP throughput in the simulator. A flooding DDoS attack was in the background.

The effect of the background DDoS attack is visible only in the throughput of file upload. When uploading a file the bandwidth of the network link from the RLS router to the Server router is shared with the DDoS attack traffic. Two competing types of traffic will share the bandwidth of this link, and less bandwidth is available for legitimate file uploading during a DDoS attack. File downloading is being sent in the reverse direction on this network link, and the DDoS attack does not consume the bandwidth of the link in this direction. Changing the TCP-based DDoS attack traffic to UDP-based (50 Pareto On/Off traffic sources) did not have any visible effect on these results. The local connection from the RLS router to the FTP server is assumed to provide the full bandwidth for both directions at the same time (for example, by separate wires).

The shape of the upload throughput curve matches reasonably well with the theoretical curve. On the other hand, the shape of the download throughput curve differs quite much from the theoretical curve. This difference comes from the type of packets discarded by the RLS router, which applies rate limiting only to the traffic being forwarded to the Server router. When downloading data from a server, only TCP acknowledgements experience increased packet loss, but none of the TCP data segments suffer from forced packet loss at the RLS router. The loss of an acknowledgement does not necessarily require a retransmission because acknowledgements are cumulative. Successive acknowledgements can recover information in earlier lost acknowledgements. A lost TCP data segment, however, cannot be recovered without a retransmission either through a fast retransmission (duplicate acknowledgements) or a timeout. File downloading is thus able to withstand a relatively high proportion of lost acknowledgements because successive acknowledgements make it unnecessary

to retransmit packets.

The theoretical model expects that only TCP data segments are lost with a certain probability. If only TCP acknowledgement packets are lost, the actual throughput curve has thus higher values than the theoretical curve.

4.2.5.3 Suitability of Rate Limiting as a DoS Attack Mitigation Mechanism

The simulation results show that the effect of one-way packet loss on TCP throughput is application-dependent. File downloading tolerates rate limiting better than file uploading to a server protected with an RLS.

According to these results rate limiting is a useful automatic reaction mechanism against flooding DoS attacks. Rate limiting can can discard up to 50% of the incoming attack traffic, but still provide a reasonable service quality for the legitimate users mis-detected as attackers. This can be achieved even when attack and legitimate traffic cannot be distinguished at all.

The effect of random packet loss inherent in real networks was not included in the simulator. Regardless of this, these simulation results show that rate limiting disturbs information downloading much less than information uploading.

4.2.6 Empirical Results

A small test system was implemented to verify the simulation results. Simulation software does not include all real-life effects, like processor load. The goal is to see whether theory and practice match each other reasonably well.

The test network consists of three Linux hosts (kernel version 2.4.20). One host acts as an FTP client and another host acts as an FTP server. The third host acts as an RLS router between the server and the client. The RLS router implements a simple AQM mechanism, which discards packets always with probability R.

4.2.6.1 FTP Download and Upload Tests

The empirical tests consisted of downloading or uploading a 450 kB file from/to the server with FTP. The throughput indicated by the FTP client was recorded after each file transfer. A test was run approximately 15 times for each value of R. There was no background traffic during these empirical tests.

The RLS router was initialized with a specific value of R before any FTP throughput tests were run. The classifier in the RLS router was initialized so that the legitimate FTP traffic is treated as attack traffic. This made it possible to study the effect of rate limiting on legitimate FTP traffic.

The empirical results for the FTP tests are shown in the figure 4.9. The thick solid line indicates the measured average TCP throughput for an FTP download as a function of packet discard probability R. The thick dotted line indicates the measured average TCP throughput for an FTP upload. The thin dotted line indicates the theoretical TCP throughput as indicated by the equation (4.1) (MSS=1448 bytes, RTT=3.9 ms, and C=0.25).

As can be seen from this figure, the empirical results are approximately the same as the simulation results. FTP download tolerates rate limiting much better than FTP upload. The maximum value of R is, however, lower than in the simulations. According to these empirical results FTP download tolerates a packet discard probability $R = 40\%$. Even though these empirical tests were rather short, they support well the simulation results.

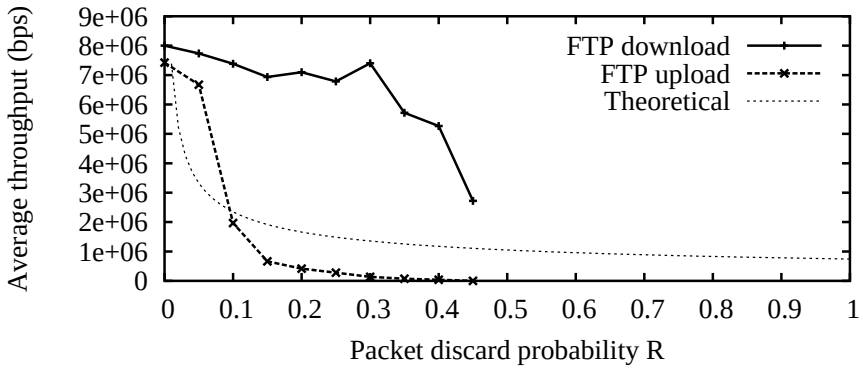

Figure 4.9: The average TCP throughput measured from the test system. No background traffic was present.

4.2.6.2 Web Browsing Tests

To see whether the file transmission results are applicable to interactive traffic, web browsing was tried shortly in the test system.

Web browsing resembles file downloading because most of the data is transmitted on an Hyper Text Transfer Protocol (HTTP) connection from a web server. This direction does not suffer from forced packet loss in the test system. After a TCP connection has been created, only HTTP request and TCP acknowledgement (ACK) packets experience increased packet loss. During the TCP three-way handshake also TCP SYN and TCP ACK packets experience increased packet-loss probability but packet-loss during connection establishment does not influence the results presented here.

The first negative effects can be perceived around the packet-loss rate of $R = 0.3$, but the quality of web browsing remains acceptable up to the packet-loss rate of $R = 0.55$. These results are highly subjective, but indicate the common properties between web browsing and file downloading.

Rate limiting seems to be a suitable DoS defense mechanism for mitigating flooding DoS attacks against WWW servers. Web browsing is an important application type because many e-commerce sites are accessed only by web browsers. Also, well-known web sites have been a target for many published DoS attacks [74].

4.2.7 Summary

TCP throughput was analyzed here in a simulated network which included a rate-limiting feature to mitigate flooding DoS attacks against a server. File uploading to the server was sensitive to rate limiting, and tolerated a packet-loss rate of less than 10%. File downloading from the server, on the other hand, tolerated one-way packet loss much better than file uploading. Downloading was able to tolerate a one-way packet-loss rate up to 50%. File downloading and web browsing are examples of applications, which seem to tolerate well the extra packet loss from rate limiting.

According to these results the effectiveness of rate limiting is limited to decreasing the intensity of a flooding DoS attack by up to 50%, when legitimate users mainly download data with a TCP-based application. This should be seen as a useful result because attack mitigation is possible even when legitimate and attack traffic cannot be distinguished at all. Rate limiting can thus be used as a fast, automatic reaction mechanism to mitigate an attack without any undue penalties for legitimate traffic.

5 DISCUSSION

This dissertation has studied defense mechanisms against DoS attacks. One very relevant question that has not yet been discussed is whether it is possible to define exactly what defense mechanisms an organization or a user should implement to mitigate these attacks. This is mainly the responsibility of risk management as has been emphasized before in this dissertation. There are, however, many practical problems in risk management in achieving an optimal level of security.

Other relevant questions not yet discussed here are related to the reliability of results from simulations and mathematical modeling. The following three sections will discuss these issues. At the end of this chapter are some suggestions for future work.

5.1 Practical Problems in Risk Management

Successful risk management depends on several issues. First, one should be able to understand existing risks and be able to compare them using some meaningful metrics. Second, one must be able to make realistic estimates of both the costs from implementing defenses and the associated benefits, such as the lack or mitigation of security incidents. This kind of cost-benefit analysis is a prerequisite for deciding, whether it is worth investing in a specific defense mechanism [81]. Third, one should continuously control whether the known sets of risks match the actual risks in the fast changing world.

It is not, however, easy to understand objectively those issues that are necessary input for a risk management process. Security in general is a very subjective issue. For example, it is possible to experience and perceive the level of security and possible costs from security incidents in many different ways. Vulnerabilities and threats also have objective characteristics. For example, humans may underestimate or even neglect some threats. Part of difficulties in risk management stem from the ever-changing set of risks, as new risks can be found and exploited any day. Other difficulties include the need to view security as having a multi-disciplined nature. In addition to technical characteristics, security must be understood also as having psychological, organizational, and cultural features. Even though this dissertation has approached network security from a technical point of view, these other perspectives must not be forgotten.

These problems in risk management are discussed in more detail in the following sections. As these problems are common to all sub-areas of security, they can be seen to be related to security in general, and not only network security.

5.1.1 Subjectivity in Security

Security is a controversial issue. When there have been no security incidents for a relatively long period of time, security is easily perceived to be too expensive, and it can be impossible to increase the level of security against new kinds of risks. People can oppose any security procedures as they can feel that these procedures cause unnecessary delays or make their life too difficult. Especially, if security is implemented in a very extensive fashion, it can restrict our ability to get work done [57], which can be seen as a kind of a DoS condition, too. On the other hand, after a

major incident, security is typically blamed for having failed and responsible people might be accused.

One reason for this contradiction is that functionality and security are competing goals, and this forces system designers to make trade-offs [66]. One example is a policy requiring passwords to be changed frequently, independently for each system. This makes it difficult to remember these passwords, especially for those systems which are not used daily. As a result, people forget their passwords and are rejected from using services, or people write passwords down where other people can find them. Another example is the emphasis on encryption technologies, which can decrease the availability of the protected information, for example, when keywords are lost.

Another reason for this contradiction is that it is extremely difficult to define and identify an optimal (good enough) level for security [181]. From a business point of view, the cost of security should be minimized, but at the same time, major risks should be prevented to guarantee business continuity. Of course, the situation is completely different when human lives are at risk than when the risk involves only limited monetary losses.

This contradiction is well-known in the area of human safety, such as in accidents involving humans and organizations ([176], pp. 107–124). Resistance against risks in an organization changes as a function of time. Real accidents force this resistance to be increased. When there have been no accidents for a relatively long time, the resistance against risks can easily drift in the direction of higher vulnerability, as productivity issues become more important than protection issues. Both individuals and organizations can become imprisoned by their prior successful achievements and traditions so that they may not be able to recognize an increased level of vulnerability against accidents ([203], p. 197). Continuous vigilance, objectivity, and self-criticism are needed to assure the appropriateness of security procedures.

There are completely contradictory opinions about how to estimate the costs of security incidents. An underlying problem here is how to estimate the costs of damages, such as lost human lives, bad reputation, loss of productivity, and angry customers. One real-life study about security incidents in computer networks of universities in the United States claimed that the loss of productive working time of all involved people must be included in the calculations to full extent [179]. For example, when a university network is down and undergraduate students cannot use it, the costs of an incident should include the average wage of all undergraduate students for the whole incident time, in addition to all other costs. On the other hand, it has been claimed that the costs of ignoring security in computer networks and getting attacked have so far not really affected the involved companies [185]. The chosen point of view on the costs of security incidents defines much of the attitude towards security.

5.1.2 Dynamically Changing Risks

The set of possible risks is not static. We cannot know all possible risks to prepare against, as new vulnerabilities can be encountered any day. Existing risks also change their severity. One day an issue can be classified as having a very low risk, but the next day it can be associated a high risk. One real-life example here is the terrorist attack in September 11, 2001. New defense mechanisms are continuously needed due to the arms race between attackers and defenders.

5.1.3 Different Areas of Security

Security in both the physical world and the virtual world encounter similar difficulties. In the physical world there are crimes, robbers, vandals, and terrorists. In the virtual world of computers there are computer crimes, hackers trying to steal valuable information, virus-writers, and cyberwarriors. These two worlds are not completely separated as the current society uses information technology heavily for everyday purposes [127]. Attacks against the physical world can be carried out through the virtual world (such as described in [38]), and vice versa. Combined attacks in the both worlds can be effective against a critical infrastructure of a society. As a simple example, it is not enough to only implement an extensive set of network defenses against DoS attacks if physical security has not been taken good care of because an attacker could simply physically destroy, for example, servers, routers, transmission cables, or access to electricity.

Mitigating any kinds of risks requires attention on many different areas, such as on organizational, human, physical, and technical issues. International Organization for Standardization (ISO) has published the standard ISO/IEC17799:2005 [105], which describes best current practices for implementing security in 11 different areas. This standard was originally published as the British Standard BS7799.

5.2 Credibility of Simulations

A prerequisite for credible simulations is a valid model of the studied network. When the simulation models for this dissertation were implemented, the guidance in [69] was considered: a model should concentrate on the relevant properties of the studied problem and make valid simplifications or abstractions for less relevant properties. One should also understand how different parameter settings affect results, and this goes hand-in-hand with measurements, as measurements help in finding useful values for parameters. In this dissertation existing measurement results are used as much as possible to find realistic probability distributions for different random variables.

However, it is not enough just to have a valid simulation model. One should also use valid simulation experiments which can be repeated independently. The following guidelines are given in [163] to achieve valid simulation experiments:

- Information about the used pseudo-random number generators (PRNG) should be given.
- The type of simulation should be specified (terminating vs. steady-state).
- The analysis method of simulation output data should be specified.
- The final statistical errors associated with the results should be specified.

In [163] it was found out that scientific publications using simulations contain very little or no information about how a simulation was run. Approximately 52% of the studied papers reporting simulation-based results did not specify whether their results came from terminating or steady-state simulation. 69–79% of the papers reporting simulation-based results did not care about the random nature of output data generated by stochastic simulation. Vast majority of papers did not specify any information about the time horizon over which systems were analyzed.

According to the above requirements, the simulation experiments in this thesis can be seen to be reasonable, even though not perfect. This is studied in more detail in the following subsections. Simulations are used in the publications P2, P3, P4, and P6.

5.2.1 Pseudo-Random Number Generators

This dissertation uses the ns-2 network simulator for all simulations. This is explicitly specified in all simulation-based publications. According to the ns-manual [202], random number generators used in the ns-2 simulator are an implementation of the multiple recursive generator MRG32k3a specified in [124], and the C++ code is adapted from [125]. The MRG32k3a generator provides $1.8x10^{19}$ independent streams of random numbers, each of which consists of $2.3x10^{15}$ substreams. Each substream has a period of $6.7x10^{22}$.

These PRNGs are used as sources of elementary randomness to implement different probability distributions, such as normal, exponential, and Zipf. Each random variable is associated with a different stream of random numbers.

The publications P3 and P4 repeat a basic simulation for several times. In each subsimulation random variables are using statistically independent sequences (substreams) of pseudo-random numbers, which is stated in these two papers.

5.2.2 Type of Simulations

All simulations used in this dissertation are terminating simulations with a finite time horizon. The duration of a simulation is thus predetermined by the total simulation time which is clearly stated in all simulation-based publications.

5.2.3 Simulation Output Data Analysis

Stochastic computer simulations involving random processes have to be regarded as simulated statistical experiments, and it is mandatory to apply statistical analysis methods to simulation output data [163]. The random nature of this data should not be ignored.

To reduce the amount of statistical error in the results, it is necessary to have a reasonably large number of simulated events. While this is not directly related to the total simulation time, a longer time will increase the number of events. To get a representative set of simulation data, attention must be paid to rare events occurring very seldom. If a simulation is too short, results can be misleading as the set of simulation data can miss events occurring rarely.

Assessment of statistical errors in the final simulation results is necessary for achieving credibility. In preliminary studies, however, it can be acceptable to reduce randomness in simulation output data by simply repeating the simulation and taking an average [163]. This is exactly what has been done in all of the simulations in this dissertation.

A reasonably long simulation time is required for getting a sufficiently large sample of simulation data. The length of a simulation is always a trade off between a reasonable running time and accuracy of results. If the number of events in a simulation is too small, an increased simulation time would result in a clearly different result, for example, as a smoother delay curve.

A test was carried out to check whether multiplying the number of simulated events approximately by ten will have any effect on the average results. The figure 5.1 is the same as the figure 2.16 and uses a simulation time of 60 000 seconds. This exactly same simulation is repeated with a simulation time of 600 000 seconds, which was implemented as 150 different subsimulations each with statistically independent sequences of pseudo-random numbers. The length of one subsimulation was 4 000 seconds. These results are shown in the figure 5.2. When these two figures are

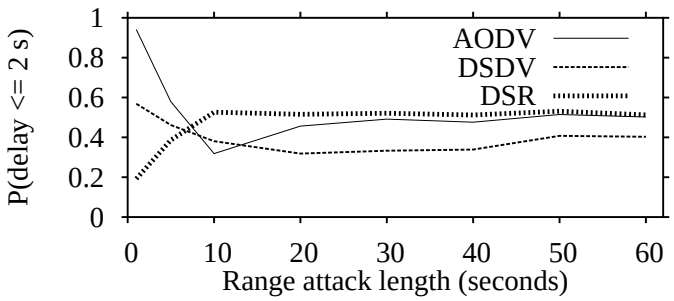

Figure 5.1: Fraction of transmissions having a delay $\leq$ 2 s for the node 4 during an attenuating range attack in an ad hoc network. Simulation time is 60 000 seconds.

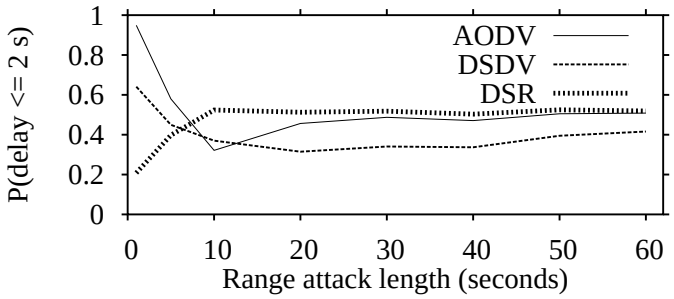

Figure 5.2: Fraction of transmissions having a delay $\leq$ 2 s for the node 4 during an attenuating range attack in an ad hoc network. Simulation time is 600 000 seconds.

compared, there are only minor variations, and the relative difference in performance of ad-hoc routing protocols is clearly visible in both cases.

5.2.4 Statistical Errors

Statistical error of the simulation result is typically measured by indicating both the confidence interval and the confidence level. The final result is expected to be within the indicated confidence interval with the probability indicated by the confidence level. The statistical error was not evaluated in any of the simulations here.

Simulation results are always approximations of the real behavior, especially when malicious attacks are in question. Any attacker is free to change his/her attack strategies.

This dissertation has studied the relative effectiveness when a specific defense mechanism is either used or not used. The simulation times were selected to be relatively long, so that the number of resulting samples would be large enough. Averaging over all simulation results will be more accurate, when more events are simulated.

5.3 Credibility of Mathematical Modeling

Modeling network attacks mathematically, such as using game theory, is not straightforward. Some reasons for this include the following: it is not easy to describe attacker behavior exactly, attackers have a large set of attack mechanisms available for them, attackers can change their attack strategies whenever they want, and risks related to network attacks can change suddenly when new vulnerabilities are found. In other words, attackers do not have to play according to the rules, as they only need a

single vulnerability for carrying out a successful attack, and they can use their imagination in finding new ways to exploit current computer systems. This is clearly seen in the arms race between attackers and defenders. Even though it is difficult (or impossible) to model vastly varying attacks mathematically, modeling will nevertheless help understanding specific attack cases, and this will bring at least some proof for initial assumptions.

Game theory, as used in this dissertation, should be seen as a tool for making it easier to understand the suitability of different defense strategies. A widely known example is the application of game theory in arms race, sometimes also called as the prisoners' dilemma [78]. Any theory or modeling method, that provides some new understanding for the problem of network attacks, is needed.

5.4 Suggestions for Future Work

This section lists some suggestions for future research. Some of the suggested issues are related to providing further evidence on the correctness of the results presented in this dissertation.

- *Analytical solutions.* The simulation results have not been verified by analytical solutions in the dissertation. Analytical approach would be possible at least in the case of DNS lookup failures. This is an item for further work.

- *Comparison of different attack mechanisms.* This dissertation described the range attack as a new attack mechanism against ad hoc networks. In the dissertation it was not investigated whether this new attack would be the worst possible attack mechanism in any realistic situation. It would be interesting to compare the effectiveness of the range attack with other attack mechanisms, such as destroying a wireless node or blocking an antenna permanently. The range attack is more difficult to detect than the loss of a node, but a lost node may force an ad hoc network to be partitioned permanently, depending on the mobility and topology of the network. An enemy's decision on choosing an attack mechanism depends, for example, on how much one can degrade the network performance, how much effort is needed to carry out an attack, and how long an attack can be continued without being detected and mitigated.

- *Effect of different network setups on simulation results.* Relatively simple network setups were used especially for studying the range attack. It would be interesting to study whether simulation results change when, for example, network topology, number of nodes in the network, transmission scheme, or mobility scheme changes. The simulation results in the small ad hoc network should be verified in a random network of much larger number of nodes.

- *Suitability of rate limiting on different kinds of traffic types.* The publication P6 concentrated on studying file transmissions. HTTP traffic was studied only shortly. Different traffic types, however, may react differently to rate limiting. This would be worth studying in more detail.

- *Recommendations about defense mechanisms.* This dissertation has described shortly many existing defense mechanisms and defined some new ones. For a network user (for example, an organization providing services in the Internet) it would be useful to have some recommended sets of defense mechanisms against the most prevalent attack types.

- *Effect of the DNS request retransmission scheme on tolerance against DoS attacks.* Different name server implementations (such as djbdns and BIND) use slightly different retransmission scheme for DNS requests. This retransmission scheme can have an effect on how well the domain name system tolerates flooding DoS attacks against name servers. It would be worth studying what kind of scheme would provide the best response times under various conditions, and what kind of possible side effects these schemes might have (such as increased network load).

6 CONCLUSIONS

Mitigation of DoS attacks is a part of an overall risk management strategy for an organization. Each organization must identify the most important DoS risks, and implement a cost-effective set of defense mechanisms against those attack types causing the highest risk for business continuity. Studies and news about real-life DoS attacks indicate that these attacks are not only among the most prevalent network security risks, but that these attacks can also block whole organizations out of the Internet for the duration of an attack. The risk from DoS attacks should not thus be underestimated, but not overestimated, either.

In the future the problem from DoS attacks will most probably increase because the number of hosts connected in the Internet increases, access lines get faster, software products get more complex, and security continues to be difficult for an ordinary home user and even many organizations. The more there are hosts in the Internet, the more of them can potentially be used for DoS purposes. The intensity of DoS attacks can also increase, as a higher number of hosts can produce more traffic over faster Internet access lines. As software gets more complex, more vulnerabilities will reside in them to be used for compromising hosts. The fast pace of new revisions does not make the situation easier. Finally, it will continue to be difficult to evaluate security risks in existing computer systems, especially by ordinary people.

This dissertation has studied the problem of DoS attack mitigation from several points of view. Individual defense mechanisms were described and analyzed, selection of defenses was studied, and comprehensive taxonomy of criteria for evaluating defense mechanisms was given. The main contributions of each publication are the following.

- *The publication P1* gives an extensive, structured description of both existing DoS attack mechanisms and available defense mechanisms. A comprehensive set of defenses is needed to get defense in depth, both during the deployment and the attack phases. The actual selection of defense mechanisms should be based on a risk management process. Important factors, that need to be considered during the selection process, are listed.

- *The publication P2* describes a new defense mechanism for protecting name servers from flooding DoS attacks. The simulation results show that the presented mechanism decreases clearly both the average DNS lookup delay and the number of failed DNS lookups.

- *The publication P3* describes a new DoS attack, called the amplifying range attack, against mobile ad hoc networks. This publication also presents two cross-layer designs for mitigating this DoS attack. The simulation results show that it is possible to even double the number of transmissions fulfilling a specific round-trip delay requirement in an ad hoc network by using simple cross-layer designs.

- *The publication P4* presents a new DoS attack, called the attenuating range attack, against mobile ad hoc networks. This publication simulated the resilience of three ad-hoc routing protocols against this attack. When applications require a very short delay, DSDV was found to provide the best performance. When applications tolerate a longer delay, AODV was found to provide the best performance. The resilience of ad-hoc routing protocols against the range attack was thus situation dependent.

- *The publication P5* describes a taxonomy of criteria for evaluating defense mechanisms against flooding DoS attacks. This list brings out important issues to be considered when evaluating the real usefulness of different defense mechanisms.

- *The publication P6* studies the effectiveness of rate limiting when the QoS of all legitimate traffic is important. Both simulations and empirical measurements in a test network showed that rate limiting is most suitable for protecting servers from which data is mainly downloaded, such as typical web-servers. In this kind of a case up to 50% of incoming attack packets to a server can be discarded without any clear problems for any legitimate flows.

- *The publication P7* applies game theory for modeling four example situations in information warfare. One of these games analyzes an attack where an attacker tries to compromise a host to be later used for DoS attack purposes. The solution to this example game showed that using different strategies randomly one at a time results in best protection against the studied attack type. A game theoretic approach for perception management was also described shortly, and the goal here is to cause a victim to make wrong decisions in his OODA-loop due to errors in perceiving the attacker's payoffs.

REFERENCES

[1] I. Aad, J.-P. Hubaux, and E. W. Knightly, "Denial of service resilience in ad hoc networks," in *Proceedings of the ACM MobiCom*, Philadelphia, USA, Sept. 2004, pp. 202–215.

[2] L. A. Adamic, "Zipf, Power-laws, and Pareto - a ranking tutorial," Xerox Palo Alto Research Center, Tech. Rep., 2000. [Online]. Available: http://ginger.hpl.hp.com/research/idl/papers/ranking/ranking.html, [accessed Jan. 29, 2006].

[3] D. Adkins, K. Lakshminarayanan, A. Perrig, and I. Stoica, "Towards a more functional and secure network infrastructure," University of California, Berkeley, Tech. Rep. UCB/CSD-03-1242, 2003.

[4] A. K. Agarwal and W. Wang, "An experimental study of cross-layer security protocols in public access wireless networks," in *Proceedings of the IEEE GLOBECOM*, St. Louis, USA, Nov. 2005.

[5] D. Ahmad, "The rising threat of vulnerabilities due to integer errors," *IEEE Security & Privacy*, vol. 1, no. 4, pp. 77–82, July/Aug. 2003.

[6] Akamai, "Press release: Akamai provides insight into Internet denial of service attack," June 2004. [Online]. Available: http://www.akamai.com/en/html/about/press/press459.html, [accessed Jan. 7, 2006].

[7] P. Albitz and C. Liu, *DNS and BIND*, 4th ed. Sebastopol, USA: O'Reilly & Associates, Inc., Apr. 2001.

[8] Aleph One, "Smashing the stack for fun and profit," *Phrack*, vol. 7, no. 49, Nov. 1996.

[9] J. Allen, A. Christie, W. Fithen, J. McHugh, J. Pickel, and E. Stoner, "State of the practice of intrusion detection technologies," Carnegie Mellon University, Software Engineering Institute, Tech. Rep. CMU/SEI-99-TR-028, Jan. 2000.

[10] T. Alpcan and T. Basar, "A game theoretic approach to decision and analysis in network intrusion detection," in *Proceedings of the IEEE Conference on Decision and Control*, Maui, Hawaii, USA, Dec. 2003, pp. 2595–2600.

[11] American National Standards Institute, Inc., *American National Standard T1.523-2001, Telecom Glossary 2000*, Alliance for Telecommunications Industry Solutions, Feb. 2001.

[12] M. Andrews and J. A. Whittaker, "Computer security," *IEEE Security & Privacy*, vol. 2, no. 5, pp. 68–71, Sept./Oct. 2004.

[13] Anonymous, *Webster's Encyclopedic Unabridged Dictionary of the English Language*. New York, USA: Gramercy Books, 1989.

[14] X. Ao, "Report on DIMACS workshop on large-scale Internet attacks," Rutgers University, Tech. Rep., Sept. 2003.

[15] I. Arce and E. Levy, "An analysis of the Slapper worm," *IEEE Security & Privacy*, vol. 1, no. 1, pp. 82–87, Jan./Feb. 2003.

[16] T. Aura, P. Nikander, and J. Leiwo, "DOS-resistant authentication with client puzzles," in *Proceedings of the 8th International Workshop on Security Protocols*, Cambridge, UK, Apr. 2000.

[17] T. Aura and M. Roe, "Designing the mobile IPv6 security protocol," Microsoft Research, Tech. Rep. MSR-TR-2006-42, Apr. 2006.

[18] AusCERT, "2005 Australian computer crime and security survey," Australian Computer Emergency Response Team, Tech. Rep., 2005. [Online]. Available: http://www.auscert.org.au/crimesurvey [accessed Jan. 4, 2006].

[19] A. Avižienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11–33, Jan.–Mar. 2004.

[20] A. Barbir, S. Murphy, and Y. Yang, "Generic threats to routing protocols," Internet Society, Internet Draft draft-ietf-rpsec-routing-threats-07, work in progress, Oct. 2004.

[21] S. Beattie, S. Arnold, C. Cowan, P. Wagle, C. Wright, and A. Shostack, "Timing the application of security patches for optimal uptime," in *Proceedings of the 16th Systems Administration Conference*, Philadelphia, USA, Nov. 2002.

[22] A. Belenky and N. Ansari, "On IP traceback," *IEEE Commun. Mag.*, vol. 41, no. 7, pp. 142–153, July 2003.

[23] S. Bellovin, M. Leech, and T. Taylor, "ICMP traceback messages," Internet Society, Internet Draft draft-ietf-itrace-04.txt, work in progress, Feb. 2003.

[24] S. M. Bellovin, "The state of software security," Nov. 2002. [Online]. Available: http://www.research.att.com/~smb/talks/vuln-legal.pdf, [accessed Jan. 28, 2006].

[25] M. Bialoglowy, "Bluetooth security review, part 2," Security Focus, Tech. Rep., 2005.

[26] M. Bishop, "Early computer security papers, part 1," 1998. [Online]. Available: http://csrc.nist.gov/publications/history/index.html [accessed Jan. 3, 2006].

[27] M. Bishop, *Computer Security: Art and Science*. Boston, USA: Pearson Education, 2003.

[28] M. Bishop, "What is computer security," *IEEE Security & Privacy*, vol. 1, no. 1, pp. 67–69, Jan/Feb 2003.

[29] P. Bouchareine, "Format string vulnerability," Hacker Emergency Response Team, Tech. Rep., July 2000.

[30] J. R. Boyd, "The essence of winning and losing," Jan. 1996, presentation slides.

[31] K. A. Bradley, S. Cheung, N. Puketza, B. Mukherjee, and R. A. Olsson, "Detecting disruptive routers: A distributed network monitoring approach," *IEEE Network*, vol. 12, no. 5, pp. 50–60, Sept./Oct. 1998.

[32] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proceedings of the IEEE INFOCOM*, New York, USA, Mar. 1999.

[33] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the Fourth Annual ACM/IEEE International Cnference on Mobile Computing and Networking*, Dallas, USA, Oct. 1998, pp. 85–97.

[34] N. Brownlee, K. C. Claffy, and E. Nemeth, "DNS measurements at a root server," in *Proceedings of the IEEE GlobeCom*, San Antonio, USA, Nov. 2001.

[35] N. Brownlee and I. Ziedins, "Response time distributions for global name servers," in *Proceedings of the Passive & Active Measurement Workshop*, Fort Collins, USA, Mar. 2002.

[36] Bulba and Kil3r, "Bypassing StackGuard and StackShield," *Phrack*, vol. 10, no. 56, Jan. 2000.

[37] R. Bush, D. Karrenberg, M. Kosters, and R. Plzak, "Root name server operational requirements," Internet Engineering Task Force, Request for Comments RFC 2870, June 2000.

[38] S. Byers, A. D. Rubin, and D. Kormann, "Defending against an Internet-based attack on the physical world," in *Proceedings of the Workshop on Privacy in the Electronic Society*, Washington, DC, USA, Nov. 2002.

[39] CAIDA, "Nameserver DoS attack October 2002," 2002. [Online]. Available: http://www.caida.org/projects/dns-analysis/oct02dos.xml, [accessed Jan. 7, 2006].

[40] Center for Infrastructure Expertise, *Critical Infrastructure Glossary*, National Infrastructure Institute. [Online]. Available: http://www.ni2ciel.org/Glossary/, [accessed Jan. 29, 2006].

[41] CERT Coordination Center, "Denial of service attacks," Oct. 1997. [Online]. Available: http://www.cert.org/tech_tips/denial_of_service.html, [accessed Jan. 5, 2006].

[42] CERT Coordination Center, "Results of the Distributed-Systems Intruder Tools Workshop," Nov. 1999.

[43] CERT Coordination Center, "CERT incident note IN-2000-04, denial of service attacks using nameservers," Apr. 2000.

[44] CERT Coordination Center, "Overview of attack trends," Feb. 2002. [Online]. Available: http://www.cert.org/archive/pdf/attack_trends.pdf, [accessed Jan. 5, 2006].

[45] CERT Coordination Center, "CERT Advisory CA-2003-17: Exploit available for the Cisco IOS Interface Blocked Vulnerabilities," July 2003.

[46] A. Chakrabarti and G. Manimaran, "Internet infrastructure security: A taxonomy," *IEEE Network*, vol. 16, no. 6, pp. 13–21, 2002.

[47] M.-K. Chan and M. Hamdi, "An active queue management scheme based on a capture-recapture model," *IEEE J. Select. Areas Commun.*, vol. 21, no. 4, pp. 572–583, May 2003.

[48] R. K. Chang, "Defending against flooding-based distributed denial-of-service attacks: A tutorial," *IEEE Commun. Mag.*, vol. 40, no. 10, pp. 42–51, Oct. 2002.

[49] Cisco Systems, Inc., "Configuring TCP intercept (preventing denial-of-service attacks)," 1997.

[50] Cisco Systems, Inc., "Characterizing and tracing packet floods using cisco routers," May 2005.

[51] Cisco Systems, Inc., "Strategies to protect against distributed denial of service (DDos) attacks," Jan. 2006.

[52] M. Conti, G. Maselli, G. Turi, and S. Giordano, "Cross-layering in mobile ad hoc network design," *IEEE Computer*, vol. 37, no. 2, pp. 48–51, Feb. 2004.

[53] C. Cowan, "Software security for open-source systems," *IEEE Security & Privacy*, vol. 1, no. 1, pp. 38–45, Jan./Feb. 2003.

[54] C. Cowan, M. Barringer, S. Beattie, and G. Kroah-Hartman, "FormatGuard: Automatic protection from printf format string vulnerabilities," in *Proceedings of the 10th USENIX Security Symposium*, Washington, D.C., Aug. 2001.

[55] C. Cowan, S. Beattie, R. F. Day, C. Pu, P. Wagle, and E. Walthinsen, "Protecting systems from stack smashing attacks with StackGuard," in *Proceedings of the LinuxExpo*, Raleigh, USA, May 1999.

[56] C. Cowan, C. Pu, D. Maier, J. Walpole, P. Bakke, S. Beattie, A. Grier, P. Wagle, Q. Zhang, and H. Hinton, "StackGuard: Automatic adaptive detection and prevention of buffer-overflow attacks," in *Proceedings of the 7th USENIX Security Conference*, San Antonio, Texas, Jan. 1998, pp. 63–78.

[57] L. F. Cranor and S. Garfinkel, "Secure or usable," *IEEE Security & Privacy*, vol. 2, no. 5, pp. 16–18, Sept./Oct. 2004.

[58] P. J. Criscuolo, "Distributed denial of service," Department of Energy and Lawrence Livermore National Laboratory, Tech. Rep. CIAC-2319, Feb. 2000.

[59] S. A. Crosby and D. S. Wallach, "Denial of service via algorithmic complexity attacks," in *Proceedings of the 12th USENIX Security Symposium*, Washington, D.C., USA, Aug. 2003.

[60] G. Cybenko, A. Giani, and P. Thompson, "Cognitive hacking: A battle for the mind," *IEEE Computer*, vol. 35, no. 8, pp. 50–56, Aug. 2002.

[61] M. Dahlin, B. B. V. Chandra, L. Gao, and A. Nayate, "End-to-end WAN service availability," *IEEE/ACM Transactions on Networking*, vol. 11, no. 2, pp. 300–313, Apr. 2003.

[62] E. Desautels, "Software license agreements: Ignore at your own risk," US-CERT, Tech. Rep., 2005.

[63] S. Desilva and R. V. Boppana, "Mitigating malicious control packet floods in ad hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, New Orleans, USA, Mar. 2005.

[64] R. Durst, T. Champion, B. Witten, E. Miller, and L. Spagnuolo, "Testing and evaluating computer intrusion detection systems," *Communications of the ACM*, vol. 42, no. 7, pp. 53–61, July 1999.

[65] G. F. Elmasry, C. J. McCann, and R. Welsh, "Partitioning QoS management for secure tactical wireless ad hoc networks," *IEEE Communications Magazine*, vol. 43, no. 12, Nov. 2005.

[66] S. Evans, D. Heinbuch, E. Kyle, J. Piorkowski, and J. Wallner, "Risk-based systems security engineering: Stopping attacks with intention," *IEEE Security & Privacy*, vol. 2, no. 6, pp. 59–62, Nov./Dec. 2004.

[67] P. Ferguson and D. Senie, "Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing," The Internet Society, Request for Comments RFC 2827, May 2000.

[68] S. Floyd, S. Bellovin, J. Ioannidis, K. Kompella, R. Mahajan, and V. Paxson, "Pushback messages for controlling aggregates in the network," Internet Engineering Task Force, Internet Draft draft-floyd-pushback-messages-00.txt, work in progress, July 2001.

[69] S. Floyd and E. Kohler, "Internet research need better models," in *Proceedings of the First Workshop on Hot Topics in Networks*, Princeton, New Jersey, USA, Oct. 2002.

[70] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A sense of self for Unix processes," in *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, California, May 1996, pp. 120–128.

[71] F. C. Freiling, T. Holz, and G. Wicherski, "Botnet tracking: Exploring a root-cause methodology to prevent distributed denial-of-service attacks," RWTH Aachen, Dept. of Computer Science, Tech. Rep. AIB-2005-07, Apr. 2005.

[72] M. Fullmer and S. Romig, "The OSU flow-tools package and Cisco NetFlow logs," in *Proceedings of the 14th Systems Administration Conference*, New Orleans, Louisiana, Dec. 2000.

[73] P. Galli, "DoS attack brings down Sun Grid demo," Mar. 2006. [Online]. Available: http://www.eweek.com/article2/0,1895,1941574,00.asp, [accessed Mar. 10, 2006].

[74] L. Garber, "Denial-of-service attacks rip the Internet," *IEEE Computer*, vol. 33, no. 4, pp. 12–17, Apr. 2000.

[75] A. Garg and A. L. N. Reddy, "Mitigation of DoS attacks through QoS regulation," in *Proceedings of IEEE International Workshop on Quality of Service*, Miami Beach, Florida, USA, May 2002.

[76] D. Geer, "Risk management is still where the money is," *IEEE Computer*, vol. 36, no. 12, pp. 129–131, Dec. 2003.

[77] A. Ghosh, "Sizing the opportunity for opportunistic cybercriminals," *Journal of Information Warfare*, vol. 1, no. 2, pp. 80–89, 2002.

[78] R. Gibbons, *A Primer in Game Theory*. Essex, UK: Pearson Education, 1992.

[79] D. Gollmann, *Computer Security*. Chichester, England: John Wiley & Sons, 1999.

[80] C. Gonsalves, "Akamai DDoS attack whacks Web traffic," June 2004. [Online]. Available: http://www.eweek.com/article2/0,1895,1612739,00.asp, [accessed Jan. 7, 2006].

[81] L. A. Gordon and M. P. Loeb, *Managing cyber-security resources: A cost-benefit analysis*. McGraw-Hill, 2006.

[82] L. A. Gordon, M. P. Loeb, W. Lucyshyn, and R. Richardson, "2005 CSI/FBI computer crime and security survey," Computer Security Institute, Tech. Rep., 2005.

[83] L. A. Gordon, M. P. Loeb, and T. Sohail, "A framework for using insurance for cyber-risk management," *Communications of the ACM*, vol. 46, no. 3, Mar. 2003.

[84] J. Gozdecki, A. Jajszczyk, and R. Stankiewicz, "Quality of service terminology in IP networks," *IEEE Commun. Mag.*, vol. 41, no. 3, pp. 153–159, Mar. 2003.

[85] B. Guha and B. Mukherjee, "Network security via reverse engineering of TCP code: Vulnerability analysis and proposed solutions," *IEEE Network*, vol. 11, no. 4, pp. 40–48, July/Aug. 1997.

[86] A. Hackworth, "Spyware," US-CERT, Tech. Rep., 2005.

[87] J. Haines, D. K. Ryder, L. Tinnel, and S. Taylor, "Validation of sensor alert correlators," *IEEE Security & Privacy*, vol. 1, no. 1, pp. 46–56, Jan./Feb. 2003.

[88] M. Handley, V. Paxson, and C. Kreibich, "Network intrusion detection: Evasion, traffic normalization, and end-to-end protocol semantics," in *Proceedings of the 10th USENIX Security Symposium*, 2001.

[89] W. Hardaker, D. Kindred, R. Ostrenga, D. Sterne, and R. Thomas, "Justification and requirements for a national DDoS defense technology evaluation facility," Network Associates Laboratories, Tech. Rep., July 2002.

[90] Haymarket Media, "Al-Jazeera hacked in DoS attack," Jan. 2005. [Online]. Available: http://www.itnews.com.au/newsstory.aspx?CIaNID=17603, [accessed Mar. 10, 2006].

[91] J. Heiser, "A risky business," *Telecommunications International*, pp. 33–35, Nov. 2003.

[92] T. Holz and F. Raynal, "Detecting honepots and other suspicious environments," in *Proceedings of the IEEE Workshop on Information Assurance and Security*, West Point, New York, USA, June 2005.

[93] K. J. Houle, G. M. Weaver, N. Long, and R. Thomas, "Trends in denial of service attack technology," CERT Coordination Center, Tech. Rep., Oct. 2001.

[94] A. Householder, A. Manion, L. Pesante, G. M. Weaver, and R. Thomas, "Managing the threat of denial-of-service attacks," CERT Coordination Center, Tech. Rep., Oct. 2001.

[95] J. D. Howard, "An analysis of security incidents on the Internet 1989–1995," Ph.D. dissertation, Carnegie Mellon University, Apr. 1997.

[96] J. D. Howard and T. A. Longstaff, "A common language for computer security incidents," Sandia National Laboratories, Tech. Rep. SAND98-8667, Oct. 1998.

[97] J. Hu, "'Zombie' PCs caused Web outage, Akamai says," June 2004. [Online]. Available: http://news.zdnet.com/2100-1009_22-5236403.html, [accessed Jan. 7, 2006].

[98] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Rushing attacks and defense in wireless ad hoc network routing protocols," in *Proceedings of the ACM Workshop on Wireless Security*, San Diego, USA, Sept. 2003, pp. 30–40.

[99] A. Huhtinen and J. Rantapelkonen, "Perception management in the art of war. a review of Finnish war propaganda and present-day information warfare," *Journal of Information Warfare*, vol. 2, no. 1, pp. 50–58, 2002.

[100] C. Huitema, *Routing in the Internet*, 2nd ed. Upper Saddle River, USA: Prentice Hall PTR, 2000.

[101] A. Hussain, J. Heidemann, and C. Papadopoulos, "A framework for classifying denial of service attacks," in *Proceedings of ACM SIGCOMM*, Karlsruhe, Germany, Aug. 2003.

[102] W. Hutchinson and M. Warren, "Principles of information warfare," *Jounal of Information Warfare*, vol. 1, no. 1, pp. 1–6, 2001.

[103] N. Ianelli and A. Hackworth, "Botnets as a vehicle for online crime," CERT Coordination Center, Tech. Rep., Dec. 2005.

[104] International Organization for Standardization, *ISO 7498-2:1989, Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture*, Geneva, Switzerland, 1989.

[105] International Organization for Standardization, *ISO/IEC 17799:2005, Code of Practice for Information Security Management*, 2005.

[106] Internet System Consortium, "Internet domain survey," July 2005. [Online]. Available: http://www.isc.org/index.pl?/ops/ds/, [accessed Jan. 9, 2006].

[107] ISP BGP & DNS Working Group, "Enhancing the security of name resolution and inter-domain Internet routing," May 2002.

[108] ITworld.com, "CERT hit by DDoS attack for a third day," May 2001. [Online]. Available: http://security.itworld.com/4339/IDG010524CERT2/pfindex.html, [accessed Jan. 25, 2006].

[109] A. P. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri, "Real-world environment models for mobile network evaluation," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 622–632, Mar. 2005.

[110] D. B. Johnson, D. A. Maltz, and Y.-C. Hu, "The dynamic source routing protocol for mobile ad hoc networks (DSR)," Internet Society, Internet Draft draft-ietf-manet-dsr-10.txt, work in progress, July 2004.

[111] K. L. Johnson, J. F. Carr, M. S. Day, and M. F. Kaashoek, "The measured performance of content distribution networks," in *Proceedings of the 5th International Web Caching and Content Delivery Workshop*, Lisbon, Portugal, May 2000.

[112] A. Juels and J. Brainard, "Client puzzles: A cryptographic countermeasure against connection depletion attacks," in *Proceedings of the Network and Distributed System Security Symposium*, San Diego, California, USA, Feb. 1999.

[113] J. Jung, E. Sit, H. Balakrishnan, and R. Morris, "DNS performance and the effectiveness of caching," *IEEE/ACM Trans. Networking*, vol. 10, no. 5, pp. 589–603, Oct. 2002.

[114] V. Kawadia and P. R. Kumar, "A cautionary perspective on cross-layer design," *IEEE Wireless Communications*, vol. 12, no. 1, pp. 3–11, Feb. 2005.

[115] V. Kawadia and P. R. Kumar, "Principles and protocols for power control in wireless ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 76–88, Jan. 2005.

[116] A. D. Keromytis, V. Misra, and D. Rubenstein, "SOS: An architecture for mitigating DDoS attacks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 176–188, Jan. 2004.

[117] T. Killalea, "Recommended Internet service provider security services and procedures," The Internet Society, Request for Comments RFC 3013, Nov. 2000.

[118] S. Krasser, J. B. Grizzard, H. L. Owen, and J. G. Levine, "The use of honeynets to increase computer network security and user awareness," *Journal of Security Education*, vol. 1, no. 2/3, pp. 23–37, 2005.

[119] N. Krawetz, "Anti-honeypot technology," *IEEE Security & Privacy*, vol. 2, no. 1, pp. 76–79, Jan./Feb. 2004.

[120] B. Krishnamurthy, C. Wills, and Y. Zhang, "On the use and performance of content distribution networks," AT&T Labs - Research, Tech. Rep. TD-52AMHL, Aug. 2001.

[121] A. Kumar, J. Postel, C. Neuman, P. Danzig, and S. Miller, "Common DNS implementation errors and suggested fixes," Internet Engineering Task Force, Request for Comments RFC 1536, Oct. 1993.

[122] K. Lakshminarayanan, D. Adkins, A. Perrig, and I. Stoica, "Taming IP packet flooding attacks," in *Proceedings of the 2nd ACM Workshop on Hot Topics in Networks*, Cambridge, Massachusetts, USA, Nov. 2003.

[123] Lawrence Livermore National Laboratory and Sandia National Laboratories, "Intrusion detection and response," Dec. 1996. [Online]. Available: http://all.net/journal/ntb/ids.html, [accessed Jan. 29, 2006].

[124] P. L'Ecuyer, "Good parameters and implementations for combined multiple recursive random number generators," *Operations Research*, vol. 47, no. 1, pp. 159–164, 1999.

[125] P. L'Ecuyer, R. Simard, E. J. Chen, and W. D. Kelton, "An object-oriented random number package with many long streams and substreams," *Operations Research*, vol. 50, no. 6, pp. 1073–1075, 2002.

[126] W. Lee, W. Fan, M. Miller, S. J. Stolfo, and E. Zadok, "Toward cost-sensitive modeling for intrusion detection and response," *Journal of Computer Security*, vol. 10, no. 1–2, 2002.

[127] E. Levy, "Crossover: Online pests plaguing the offline world," *IEEE Security & Privacy*, vol. 1, no. 6, Nov./Dec. 2003.

[128] E. Levy, "Criminals become tech savvy," *IEEE Security & Privacy*, vol. 2, pp. 65–68, Mar./Apr. 2004.

[129] J. Leyden, "Scottish ISP floored as DDoS attacks escalate," Apr. 2002. [Online]. Available: http://www.theregister.co.uk/2002/04/09/scottish_isp_floored_as_ddos/print.html, [accessed Jan. 25, 2006].

[130] M. Libicki, “What is information warfare?” National Defense University, Tech. Rep., Aug. 1995.

[131] R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, and M. A. Zissman, “Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation,” in *Proceedings of the DARPA Information Survivability Conference and Exposition*, 2000.

[132] P. Liu and W. Zang, “Incentive-based modeling and inference of attacker intent, objectives, and strategies,” in *Proceedings of the ACM Conference on Computer and Communications Security*, Washington, DC, USA, Oct. 2003.

[133] K.-W. Lye and J. M. Wing, “Game strategies in network security,” in *Proceedings of the Workshop on Foundations of Computer Security*, Copenhagen, Denmark, July 2002.

[134] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker, “Controlling high bandwidth aggregates in the network,” *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 3, pp. 62–73, July 2002.

[135] C. Manikopoulos and S. Papavassiliou, “Network intrusion and fault detection: A statistical anomaly approach,” *IEEE Commun. Mag.*, vol. 40, no. 10, pp. 76–82, Oct. 2002.

[136] M. Mathis, J. Semke, and J. Mahdavi, “The macroscopic behavior of the TCP congestion avoidance algorithm,” *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 3, 1997.

[137] S. McCreary and K. C. Claffy, “Trends in wide area IP traffic patterns - a view from Ames Internet Exchange,” in *Proceedings of the ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, Monterey, CA, Sept. 2000.

[138] A. McCue, “Bookie reveals $100,000 cost of denial-of-service extortion attacks,” June 2004. [Online]. Available: http://software.silicon.com/security/0,39024655,39121278,00.htm, [accessed Jan. 6, 2006].

[139] J. McHugh, “Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection system Evaluations as Performed by Lincoln Laboratory,” *ACM Transactions on Information and System Security*, vol. 3, no. 4, pp. 262–294, Nov. 2000.

[140] C. Meadows, “A cost-based framework for analysis of denial of service in networks,” *Journal of Computer Security*, vol. 9, no. 1–2, pp. 143–164, 2001.

[141] Men&Mice, “Domain health survey for .COM,” Feb. 2003. [Online]. Available: http://www.menandmice.com/6000/61_recent_survey.html, [accessed Jan. 8, 2006].

[142] MessageLabs, “2005 annual security report,” MessageLabs Ltd., Tech. Rep., 2006.

[143] P. Michiardi and R. Molva, “A game theoretical approach to evaluate cooperation enforcement mechanisms in mobile ad hoc networks,” in *Proceedings of the WiOpt 03: Modeling and Optimization in Mobile, Ad Hoc and Wireless networks*, Sophia-Antipolis, France, Mar. 2003.

[144] J. Milletary, "Technical trends in phishing attacks," US-CERT, Tech. Rep., 2005.

[145] J. Mirkovic, S. Dietrich, D. Dittrich, and P. Reiher, *Internet Denial of Service: Attack and Defense Mechanisms*. Upper Saddle River, USA: Pearson Education, Inc., 2005.

[146] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, Apr. 2004.

[147] J. Mirkovic, M. Robinson, P. Reiher, and G. Kuenning, "Alliance formation for DDoS defense," in *Proceedings of the Workshop on New Security Paradigms*, Ascona, Switzerland, Aug. 2003.

[148] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "Inside the Slammer worm," *IEEE Security & Privacy*, vol. 1, no. 4, pp. 33–39, July/Aug. 2003.

[149] D. Moore, C. Shannon, and J. Brown, "Code-Red: a case study on the spread and victims of an Internet worm," in *Proceedings of the Internet Measurement Workshop*, Marseille, France, Nov. 2002.

[150] D. Moore, C. Shannon, G. M. Voelker, and S. Savage, "Internet quarantine: Requirements for containing self-propagating code," in *Proceedings of the IEEE INFOCOM*, Mar. 2003.

[151] D. Moore, G. M. Voelker, and S. Savage, "Inferring Internet denial-of-service activity," in *Proceedings of the 10th USENIX Security Symposium*, Washington, D.C., Aug. 2001.

[152] W. G. Morein, A. Stavrou, D. L. Cook, A. D. Keromytis, V. Misra, and D. Rubenstein, "Using graphic Turing tests to counter automated DDoS attacks against web servers," in *Proceedings of the ACM conference on computer and communications security*, Washington, DC, USA, Oct. 2003, pp. 8–19.

[153] P. Mueller and G. Shipley, "Dragon claws its way to the top," *Network Computing*, pp. 45–67, Aug. 2001.

[154] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network intrusion detection," *IEEE Network*, vol. 8, no. 3, pp. 26–41, May/June 1994.

[155] S. Mukkamala and A. H. Sung, "A framework for countering denial of service attacks," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, The Hague, Netherlands, Oct. 2004.

[156] J. A. Mulligan, "A comparison framework for proactive Internet denial of service solutions," Massachusetts Institute of Technology, Tech. Rep., Mar. 2005.

[157] S. Northcutt and J. Novak, *Network Intrusion Detection*, 3rd ed. Indianapolis, Indiana: New Riders Publishing, Sept. 2002.

[158] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *Proceedings of the ACM SIGCOMM conference*, Vancouver, Canada, Sept. 1998.

[159] V. N. Padmanabhan and D. R. Simon, "Secure Traceroute to detect faulty or malicious routing," *ACM SIGCOMM Computer Communications Review*, vol. 33, no. 1, pp. 77–82, Jan. 2003.

[160] P. Papadimitratos and Z. J. Haas, "Securing the Internet routing infrastructure," *IEEE Commun. Mag.*, vol. 40, no. 10, pp. 60–68, Oct. 2002.

[161] K. Park and H. Lee, "On the effectiveness of route-baed packet filtering for distributed DoS attack prevention in power-law Internets," in *Proceedings of the ACM SIGCOMM Conference*, San Diego, California, USA, Aug. 2001.

[162] J. Parker, A. Patwardhan, and A. Joshi, "Cross-layer analysis for detecting wireless misbehavior," in *Proceedings of the IEEE Consumer Communications and Networking Conference*, Las Vegas, Nevada, USA, Jan. 2006.

[163] K. Pawlikowski, H.-D. J. Jeong, and J.-S. R. Lee, "On credibility of simulation studies of telecommunication networks," *IEEE Communications Magazine*, vol. 40, no. 1, pp. 132–139, Jan. 2002.

[164] V. Paxson, "Bro: A system for detecting network intruders in real-time," *Computer Networks*, vol. 31, no. 23-24, pp. 2435–2463, Dec. 1999.

[165] V. Paxson, "An analysis of using reflectors for distributed denial-of-service attacks," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 3, July 2001.

[166] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," The Internet Society, Request for Comments RFC 3561, July 2003.

[167] C. E. Perkins, Ed., *Ad Hoc Networking*. Upper Saddle River, USA: Addison-Wesley, 2001.

[168] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proceedings of the SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications*, London, England, UK, Aug. 1994, pp. 234–244.

[169] D. Plonka, "Flawed routers flood University of Wisconsin Internet time server," University of Wisconsin, Tech. Rep., Aug. 2003. [Online]. Available: http://www.cs.wisc.edu/˜plonka/netgear-sntp/, [accessed May 6, 2006].

[170] F. Pouget, M. Dacier, and V. H. Pham, "Leurre.com: on the advantages of deploying a large scale distributed honeypot platform," in *Proceedings of the E-Crime and Computer Evidence Conference*, Monaco, Mar. 2005.

[171] T. H. Ptacek and T. N. Newsham, "Insertion, evasion, and denial of service: Eluding network intrusion detection," Secure Networks, Inc., Tech. Rep., Jan. 1998.

[172] H. Pucha, S. M. Das, and Y. C. Hu, "The performance impact of traffic patterns on routing protocols in mobile ad hoc networks," in *Proceedings of the 7th ACM International Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, Venice, Italy, Oct. 2004, pp. 211–219.

[173] R. F. Puppy, "A look at whisker's anti-IDS tactics: Just how bad can we ruin a good thing?" Dec. 1999. [Online]. Available: http://www.ussrback.com/docs/papers/IDS/whiskerids.html, [accessed Jan. 28, 2006].

[174] S. Radosavac, N. Benammar, and J. S. Baras, "Cross-layer attacks in wireless ad hoc networks," in *Proceedings of the Conference on Information Sciences and Systems*, Princeton, New Jersey, USA, Mar. 2004.

[175] R. Ramanathan, J. Redi, C. Santivanez, D. Wiggins, and S. Polit, "Ad hoc networking with directional antennas: A complete system solution," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 496–506, Mar. 2005.

[176] J. Reason, *Managing the Risks of Organizational Accidents*. Burlington, USA: Ashgate Publishing Company, 1997.

[177] E. Rescorla, "Security holes... who cares?" in *Proceedings of the 12th USENIX Security Conference*, Washington, DC, USA, Aug. 2003.

[178] Reuters, "Scotland Yard and the case of the rent-a-zombies," July 2004. [Online]. Available: http://news.zdnet.com/2100-1009_22-5260154.html, [accessed Jan. 6, 2006].

[179] V. Rezmierski, A. Carroll, and J. Hine, "A study on incident costs and frequencies," *;login:*, vol. 25, no. 5, Aug. 2000.

[180] K. Sallhammar and S. J. Knapskog, "Using game theory in stochastic models for quantifying security," in *Proceedings of the Ninth Nordic Workshop on Secure IT Systems*, Espoo, Finland, Nov. 2004, pp. 39–44.

[181] R. Sandhu, "Good-enough security: Toward a pragmatic business-driven discipline," *IEEE Internet Computing*, vol. 7, no. 1, pp. 66–68, Jan/Feb 2003.

[182] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical network support for IP traceback," in *Proceedings of the ACM SIGCOMM Conference*, Stockholm, Sweden, Aug. 2000, pp. 295–306.

[183] B. Schneier, *Secrets & Lies, Digital Security in a Networked World*. New York, USA: John Wiley & Sons, Inc., 2000.

[184] B. Schneier, *Beyond Fear: Thinking Sensibly about Security in an Uncertain World*. New York, USA: Copernicus Books, 2003.

[185] B. Schneier, "Hacking the business climate for network security," *IEEE Computer*, vol. 37, no. 4, pp. 87–89, Apr. 2004.

[186] B. Schneier, "Attack trends 2004 and 2005," *ACM Queue*, vol. 3, no. 5, pp. 2–3, June 2005.

[187] C. L. Schuba, I. V. Krsul, M. G. Kuhn, E. H. Spafford, A. Sundaram, and D. Zamboni, "Analysis of a denial of service attack on TCP," in *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, California, May 1997, pp. 208–223.

[188] SecuriTeam, "Kiss of Death - a new denial of service attack," 1999.

[189] D. Senie, "Changing the default for directed broadcasts in routers," Internet Engineering Task Force, Request for Comments RFC 2644, Aug. 1999.

[190] S. Sesay, Z. Yang, B. Qi, and J. He, "Simulation comparison of four wireless ad hoc routing protocols," *Information Technology Journal*, vol. 3, no. 3, pp. 219–226, 2004.

[191] A. Shaikh, R. Tewari, and M. Agrawal, "On the effectiveness of DNS-based server selection," in *Proceedings of the IEEE INFOCOM*, Anchorage, USA, Apr. 2001.

[192] C. Shannon and D. Moore, "The spread of the Witty worm," CAIDA, Tech. Rep., Mar. 2004.

[193] C. Shannon, D. Moore, and K. C. Claffy, "Beyond folklore: Observations on fragmented traffic," *IEEE/ACM Trans. Networking*, vol. 10, no. 6, pp. 709–720, Dec. 2002.

[194] S. Siewert, "Big iron lessons, part 2: Reliability and availability: What's the difference?" Mar. 2005. [Online]. Available: http://www-128.ibm.com/developerworks/power/library/pa-bigiron2/, [accessed Jan. 21, 2006].

[195] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, B. Schwartz, S. T. Kent, and W. T. Strayer, "Single-packet IP traceback," *IEEE/ACM Trans. Networking*, vol. 10, no. 6, pp. 721–734, Dec. 2002.

[196] S. Staniford, V. Paxson, and N. Weaver, "How to 0wn the Internet in your spare time," in *Proceedings of the 11th USENIX Security Symposium*, San Francisco, California, Aug. 2002.

[197] S. Staniford-Chen and L. T. Heberlein, "Holding intruders accountable on the Internet," in *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, California, May 1995.

[198] C. Stavroulopoulos, T. Antonakopoulos, and V. Makios, "Performance evaluation of mobile ad hoc network routing protocols for real time applications support," in *Proceedings of the 8th International Conference on Advances in Communications and Control*, Crete, Greece, June 2001.

[199] D. Sterne, K. Djahandari, B. Wilson, B. Babson, D. Schnackenberg, H. Holliday, and T. Reid, "Autonomic response to distributed Denial of Service attacks," in *Proceedings of Recent Advances in Intrusion Detection, 4th International Symposium*, Davis, California, USA, Oct. 2001, pp. 134–149.

[200] R. R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. J. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "Stream control transmission protocol," The Internet Society, Request for Comments RFC 2960, 2000.

[201] R. Stone, "Centertrack: An IP overlay network for tracking DoS floods," in *Proceedings of the 9th USENIX Security Symposium*, Denver, Colorado, Aug. 2000.

[202] The VINT Project, "The ns manual," 2003.

[203] S. Tyson and T. Jackson, *The Essence of Organizational Behaviour*. Hertfordshire, UK: Prentice Hall Europe, 1992.

[204] A. Urpi, M. Bonuccelli, and S. Giordano, "Modelling cooperation in mobile ad hoc networks: a formal description of selfishness," in *Proceedings of the WiOpt 03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, Sophia-Antipolis, France, Mar. 2003.

[205] U.S. Computer Emergency Readiness Team, "Vulnerability Note VU#181038: Microsoft Windows Metafile handler SETABORTPROC GDI Escape vulnerability," Jan. 2006.

[206] U.S. Department of Homeland Security, "Look before you click: Trojan horses and other attempts to compromise networks," 2005.

[207] P. Vixie, G. Sneeringer, and M. Schleifer, "Events of 21-Oct-2002," ISC/UMD/Cogent, Tech. Rep., Nov. 2002.

[208] L. von Ahn, M. Blum, and J. Langford, "Telling humans and computers apart automatically," *Communications of the ACM*, vol. 47, no. 2, pp. 57–60, Feb. 2004.

[209] D. Wagner, J. S. Foster, E. A. Brewer, and A. Aiken, "A first step towards automated detection of buffer overrun vulnerabilities," in *Proceedings of the Network and Distributed System Security Symposium*, San Diego, California, Feb. 2000.

[210] W. Wang, Y. Lu, and B. K. Bhargava, "On security study of two distance vector routing protocols for mobile ad hoc networks," in *Proceedings of the IEEE PerCom*, Fort Worth, USA, Mar. 2003, pp. 179–186.

[211] S. Weber and L. Cheng, "A survey of anycast in IPv6 networks," *IEEE Commun. Mag.*, vol. 42, no. 1, pp. 127–132, Jan. 2004.

[212] G. B. White, E. A. Fisch, and U. W. Pooch, *Computer System and Network Security*. Boca Raton, USA: CRC Press, 1996.

[213] C. D. Wickens and J. G. Hollands, *Engineering Psychology and Human Performance*, 3rd ed. Upper Saddle River, USA: Prentice Hall, 2000.

[214] Wikipedia, "Information society," 2005. [Online]. Available: http://en.wikipedia.org/wiki/Information_society, [accessed Jan. 7, 2006].

[215] M. M. Williamson, "Throttling viruses: Restricting propagation to defeat malicious mobile code," in *Proceedings of Annual Computer Security Applications Conference*, Las Vegas, Nevada, USA, Dec. 2002.

[216] J. M. Wing, "A call to action: Look beyond the horizon," *IEEE Security & Privacy*, vol. 1, no. 6, pp. 62–67, Nov./Dec. 2003.

[217] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *IEEE Computer*, vol. 35, no. 10, pp. 54–62, Oct. 2002.

[218] Y. Xiao, X. Shan, and Y. Ren, "Game theory models for IEEE 802.11 DCF in wireless ad hoc networks," *IEEE Communications Magazine*, vol. 43, no. 3, pp. S22–S26, Mar. 2005.

[219] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, "Security in mobile ad hoc networks: Challenges and solutions," *IEEE Wireless Communications*, vol. 11, no. 1, pp. 38–47, Feb. 2004.

[220] Y. Yuan, "Report on DIMACS workshop on mobile and wireless security," University of Maryland, Tech. Rep., Nov. 2004.

[221] Y. Zhang and V. Paxson, "Detecting backdoors," in *Proceedings of the 9th USENIX Security Symposium*, Denver, Colorado, USA, Aug. 2000.

[222] Y. Zhang and V. Paxson, "Detecting stepping stones," in *Proceedings of the 9th USENIX Security Symposium*, Denver, Colorado, USA, Aug. 2000.

ISBN 951-22-8214-3
ISBN 951-22-8215-1 (PDF)
ISSN 1795-2239
ISSN 1795-4584 (PDF)