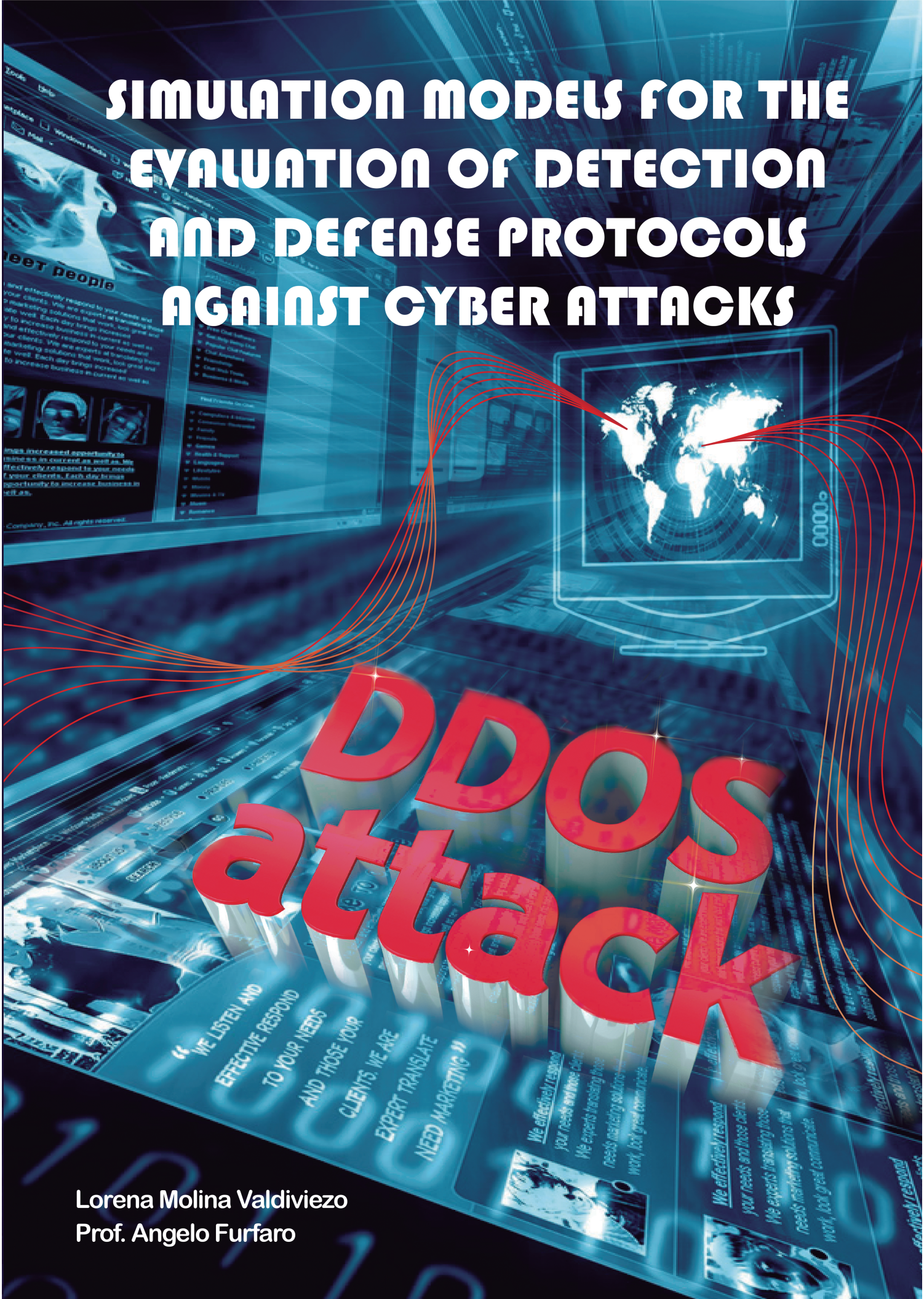
SIMULATION MODELS FOR THE EVALUATION OF DETECTION AND DEFENSE PROTOCOLS AGAINST CYBER ATTACKS



Lorena Molina Valdiviezo
Prof. Angelo Furfaro

UNIVERSITÀ
DELLA CALABRIA



UNIVERSITA' DELLA CALABRIA

Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica (DIMES)

Dottorato di Ricerca in

Ingegneria dei Sistemi e Informatica

Con il contributo di

"Secretaria Nacional de Educación Superior, Ciencia, Tecnología e Innovación"

(SENESCYT - ECUADOR)

CICLO

XXVII

"SIMULATION MODELS FOR THE EVALUATION OF DETECTION AND DEFENSE PROTOCOLS AGAINST CYBER ATTACKS"

Settore Scientifico Disciplinare

ING-INF/05 INFORMATION PROCESSING SYSTEMS

Coordinatore: Ch.mo Prof. Felice Crupi.

Firma

Supervisore: Ch.mo Prof. Angelo Furfaro.

Firma

Dottoranda: Dott.ssa Lorena P. Molina Valdiviezo.

Firma

I would like to dedicate this thesis to my loving family (My father Segundo, my mother Zoila, and sisters and brothers: María Fernanda, Sixto†, Marco, Anita, Lida, Edward, Milton and especially to Emilita).

Acknowledgements

First of all, my heartfelt appreciation goes to God and to the Virgin of Guadalupe.

I would like to thank my thesis supervisor Professor Angelo Furfaro from the bottom of my heart for his patience, kindness and assistance, he continuously guided me by overseeing my research activities.

I also would like to take this opportunity to thank the University of Calabria (UNICAL), the Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT), and the National University of Chimborazo (UNACH) for bringing me to the academic world.

I would also like to acknowledge my colleagues, Giovanna Malena and Andrea Parise, for their collaboration and especially for their friendship.

I heartily thank my friends, for their full support. The list is too long to complete here, but I do appreciate their time and help. I also owe a very important debt to Pauly and Yessy, who always gave me emotional support.

Finally, I would like to give special thanks to my father, my mother and Emilita, who let me come to Italy and who inspired me to do my best, and my brothers, sisters, nieces, nephews, aunts, uncles, and cousins for their sweet support and effort.

Preface

Data networks have become an essential part of our life, currently, they are used to accomplish almost all our common activities, this has permitted (depending on the infrastructure) to develop individual, organization, government and national activities. Nevertheless, a few years ago, US-CERT (United States Computer Emergency Readiness Team) received users' and administrators' reports due to anomalies and attacks detected in their networks services and systems.

Unfortunately, data networks breaches have become both an everyday occurrence and fashionable. Therefore, researchers and organizations have been developing efficient cyber defense methods and technologies. These methods suggest to experiment the developed defense solutions into a simulated network environment and after testing them in real systems. When cyber attacks are simulated using modelling applications, the obtained results establish the possibility to develop and classify attack mechanisms, and allow for the efficient evaluation of cyber attack scenarios. In this context, computer simulation have become a powerful and necessary tool in cyber defense (e.g. Discrete Event Simulation (DES)).

Several cyber attacks are directed against service availability. These attacks aim to hinder the normal operation of the target systems. One of the most common and effective attack is the Distributed Denial of Service (DDoS), which is achieved by exploiting a network of controlled hosts (botnet). The attacker remotely governs these computers and instructs them to send malicious traffic towards the victim. A DDoS can be performed at internet, transport and/or application layer. These attacks work by sending a large amount of messages to a destination, which is the victim of the attack, in order to consume its resources, e.g. by issuing requests that need a large processing time, by exhausting the available communication bandwidth or by exhausting its memory. Once a resource is exhausted, the legitimate clients cannot use the service. Attackers can send a variety of packets, which could be similar to legitimate traffic and adopt, to a certain extent, the structure and arbitrary statistics, which greatly facilitates concealment of the attack.

From the above reasons, it is necessary to devise and develop new and efficient defense mechanisms to face DDoS attacks in real time. To this goal it is important to exploit software tools to simulate attacks and evaluate the developed defense mechanisms.

This Ph.D. thesis is devoted to model and to simulate different scenarios with Distributed Denial of Service (DDoS) attacks, which consists mainly in designing (or extending) and implementing *StopIt* [1] defense mechanism using NeSSi^2 and ns-3 as network simulator tools.

The specific contributions of the thesis are the following:

- Design and implementation of a simulation model of DDoS attacks in the context of NeSSi^2 network simulator;
- performance evaluation and limitations of the standard *StopIt* mechanism to face DDoS security attacks by implementing a specific simulation model using the ns-3 network simulator;
- extension of *StopIt* by taking advantage from the integration of the *DiffServ* architecture, typically used to guarantee specific quality of service levels within a communication network;

The thesis is structured as follows:

- *Chapter 1* introduces basic concepts related to Cyber attacks and Standards for Information Security.
- *Chapter 2* gives a survey on Distributed Denial of Service (DDoS), detection, defense techniques. This chapter mainly focuses on the *StopIt* defense mechanisms.
- *Chapter 3* centres on the modeling and simulating of DDoS attack scenarios at internet and application layer with NeSSi^2 network simulator tool, and it deeply analyzes how attacks may evolve and their effects.
- *Chapter 4* analyzes the DDoS amplification attacks effects, over NTP and DNS protocols.
- *Chapter 5* centres on the proposal of a new strategy to improve the performance of a widely used hybrid filter-based mechanism named *StopIt*.
- *Chapter 6* presents conclusions and a future work.

.

Arcavacata di Rende, Cosenza, Italy *Lorena Paulina Molina Valdiviezo*
November 2015

Prefazione

Le reti di dati sono ormai diventate parte integrante delle nostre vite e, al momento, sono utilizzate per portare a termine quasi tutte le nostre azioni più comuni. Ciò ha permesso (a seconda dell'infrastruttura) lo sviluppo di attività a livello individuale, aziendale, governativo e nazionale. Ciò nondimeno, qualche anno fa, il CERT degli Stati Uniti (United States Computer Emergency Readiness Team) ha ricevuto report da parte di utenti e amministratori dovuti ad anomalie ed attacchi individuati all'interno dei loro servizi e sistemi di rete.

Sfortunatamente, le violazioni delle reti di dati sono diventate un evento quotidiano e persino di moda. Dunque, ricercatori e organizzazioni hanno sviluppato metodi e tecnologie di cyber difesa efficienti. Le metodologie di supporto consigliano di sperimentare le soluzioni di difesa sviluppate all'interno di un ambiente di rete simulato e successivamente di testarle in sistemi reali. Quando gli attacchi cibernetici sono simulati utilizzando applicazioni di modellazione, i risultati ottenuti stabiliscono la possibilità di sviluppare e classificare i meccanismi di attacco e permettono una valutazione efficiente degli scenari di attacco. In questo contesto, le tecniche di simulazione (ad esempio, la simulazione ad eventi discreti) sono divenute uno strumento necessario e potente per la difesa dagli attacchi informatici.

Molti attacchi cibernetici sono diretti contro la disponibilità del servizio. Questi attacchi hanno lo scopo di intralciare il normale funzionamento dei sistemi obiettivo. Uno degli attacchi più comuni ed efficaci è il Distributed Denial of Service (DDoS), che è ottenuto sfruttando una rete di host controllati (botnet). L'attaccante dirige in remoto questi computer e dà istruzioni affinchè inviino traffico dannoso alla vittima. Un DDoS può essere effettuato a livello di rete, trasporto e/o applicazione. Questi attacchi procedono inviando una grande quantità di messaggi ad una destinazione precisa, che è la vittima dell'attacco, allo scopo di consumarne le risorse, ad esempio formulando richieste che necessitano un tempo di processazione molto lungo, esaurendo così la larghezza di banda di comunicazione disponibile o la sua memoria. Una volta consumata una risorsa, i clienti legittimi non possono più utilizzare il servizio.

Gli attaccanti possono inviare diversi pacchetti, che potrebbero essere simili al traffico legittimo riproducendone, fino ad un certo punto, la struttura e alcuni indici statistici, il che facilita largamente l'occultamento dell'attacco.

Per tutte queste ragioni, è necessario concepire e sviluppare meccanismi di difesa nuovi ed efficaci per affrontare in tempo reale gli attacchi DDoS. A questo scopo, è importante utilizzare strumenti software per simulare gli attacchi e valutare i meccanismi di difesa sviluppati.

La presente tesi di dottorato ha lo scopo di modellare e simulare diversi scenari con attacchi di tipo Distributed Denial of Service (DDoS). Ciò consiste principalmente nella modellazione, estensione e implementazione del meccanismo di difesa *StopIt* [1] utilizzando come strumenti di simulazione di rete NeSSi^2 e ns-3.

I contributi specifici della tesi sono i seguenti:

- Definizione e implementazione di un modello di simulazione di attacchi DDoS nel contesto del simulatore di rete NeSSi^2;
- valutazione della performance e dei limiti del meccanismo *StopIt* standard nell'affrontare attacchi DDoS alla sicurezza tramite l'implementazione di un modello di simulazione specifico con l'uso del simulatore di rete ns-3;
- estensione di *StopIt* sfruttando l'integrazione dell'architettura *DiffServ*, utilizzata tipicamente per garantire livelli specifici di qualità del servizio all'interno di una rete di comunicazione;

La tesi è strutturata come segue:

- Il *Capitolo 1* introduce concetti base relativi ai cyber attacchi e agli standard di sicurezza informatica..
- Il *Capitolo 2* offre un quadro generale sugli attacchi di tipo Distributed Denial of Service (DDoS), e sulle tecniche di rilevamento e difesa. Questo capitolo è incentrato principalmente sul meccanismo di difesa *StopIt*.
- Il *Capitolo 3* è focalizzato sulla modellazione e simulazione di scenari d'attacco di tipo DDoS a livello di rete e applicazione tramite lo il simulatore di rete NeSSi^2, con un'analisi approfondita di come gli attacchi possono evolvere e sui loro potenziali effetti.
- Il *Capitolo 4* analizza gli effetti di amplificazione degli attacchi DDoS, sui protocolli NTP e DNS.
- Il *Capitolo 5* si incentra sulla proposta di una nuova strategia per migliorare le prestazioni di un meccanismo largamente utilizzato, ibrido e filter-based, chiamato *StopIt*.
- Il *Capitolo 6* presenta le conclusioni e il lavoro futuro.

Arcavacata di Rende, Cosenza, Italia *Lorena Paulina Molina Valdiviezo*
Novembre 2015

Contents

Part I An overview of Cyber security and Distributed Denial of Service (DDoS) Attacks and Defense

1 Cyber security . . . 3
1.1 Introduction . . . 3
1.2 Cyber attacks . . . 4
1.2.1 Type of attacks . . . 5
1.3 Malicious Network . . . 6
1.4 Standard for Information Security . . . 7
1.5 Conclusions . . . 9

References . . . 9

2 DDoS Attacks Detection and Defense . . . 11
2.1 Introduction . . . 11
2.2 Distributed Denial of Service (DDoS) Attacks . . . 12
2.3 Detection and Defense Techniques . . . 14
2.3.1 Low Rate DDoS (LorDDoS) Attack Detection . . . 14
2.3.2 Network Traffic(Netra) Based Detection . . . 19
2.3.3 *StopIt* defense mechanisms . . . 20
2.4 Conclusions . . . 23

References . . . 23

Part II DDoS: Simulation Experiments

3 Modeling and Simulation of Internet and Application Layer DDoS Attacks . . . 29
3.1 Introduction . . . 29
3.2 IntDDoS and AppDDoS Attacks . . . 31

3.3 Domain Name System (DNS) server . . . 32
3.4 NeSSi^2 Network Simulator . . . 33
3.5 Modeling and Simulation DDoS Attacks . . . 33
3.5.1 Client . . . 34
3.5.2 Bot . . . 36
3.5.3 Zombie . . . 36
3.5.4 DNS Server . . . 41
3.6 Results . . . 41
3.7 Conclusions . . . 46

References . . . 46

4 A Simulation Model of DDoS Amplification Attacks . . . 49
4.1 Introduction . . . 49
4.2 DDoS Amplification/Reflection Attacks . . . 50
4.2.1 DNS Amplification Attack . . . 51
4.2.2 NTP Amplification Attack . . . 52
4.3 Simulation Model . . . 54
4.4 Simulation Results . . . 57
4.5 Conclusions . . . 59

References . . . 59

5 A Defense Technique Against Indirect DDoS Flooding Attacks . . . 61
5.1 Introduction . . . 61
5.2 The Proposal . . . 62
5.2.1 DiffServ for traffic management . . . 63
5.3 Simulation Analysis and Results . . . 65
5.3.1 Modelling with ns-3 . . . 66
5.3.2 Simulation Scenario . . . 67
5.3.3 Direct Flooding Attack . . . 70
5.3.4 Shared Link Flooding Attack . . . 70
5.4 Conclusion . . . 71

References . . . 72

6 Conclusions and Outlook . . . 73

References . . . 75

List of Figures

1.1 Five Common Cyber Crimes Patterns 4
1.2 Elements of a Botnet 6
1.3 COBIT Framework Evolution 8

2.1 Report 2014 14
2.2 Report 2013 15
2.3 Report 2012 16
2.4 DDoS Attack Network 16
2.5 DDoS attacks classification 17
2.6 A collaborative *LorDDoS* attack detection algorithm [2] 18
2.7 Pseudocode Optimization 19
2.8 StopIt operation 22

3.1 internet Layer Attack 31
3.2 Application Layer Attack 32
3.3 Network Topology 34
3.4 Creation of an IP address 35
3.5 `IPFactory` 35
3.6 TicketEvent 36
3.7 Next tick 36
3.8 Event Record 36
3.9 `Bot` class implemented 37
3.10 `Zombie` class implemented 38
3.11 `handleIncomingPacket` function implemented 39
3.12 `handleEvent` function implemented 40
3.13 Server behavior 41
3.14 Request client 42
3.15 Server responses 43
3.16 Cumulative requests. 43
3.17 Attacked Network Topology 44
3.18 Requests sent to DNS server 44

3.19 Sever under attack 45
3.20 DDoS Packets and Regular Packets 45

4.1 Amplification/reflection DDoS attack schema [3] 51
4.2 DNS Amplification Attack 52
4.3 NTP reflection DDoS Attack 53
4.4 Class diagram 54
4.5 Master packet format 55
4.6 Topology 56
4.7 Comparison between DNS and NTP impact on victim link 58
4.8 DNS server in-out 58
4.9 NTP server in-out 59

5.1 *StopIt* packet format for *DiffServ* activation 64
5.2 *StopIt* and *DiffServ* routers interactions 65
5.3 *StopIt* class hierarchy 66
5.4 AccessRouter class 67
5.5 StopItServer Class 68
5.6 `StopItServer` interface definition 69
5.7 Network topology 70
5.8 (a) Direct DNS DDoS attack (b) Detail of legal and malicious DNS traffic 70
5.9 Indirect DDoS attack (StopIt) 71
5.10 Indirect DDoS attack (*StopIt* + *DiffServ*) 71

List of Tables

1.1 Command and Control (C&C) topologies [4] 6
1.2 Overview of COBIT, ITIL and ISO 27002 [5] 9

4.1 Bandwidth of links between subnets 57
4.2 Subnet internal link bandwidths 57
4.3 Statistics for the victim's link 58
4.4 Incoming and outgoing packets for NTP and DNS server 59

5.1 Simulation parameters 68

Part I

An overview of Cyber security and Distributed Denial of Service (DDoS) Attacks and Defense

1

Cyber security

Abstract

Nowadays, the Internet plays a role of utmost importance in our society, because many kinds of transactions are made through computer systems in homes, schools, businesses and governments in the cyber world, and this establishes strong requirements in order to warranty the security of information systems against different types of cyber threats and attacks. This chapter presents a brief overview of cyber attacks and malicious network. Additionally, a short review of Standards for Information Security, is reported.

1.1 Introduction

In the last few years, cyber crimes and attackers have exponentially increased all the time exploring new ways to evade security solutions to get dishonest access to data networks. In 2000, it was reported the first cyber attack, able to infect almost 45 Million of Internet users. These cyber attacks have led to the development of the field of *Cyber Security* [1]. In recent years, issues related to *Cyber Security* aspects, mainly focused on the security of computer systems and the services they offer, have gained considerable importance. Many aspects of the daily people lives, companies and even national governments, are incessantly affected by these issues to ensure the integrity of information systems and data managed and transmitted through communication networks [2, 3]. In addition, open and accessible networks such as the Internet, are particularly suitable to the widespread of malwares able to exploit vulnerabilities in connected systems with the aim of launching targeted attacks to undermine security. To deal with these cyber attackers, a careful and accurate analysis of the most frequent attacks in both wired and wireless communication networks is crucial to design more effective defense.

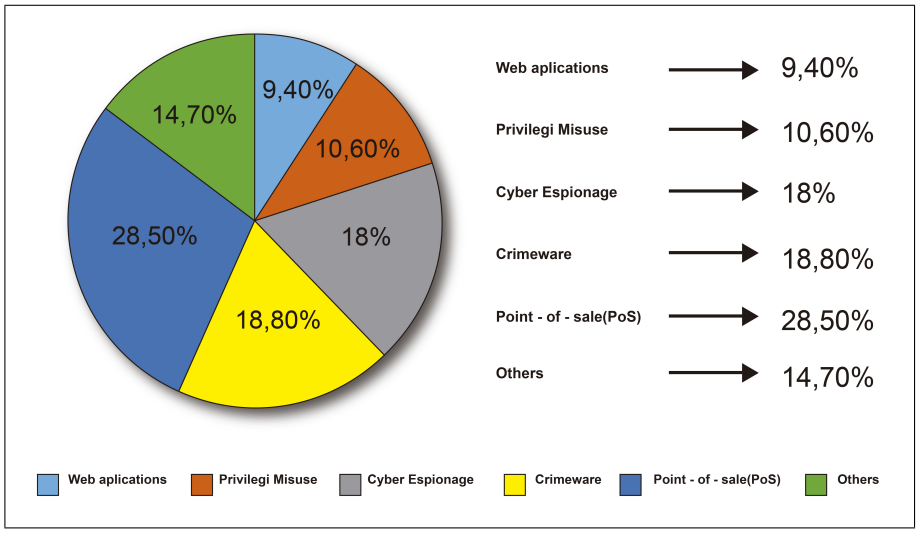


Fig. 1.1. Five Common Cyber Crimes Patterns

A report [4], published in 2015 identified five common cyber crimes patterns behind todays data breaches. There are: Web applications with 9.4% of incidents, then, Privilege Misuse (10.6%), Cyber espionage (18%), Crimeware (18.8% of incidents), and finally Point-of-sale (PoS) with 28.5% of incidents. Figure 1.1 summarizes the percentage distribution of these attack types. In order to counteract, there exist standards that help organizations keep information assets secure [5], therefore, these organizations should be implemented properly to prevent, defend and respond to cyber attacks and other threats.

From that mentioned above, this chapter is dedicated to doing a general review of the concepts to begin the research. This chapter has the following structure: Section 1.2 present an overview brief about cyber attacks and their types. Section 1.3 describe malicious network (Botnet). Section 1.4 summarized the Standards for Information Security. Finally, the conclusions are presented together with an indication of future work in Section 1.5.

1.2 Cyber attacks

The ever increasing growth of Internet has achieved global connections among millions of networks and devices which exchange information through computer systems in homes, schools, businesses and governments in the cyberworld. At the same time, cybercriminals have developed skills that imply critical issues related to network and information security. Cyber crime is extensive of dishonest digital activities targeted at businesses and governments in the cyber world in order to alter, disrupt, or destroy computer systems, networks, or the information or programs. This term applies to a wide range of

targets and cyber attack methods [6, 7, 8]. The cyber opponent can be a person or organization that is carrying it out the cyber attacks. Cyber oponents can be classified into four groups [6, 7]:

- An individual attacker
- The activist
- Organized crime, and
- Governments.

Cyber attacks by these cyber opponents have different features and patterns, such as the scale of impact, the type of target, the attack methods, among others.

1.2.1 Type of attacks

In 1988, Robert Tappan Morris a student at Cornell University in the US created the first computer worm transmitted over the data networks. The aim was to determine the vastness of the cyber world [9].

Today's cyber world is full of cyber attacks. The following list reports some of the most important attacks on cyber space [1, 10]:

- Denial of Service (DoS)
- Distributed Denial of Service (DDoS)
- Malvertising
- Man in the Middle (MITM)
- Malware
- Search Poisoning
- Spamming
- Password Attacks
- Phishing
- Website Threats.

Cyber security, also referred to as information technology security, focuses on protecting computers, networks, programs and data from unintended or unauthorized access, change or destruction. According to [1], the important goals of *cyber security* are:

1. securely obtain and share information for accurate decision-making
2. find and deal with vulnerabilities within applications
3. prevent unauthorized access, and
4. protect confidential information.

There are companies that provide *cyber security* solutions to certain types of cyber attacks, such as: Accenture, Arbor Networks, CISCO, EADS, HP, IBM, Invensys, Unisys, so on.

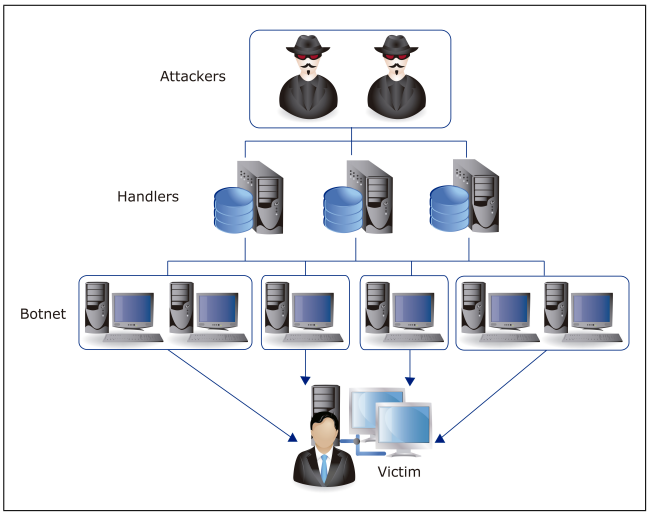


Fig. 1.2. Elements of a Botnet

1.3 Malicious Network

A malicious network is defined as a set of dishonest activities to alter, disrupt, deny, degrade or destroy information and service resident in data networks. A typical and dominant malicious network are called *botnets* (depicted in Fig. 1.2); in which an attacker remotely controls a large number of zombies. In the malicious network, zombies are called bots, these zombies are infected with malware that spread like viruses or worms. A *Botnet* is controlled by botmasters through some Command and Control (C&C) infrastructure [11, 12, 13]. According to [13], a problem for botnet cybers opponents is how to communicate with each bot instance. The authors have identified three topologies with their benefits and weaknesses as shown in Table 1.1, to examine the implications of several bot communication methods.

Table 1.1. Command and Control (C&C) topologies [13]

Topology	**Design Complexity**	**Detect ability**	**Message Latency**	**Surviv ability**
Unstructured	Low	High	High	High
Peer-to-Peer	Medium	Low	Medium	Medium
Centralized	Low	Medium	Low	Low

Recently, botnets evolved into more sophisticated structures allowing to achieve complex forms of cyber attacks. Botnets an employed for DDoS attacks involving an amount of computers. Agobot and SDBot [14, 15] both

have remotely accessible commands for initiating DDoS attacks. Such capabilities were used in DDoS extortion scams that provided cyber opponents with economic purposes[13].

1.4 Standard for Information Security

Cyber security is an issue for both network managers and end-users, so organizations must implement standards to prevent, detect and react to possible cyber attacks on the data networks. A standard is a document that provides guidelines, specifications or requirements, and that can be used consistently to ensure products, materials, processes and services are fit for their purpose. There are several standards to ensure the safety, some of them are [5, 16, 17]:

- COBIT
- ISO/IEC 27002
- ITIl.

The Control Objectives for Information and related Technology (COBIT) standard provides guidelines on what can be done in an organization concerning control activities, measurement, and documentation of operations and processes. COBIT framework authors are non-profit organization Information System Audit and Control Association(ISACA) and IT Governance Institute(ITGI) [18]. Figure 1.3 depicts the development of the COBIT framework and roles it has had through his upgrading and development [19]. COBIT consist of 34 key organization control processes, it contains over 300 detailed IT control. The COBIT Standard consists of four domains [18, 20]:

- Plan and Organize
- Acquire and Implement
- Deliver and Support, and
- Monitor and Evaluate

The International Organization for Standardization (ISO), develops and publishes international standards. ISO defined over 20500 standards; these standards developed are very useful [5, 16, 21]. The following list reports some most commonly referenced ISO security standards [17, 22, 23]:

- Code of Practice for Information Security Management (ISO/IEC 27002:2005)
- Information Security Management System - Requirements (ISO/IEC 27001:2005)
- Evaluation Criteria for IT Security (ISO/IEC 15408)
- IT Security Management (ISO/IEC 13335)
- IT- Security techniques - Storage security (ISO/IEC 27040:2015)

The Information Technology Infrastructure Library (ITIL) standard is based on defining best practice processes for IT service management and support, rather than on defining a broad-based control framework. The ITIL Standard framework is the most complete and comprehensive framework for

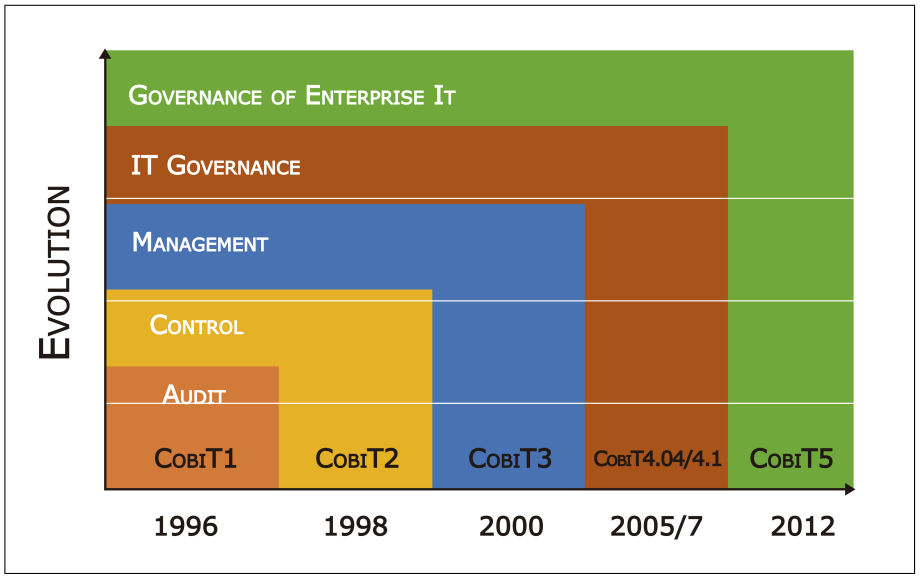


Fig. 1.3. COBIT Framework Evolution

the management of IT services worldwide [24]. The ITIL Standard consists of five processes [20]:

- Service Strategy
- Service Design
- Service Transition
- Service Operation, and
- Continual Service Improvement.

Table 1.2 summarized the difference between standards, according to [18].

Table 1.2. Overview of COBIT, ITIL and ISO 27002 [18]

Parameters	**COBIT**	**ITIL**	**ISO 27002**
Function	Mapping IT Process	Mapping IT Service Level Management	Information Security Framework
Area	34 Processes, 4 Domains	9 Processes	10 Domains
Issuer	ISACA - ITGI	OGC	ISO Board
Implementation	Information System Audit	Manage Service Level	Compliance with security standards
Consultant	Accounting Company, IT Consulting Company	IT Consulting Company	Security Company, IT and Network Consulting

1.5 Conclusions

Cyber attackers cause serious damage, so it is necessary to design and to implement *Cyber Security* systems through the adequate techniques. The knowledge of cyber threats is fundamental to devise and evaluate solutions to detect and defend a network the cyber attacks. On the other hand, cyber attacks continue to plague organizations around the cyber world. These attacks are on the rise as cyber opponents increase their focus and know-how. Therefore, the problem demands an international solution, so that organizations should apply standards, such as COBIT, ISO/IEC 27002.

References

1. T. Mahmood and U. Afzal, "Security analytics: Big data analytics for cybersecurity: A review of trends, techniques and tools," in *Information Assurance (NCIA), 2013 2nd National Conference on*, Dec 2013, pp. 129–134.
2. PwC, "UK cyber security standards research," Department for Business, Innovation & Skills Cabinet Office, 2013.
3. M. Angelini, M. C. Arcuri, R. Baldoni, C. Ciccotelli, G. A. Di Luna, L. Montanari, I. C. Panetta, L. Querzoni, and N. V. Verde, "Italian cyber security report: Critical infrastructure and other sensitive sectors readiness," Department for Business, Innovation & Skills Cabinet Office, 2013.
4. D. Bisson, "The 5 most common attack patterns of 2014," 2015. [Online]. Available: http://www.tripwire.com/state-of-security/security-data-protection/cyber-security/the-5-most-common-attack-patterns-of-2014/
5. ISO, "Iso/iec 27001 - information security management." [Online]. Available: http://www.iso.org/iso/home/standards/management-standards/iso27001.htm
6. K. A. N.V., "The five most common cyber security mistakes," 2013. [Online]. Available: https://www.kpmg.com/Global/en/IssuesAndInsights/ArticlesPublications/Documents/five-most-common-cyber-security-mistakes.PDF

7. K. LLP, "Cyber security: its not just about technology," 2014. [Online]. Available: http://www.kpmg.com/Global/en/IssuesAndInsights/ArticlesPublications/Documents/cyber-security-not-just-technology.pdf
8. M. C. Waxman, "Cyber-attacks and the use of force: back to the future of article 2 (4)," *Yale Journal of International Law*, vol. 36, 2011.
9. ARN, "Top 10 most notorious cyber attacks in history." [Online]. Available: http://www.arnnet.com.au/slideshow/341113/top-10-most-notorious-cyber-attacks-history/
10. M. Sullivan, "8 types of cyber attacks your business needs to avoid," 2015. [Online]. Available: http://quickbooks.intuit.com/r/technology-and-security/8-types-of-cyber-attacks-your-business-needs-to-avoid
11. A. Ghorbani, W. Lu, and M. Tavallaee, "Network attacks," in *Network Intrusion Detection and Prevention*, ser. Advances in Information Security. Springer US, 2010, vol. 47, pp. 1–25. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-88771-5_1
12. P. Djalaliev, M. Jamshed, N. Farnan, and J. Brustoloni, "Sentinel: Hardware-accelerated mitigation of bot-based ddos attacks," in *Computer Communications and Networks, 2008. ICCCN '08. Proceedings of 17th International Conference on*, Aug 2008, pp. 1–8.
13. M. Bailey, E. Cooke, F. Jahanian, Y. Xu, and M. Karir, "A survey of botnet technology and defenses," in *Conference For Homeland Security, 2009. CATCH'09. Cybersecurity Applications & Technology*. IEEE, 2009, pp. 299–304.
14. S. Stankovic and D. Simic, "Defense strategies against modern botnets," *arXiv preprint arXiv:0906.3768*, 2009.
15. C. Schiller and J. Binkley, *Botnets: The Killer Web Applications*. Elsevier Science, 2011. [Online]. Available: https://books.google.it/books?id=4MAuVjOx6kIC
16. C. Davis, "Iso 14001: What is it? why should i care?" 2015. [Online]. Available: http://blog.claytonhomes.com/blog/iso-14001-what-is-it-why-should-i-care
17. G. of the HKSAR, "An overview of information security standards," 2008. [Online]. Available: http://www.infosec.gov.hk/english/technical/files/overview.pdf
18. D. Radovanovic, T. Radojevic, D. Lucic, and M. Sarac, "It audit in accordance with cobit standard," in *MIPRO, 2010 Proceedings of the 33rd International Convention*, May 2010, pp. 1137–1141.
19. Y.-W. Shin, "It governance: Focusing on cobit model," 2014. [Online]. Available: http://www.lgcnsblog.com/inside-it/it-governance-focusing-on-cobit-model/
20. P. Charuenporn and S. Intakosum, "Qos-security metrics based on itil and cobit standard for measurement web services." *J. UCS*, vol. 18, no. 6, pp. 775–797, 2012.
21. R. Frost, "International organization for standardization (iso)," *The Quality Assurance Journal*, vol. 8, no. 3, pp. 198–206, 2004.
22. S. Donaldson, S. Siegel, C. Williams, and A. Aslam, "Cybersecurity frameworks," in *Enterprise Cybersecurity*. Apress, 2015, pp. 297–309. [Online]. Available: http://dx.doi.org/10.1007/978-1-4302-6083-7_17
23. ISO, "Iso/iec 27040:2015," 2015. [Online]. Available: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=44404
24. A. Sharifi, M. A. Beheshtizad, H. Sharifi, A. Nourollahi, J. Sharifi, and R. Asadi, "Designing an information technology model for audit in banks," *IJCER*, vol. 4, no. 1, pp. 17–22, 2015.

2

DDoS Attacks Detection and Defense

Abstract

Distributed Denial of Service (DDoS) flooding attack is one of the most diffused and effective threat against services and applications running over the Internet. Its distributed and cooperative nature makes it complicated to prevent and/or to counteract. *StopIt* is a robust, filter-based defence mechanism which is able to deal with various types of massive DDoS flooding attacks but which fails when the DDoS is achieved indirectly, *i.e.* by congestion of a link shared with the victim. This chapter reviews basics of the DDoS attacks, describes detection methods for these attacks, which covers low rate based detection methods and network traffic based detection methods. The final part describes in details the *StopIt* defense technique and highlights its limitations.

2.1 Introduction

Cyberworld has become a heaven for cyber attackers. Nowadays, the most common and effective threat is the Distributed Denial of Service (DDoS) attack; DDoS has become the greatest security fear for data-network managers. "DDoS attacks are an attempt to make an on-line service unavailable by overwhelming it with traffic from multiple sources. They target a wide variety of important resources and present a major challenge to making sure people can publish and access necessary information"; these attacks are a combination application layer and volumetric attacks. DDOS attacks cause a waste of the resources in the victim system, and make services work improperly [1, 2, 3, 4].

As reported by Neustar in 2015, 40% of businesses say DDoS attacks are a growing threat to their organization. Nearly 1 in 3 companies would lose over $100K per hour. Named by 33% of companies, customer support is the #1 area affected by DDoS attacks. Most companies attacked are hit multiple times,

with 30% attacked over 10 times annually. 26% of companies attacked suffer loss of customer trust and brand damage. According to an annual report from Neustar Insights, 1 in 5 financial companies estimated outages would drain their revenues by $50K per hour. CNBC reports that in early 2013 U.S. banks, targets of the largest wave of DDoS ever, were knocked off-line a total of 249 hours. Resource allocation and loss of brand equity/consumer trust should be factored in too. Another study reported 35% of companies were DDoS attacked in 2012, slightly more than in 2011. The retail sector showed the most alarming year-to-year increase. In 2011, only 16% of participating retailers reported being hit. This increased by 144% in 2012: 39% of all retailers came under attack, along with 41% of e-commerce businesses. In 2012, the survey expanded to reveal information about the size of attacks as measured in bandwidth(application layer attacks) and packets per second, a standard measurement of traffic flowing through network routers(internet layer attacks). These attacks can be executed in two ways [2, 5]:

- Using a huge amount of traffic to exhaust the resources of a victim system (bandwidth, CPU cycles, and so on).
- To crash a victim system or to jam his internet connection by sending several carefully crafted packets.

According to Neustar in 2015, 45% of businesses take more than 1 hour to detect a DDoS attack. After detection, 51% take more than an hour to respond. Several models of identification that are classified according to where the detection takes place and the response to the attack: at the source, at the destination or hybrid strategies, the latter are the ones that turn out to be more effective but also more difficult to implement and put in place. In recent years, the major threats to the Internet and data networks have been the attacks by worms and distributed denial of service (DDoS), identifying several models have been developed since 2000 and continue research to find mechanisms detection and defense in the network [6]. The author cites the need to insert devices in network intrusion detection systems (NIDS) in order to conduct an audit and a forensic analysis by reviewing the records (Nehinbe, 2010). The research of [7] exposed the need of a comprehensive, collaborative and distributed defense mechanism approach after they categorized the different DDoS flooding attacks and classified according to the ability to prevent, protect, detect, and respond to DDoS flooding attacks in real-time and as close to the attacks sources.

The chapter first summarizes an overview of DDos attacks. Then it goes on by Detection and Defense mechanisms. After that, the *StopIt* mechanism to face DDoS attacks is described.

2.2 Distributed Denial of Service (DDoS) Attacks

Definition 2.1 (Distributed Denial of Service Attack).
A Distributed Denial of Service (DDoS) attack is a threat in cyber security,

and is organized by multiple distributed computers; this attack is implemented to send a large amount of traffic to the host or network (victim system), in the hope that the victim cannot handle it and will be overwhelmed.

DDoS attack is a heaven and a challenge to the data networks with increasing resources at the hands of the attackers. Such cyber attacks are intense but shorter-lived, whose goal is extended downtime [8, 9].

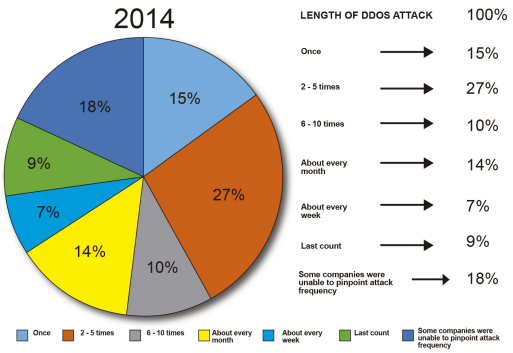
According to Arbor Networks Inc., in the Q1 2015 global DDoS attack data that shows a continuation of extremely high volume attacks, including the largest attack ever detected by Arbors ATLAS threat intelligence infrastructure, a 334Gbps attack targeting a network operator Asia. In Q1 2015, there were 25 attacks larger than 100Gbps globally. As the number of attacks continues to increase quarter by quarter, the average peak attack bandwidth and volume continue to drop. Average peak attack bandwidth was 5.95 Gbps in q1 2015, slightly down from the 6.41 Gbps average in q4 2014, and significantly lower than the average peak of 9.70 Gbps seen in q1 2014, according to Akamai. Fig. 2.1 depicts report by Neustar of length of DDOS attack and bandwidth used in DDoS attacks. In 2013, 60 percent of companies were DDoS attacked, up from 35 percent experiencing a disruptive attack in 2012, according to Arbor Network Inc. In the report submitted by the company Neustar in 2013 (see Fig. 2.2), over a third of all DDoS attacks (63%) lasted more than 24 hours: 37% in 2012 versus 35% in 2011. In fact, 17% of attacks lasted between 1 and 2 days, 7% of attacks between 3 and 7 days.

The longest attacks, those lasting over a week, increased from 10% in 2011 to 13% in 2012 (see Fig. 2.3(a)). In addition, the survey also looked at the size of internet layer attacks as measured in packets per second, a standard measurement of traffic flowing through network routers as shown Fig. 2.3(b). Any attack, including application layer attacks, can be measured in terms of bandwidth. Fig. 2.3(c) depict DDoS attacks by bandwidth size.

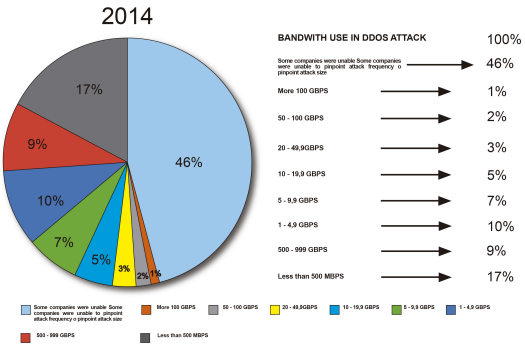
In general, the DDoS attack network is a hierarchical one. This network is composed of: attacker, handlers, agents and victim system, as depicted in Fig. 2.4 [5, 10].

- *Attacker:* The network contains one or more attacking hosts.
- *Handlers:* Some handlers are compromised hosts with a particular program running on them, capable of controlling several agents.
- *Agents:* A large number of agents or zombie hosts, who are compromised hosts that are running a special program and are responsible for generating a stream of packets towards the intended victim.
- *Victim system:* Target host or victim system.

According to [4], there are two principal classes of DDoS attacks: logic and flooding. On the other hand, DDoS attacks can be classified according to various dimensions. A taxonomy based on degree of automation, exploited vulnerability, attack rate dynamics and impact are reported in [11] and it is summarized in Fig. 2.5. A recent survey on defense mechanisms against DDoS is given in [7].



(a) Length of DDoS attacks



(b) Attacks Measured in Bandwidth (Application Layer)

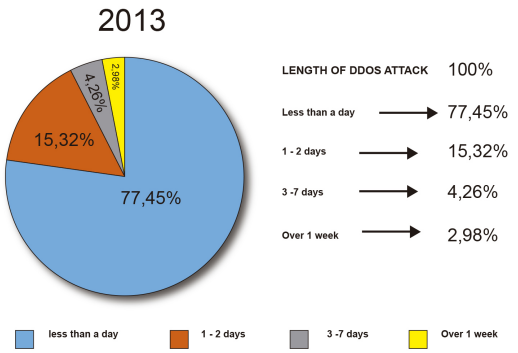
Fig. 2.1. Report 2014

2.3 Detection and Defense Techniques

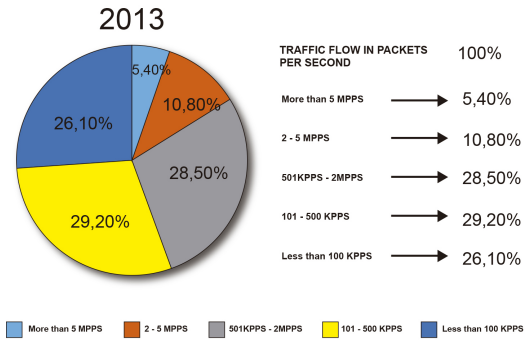
This section presents DDoS attacks detection methods and defense mechanisms.

2.3.1 Low Rate DDoS (LorDDoS) Attack Detection

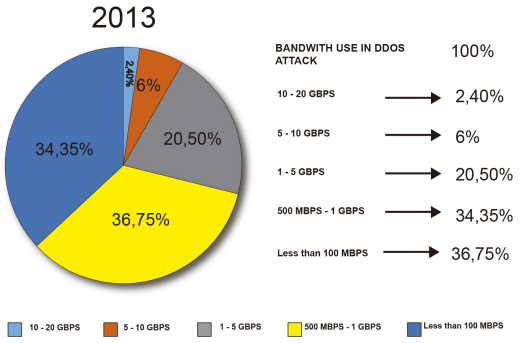
In this chapter, the term *LorDDoS attack* will be used to describe *Low rate Distribute Denial of Service attack*. A LorDDoS attack is an intelligent attack, and it has a significant ability to conceal the attack traffic because it is similar to genuine traffic [12, 13], this kind of attack is called *shrew attack* or *Reduction of Quality (RoQ) attack* [14]. To effect this attack, the attacker can send packets at an utmost low rate to victim system; these attacks are hard to detect. [12, 15]. Different schemes have been proposed to detect this kind of attack, such as the mechanism called *Collaborative Detection and Filtering (CDF)* of shrew attacks introduced by Chen and Hwang [16]. Another method



(a) Length of DDoS attacks

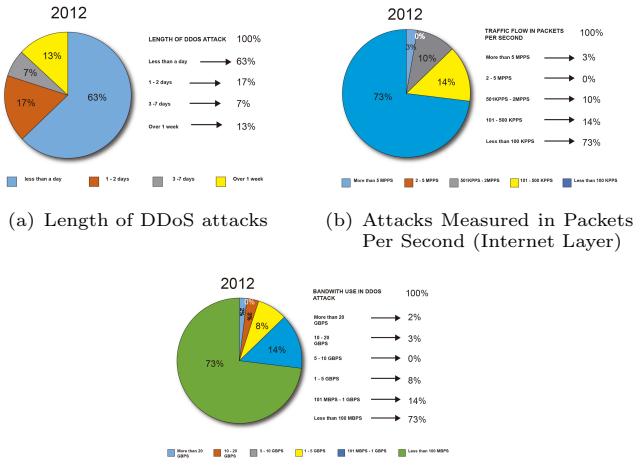


(b) Attacks Measured in Packets Per Second (Internet Layer)



(c) Attacks Measured in Bandwidth (Application Layer)

Fig. 2.2. Report 2013



(a) Length of DDoS attacks

(b) Attacks Measured in Packets Per Second (Internet Layer)

(c) Attacks Measured in Bandwidth (Application Layer)

Fig. 2.3. Report 2012

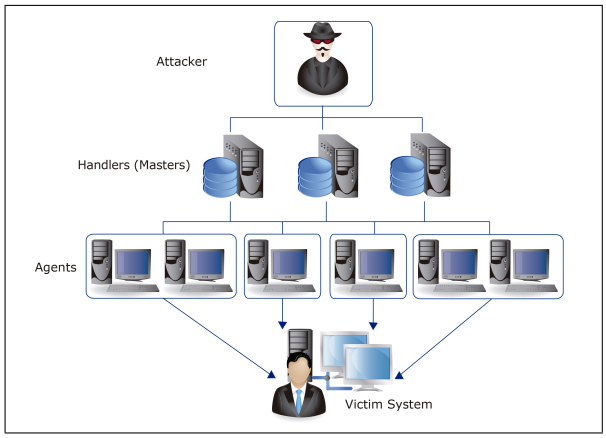


Fig. 2.4. DDoS Attack Network

has been presented by Yang Xiang, Ke Li, and Wanlei Zhou, by using information distance and entropy metrics by measuring a difference between genuine and attack traffic. There is Low-rate DDOS Attack Detection using Optimal Objective Entropy Method has been submitted by Poonam N. Jadhav and B. M. Patil [17].

CDF scheme proposed by Chen and Hwang to detect *LorDDoS* attack, is explained as follows [15, 16]:

– For a given sequence of network traffic, X, has been denoted it as x[1], x[2],..x. The autocorrelation of X is defined in 2.1.

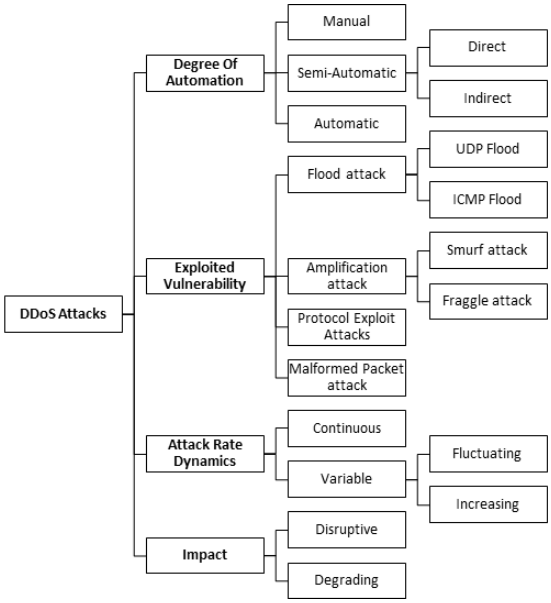


Fig. 2.5. DDoS attacks classification

$$Rxx(m) = \frac{1}{N-m} \sum_{n=0}^{N-m+1} x[n]x[n+m] \tag{2.1}$$

– The *Discrete Fourier Transform (DFT)* of the autocorrelation can display the periodicity in the frequency domain. *DFT* is defined in 2.2

$$DFT[Rxx(m), f] = \frac{1}{N} \sum_{n=0}^{N-1} Rxx(m) e^{\frac{-j2\pi fn}{N}} \tag{2.2}$$

where: f=0,1,...N-1

– *Power spectrum density (PSD)* is the output of the *DFT* processing on autocorrelation.

Authors concluded, that CDF scheme cuts off malicious flows containing *LorDDoS* attacks using a newly developed packet-filtering scheme, this scheme retained 99% of legitimate TCP flows, compared with only 20% TCP flows retained by using the Drop Tail algorithm, in detail [16].

Yang Xiang, Ke Li, and Wanlei Zhou have been proposed two new and effective information metrics for *LorDDoS* attacks detection: generalized entropy and information distance metric. The experimental results show that these metrics work effectively and stably, in detail [13]. Figure 2.6 shown the

1: Start
2: f, set the sampling frequency.
3: In routers R1 and R2, sampling the network traffic comes from the upstream routers from R3 to R6 and, LAN1, LAN2, in parallel.
4: Calculate in parallel the numbers of packet.
5: Calculate the probability distributions of the network traffic come from R3, R4, LAN1 and R5, R6, LAN2 in parallel.
6: Calculate $D_\alpha(P,Q) = D_\alpha(P \,||\, Q) + D_\alpha(Q \,||\, P)$
7: Sum the distances.
8: **if** the summed distance is more than the collaborative detection threshold σ **then**
9: the system detects the DDoS attacks
10: **else**
11: The routers forward the packets to the downstream routers.
12: **end if**
13: Return to step 3
14: End

Fig. 2.6. A collaborative *LorDDoS* attack detection algorithm [13]

designed the collaborative detection algorithm to detect a *LorDDos* attacks and discard its packets, presented by authors.

Authors concluded that proposed information metrics can substantially improve the performance of *LorDDoS* attacks detection and IP traceback over the traditional approaches.

Another method is Optimal Objective Entropy algorithm introduced by Yang Xiang, Ke Li, and Wanlei Zhou to face *LorDDoS* attacks; it is presented as follows [17]:

- *Entropy* is a measure of the uncertainty associated with a random variable.
- The probability distribution Pi is calculated in equation 2.3

$$Pi = Xi * \left(\sum_{i=1}^{m} Xi \right)^{-1} \tag{2.3}$$

  where:
  - i, denote the flow ID.
  - m, is the maxim number of flows.
  - Xi, denote number of packets flow.

- Equation 2.4 shown Entropy H.

$$H = -\sum_{i=1}^{m} Pi * log_2 Pi \tag{2.4}$$

- Figure 2.7 shown the pseudo code for optimization of threshold value, in detail [17].

```
1: Start
2: Initialize the Upper and Lower bound
3: diff = Hc-Hn. Calculate the difference of average attack and average normal
   entropy
4: repeat
5:     f=a(i)*d. Calculate deciding factor (f)
6: until diff>f
7: Trigger alarm
8: End
```

Fig. 2.7. Pseudocode Optimization

Authors concluded that this scheme using *optimal objective entropy* method permitted to increase the detection rate by implementing an objective function for optimization of the threshold value. The values derived by objective function can be tuned according to different traffic conditions.

2.3.2 Network Traffic(Netra) Based Detection

One data network attack trend is the use of network traffic (Netra) [4], which are called *Netra* in this chapter. *Netra* is an property of the cyberworld, it is routed at the level of destination prefixes. On the other hand, Netra is also a powerful feature for DDoS detection at the internet layer (According to TCP/IP model [18]).

Data network requires control of traffic at the level of *flows* or *flow aggregates* [19]. A *flow* is defined as follows:

Definition 2.2 (Flow).
At a given router in a local area network (LAN), the collections of packets that travel between the same end points are termed as one flow. [15, 4].

According to Definition 2.2, many flows may coexist at a router a LAN. In the case of a DDoS attacks that are difficult to differentiate from genuine Netra, there exists one flow addressed to the victim system [15, 20].

In [15], discussed a detection algorithm as follows:

- An attacker uses a random variable X to control rate of malicious packets, using a constant as shown Equation 2.5.

$$Pr = \{X = C\} = 1 \tag{2.5}$$

  where:
  - C, is a constant.

- Increasing the number of packets according to attacking time; it is calculated in Equation 2.6

$$X = a * t + b \tag{2.6}$$

  where:

  - a,b, are constants.
  - t, is the time.

- Equation 2.7 shown the Poisson distribution mimicking the Netra pattern.

$$Pr = X = k = \frac{\lambda^k e^{-\lambda}}{k!} \tag{2.7}$$

  where:
  - k = 0,1,...
  - λ, is constant.

- The probability distribution of the flows is depicted in Equation 2.8.

$$p(x_i) = \frac{x_i}{\sum_{j=1}^{n} x_j} \tag{2.8}$$

- Equation 2.9 represents the flow entropy of flows on a given router.

$$H_f(X) = -\sum_{i=1}^{n} p(x_i) log p(x_i) \tag{2.9}$$

- Inequality 2.10 identifies DDoS flooding attacks.

$$H_f(X)\,|_{t=t_0} - H_f(X)|_{t=t_0+\Delta_t} \geq \Delta \tag{2.10}$$

  where:
  - t, represents time
  - $\Delta_t(\Delta_t > 0)$, represents a short time interval.

- A suitable threshold Δ is defined in Equation 2.11

$$H_f^{'}(x) \leq -\Delta \tag{2.11}$$

This algorithm can build a DDoS attacks alarm when flow entropy reduces significantly in a short time interval, in detail [15].

2.3.3 *StopIt* defense mechanisms

In the last few years DDoS attacks have been the subject of various surveys that analyze them from different viewpoints [7, 21, 22, 23]. In particular, the authors of [7] discussed about the growing need of a comprehensive, collaborative and distributed defense approach after they categorized the different DDoS flooding attacks and classified existing countermeasures according to the ability to prevent, detect, and respond to DDoS flooding attacks. In [21],

the coordinated nature of the DDoS attack is explained by recalling that an effective defence strategy for these attacks should also be designed in a collaborative fashion; thus, all the routers need to work collaboratively by exchanging their caveat messages with their neighbors. For this reason, hybrid defense mechanisms are more effective than centralized ones because their components are distributed over multiple locations such as source, destination or intermediate networks by implementing cooperative behaviors among the deployment points.

Few famous hybrid DDoS defense strategies are based on the following mechanisms:

- *Hybrid packet marking* and *throttling/filtering* [24, 25] in which victims under attack install a router throttle at upstream routers several hops away in order to limit the forwarding rate of the packets to their indented. However, these mechanisms only limit the rate of malicious packets and do not harm legitimate flows.
- *Capability-based* [26] in which senders obtain short-term authorizations from the receivers and put them as stamps on their packets. According to this approach, the destination explicitly authorize the traffic it desires to receive.
- *Active Internet Traffic Filtering (AITF)* [27] in which a receiver accepts by default all the traffic and explicitly denies the traffic that has been identified as undesirable. The main drawback of this filtering scheme consists into the requiring of a credible, bounded amount of filtering resources from participating ISPs.
- *StopIt* ([28]b) which will be extensively detailed in the next section to address its main limitation with respect to an indirect DDoS attack.

Since the main contribution of this thesis is the proposal of a new strategy to improve the performance of a widely used hybrid filter-based mechanism named *StopIt* ([28]b), this section summarises the features of this reference defense technique also highlighting few limitations to be addressed in our study. In particular, the *StopIt* technique consists in the installation of a specific network filter able to block the received undesirable traffic by using Passport ([29]a) as secure source authentication system to prevent source address spoofing. It is worth to note that, this technique does not aim at the detection of an attack, but it implements an effective mechanism to face the problem in a quick and completely automatic way.

Figure 2.8 shows the operation of the *StopIt* mechanism applied to a general communication architecture consisting of more Autonomous Systems (AS) connected throughout secure connections in order to avoid the address spoofing; moreover, each specific server who wants to activate the *StopIt* has to be equipped with an algorithm to detect an attack. Each AS owns a *StopIt* server within its domain and all the *StopIt* servers use IGP [30] to exchange data with their AS and BGP [31] to learn the presence of other *StopIt* servers

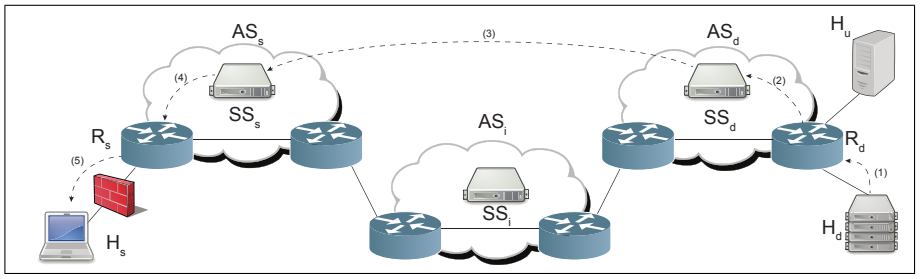


Fig. 2.8. StopIt operation

in the neighbourhood. Going into more details, we briefly recall the operation of the *StopIt* scheme as described in the following steps:

1. The victim host H_d detects the attack and send a request of source blocking toward its access router R_d
2. The access router R_d verifies that the source H_s is really sending data to the server to avoid the situation in which an attacked server uses the *StopIt* mechanism to block legal users; then, it installs a local filter and it sends a request of flow blocking $< source - server >$ to the *StopIt* server SS_d within its own AS
3. Once the *StopIt* server SS_d authenticated the received request, it forwards the request toward the *StopIt* server belonging to the sourcing AS by using the BGP protocol.
4. The *StopIt* server SS_s within the sourcing AS, once received the request, notifies the blocking request to its access router R_s
5. Finally, the access router within the sourcing AS installs the filter to block the flow $< source - server >$ for a certain period and it sends a request to the attacking source. After receiving this request, a compliant host H_s installs a local filter to stop sending to H_d. If H_s does not stop, it will be punished by its own access router R_s.

More details about the operation of the *StopIt* mechanism can be found in ([28]b).

Security aspects

The *StopIt* mechanism previously described supports different security features by addressing the following security issues:

- *Spoofing* - The *StopIt* architecture does not allow to use spoofed addresses because it is coupled with the Passport ([29]a) system to implement a robust inter and intra domain security support.
- *Compromised Server* - To avoid the activation of the *StopIt* mechanism by a compromised server to block a legal use, the access router within the victim's AS has to verify that the source to be blocked is really sending data

traffic; thus, the access router first checks the actual sending of data toward the server in a recent time interval; then, it sends a traffic interruption request to the server by using a specific request message called *End-To-End StopIt* request.

- *Filter exhaustion* - An attacker who attempts to bypass the defence system could aim at saturating the filters available to a router. To limit this strategic attack, the routers have to follow few strategies such as *Filter Aggregation* and *Random Filter Replacement*, to wisely install the filters.

Limitations

In the last years, the *StopIt* mechanism has been validated and tested in different communication scenarios demonstrating its effectiveness respect to various types of massive DDoS flooding attacks; however, it fails when the DDoS attack is achieved in an indirect way by congesting a link shared with the victim. With reference to Figure 2.8, if the DDoS attack is conducted against a host H_u belonging to AS_d, the server H_d within the same AS, experiences a high performance decrease because it shares the same link with the victim.

2.4 Conclusions

The Cyber World is a complex entity, and DDoS attacks are a very important problem. This type of attack is a daily and constant threat to data networks, producing huge losses to organizations. Therefore, an interesting research issue is the development hybrids mechanisms able to identify and face DDoS attacks in real time. Chapter 5 will present the performance of a hybrid filter-based prevention strategy to face of DDoS. The ns-3 simulator is used for the evaluation of the developed technique.

In the chapter 5 will be present the performance of hybrid filter-based prevention strategy to face attackers of DDoS, using ns-3 as a simulation tool.

References

1. D. data, "What is a ddos attack?" 2013. [Online]. Available: http://www.digitalattackmap.com/understanding-ddos/
2. S. Yu, "An Overview of DDoS Attacks," in *Distributed Denial of Service Attack and Defense*. Springer, 2014, pp. 1–12.
3. T. Bienkowski, "5 misconceptions about the modern ddos attack," 2015. [Online]. Available: http://www.slideshare.net/Arbor_Networks/5-d-do-s-misconceptionswebinarslides
4. M.-S. Kim, H.-J. Kong, S.-C. Hong, S.-H. Chung, and J. Hong, "A flow-based method for abnormal network traffic detection," in *Network Operations and Management Symposium, 2004. NOMS 2004. IEEE/IFIP*, vol. 1, April 2004, pp. 599–612 Vol.1.

5. C. Weenk, "Ddos defense mechanisms: a state of the art research," 2007. [Online]. Available: http://referaat.cs.utwente.nl/conference/6/paper/6796/ddos-defense-mechanisms-a-state-of-the-art-research.pdf
6. J. L. Darte, *Identificación de ataques Inforáticos a través de Reyes Bayesianas*. Editorial Académica Española, 2012.
7. S. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 4, pp. 2046–2069, 2013.
8. S. Ranjan, R. Swaminathan, M. Uysal, and E. W. Knightly, "Ddos-resilient scheduling to counter application layer attacks under imperfect detection." in *INFOCOM*. Citeseer, 2006.
9. Neustar, "The danger deepens," 2014. [Online]. Available: https://www.neustar.biz/
10. C. Douligeris and A. Mitrokotsa, "{DDoS} attacks and defense mechanisms: classification and state-of-the-art," *Computer Networks*, vol. 44, no. 5, pp. 643 – 666, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128603004250
11. 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, 2004.
12. S. A. Sunitha M.S, "A low rate ddos: A security threat in mobile adhoc networks," *IJCSNS International Journal of Computer Science and Network Security*, vol. 12, no. 11, pp. 1–3, 2012. [Online]. Available: http://paper.ijcsns.org/07_book/201211/20121101.pdf
13. Y. Xiang, K. Li, and W. Zhou, "Low-rate ddos attacks detection and traceback by using new information metrics," *Information Forensics and Security, IEEE Transactions on*, vol. 6, no. 2, pp. 426–437, 2011.
14. Y. Chen, K. Hwang, and Y.-K. Kwok, "Filtering of shrew ddos attacks in frequency domain," in *Local Computer Networks, 2005. 30th Anniversary. The IEEE Conference on*, Nov 2005, pp. 8 pp.–793.
15. S. Yu, "DDoS Attack Detection," in *Distributed Denial of Service Attack and Defense*. Springer, 2014, pp. 31–40.
16. Y. Chen and K. Hwang, "Collaborative detection and filtering of shrew ddos attacks using spectral analysis," *Journal of Parallel and Distributed Computing*, vol. 66, no. 9, pp. 1137–1151, 2006.
17. P. N. Jadhav and B. Patil, "Low-rate ddos attack detection using optimal objective entropy method," *International Journal of Computer Applications (0975 8887)*, vol. 78, no. 3, pp. 33–38, 2013. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.402.6134&rep=rep1&type=pdf
18. A. S. Tanenbaum and D. J. Wetherall, *Computer Networks: Pearson New International Edition: University of Hertfordshire*. Pearson Higher Ed, 2013.
19. S. Natarajan, X. Huang, and T. Wolf, "Efficient conflict detection in flow-based virtualized networks," in *Computing, Networking and Communications (ICNC), 2012 International Conference on*, Jan 2012, pp. 690–696.
20. X. Ma and Y. Chen, "Ddos detection method based on chaos analysis of network traffic entropy," *Communications Letters, IEEE*, vol. 18, no. 1, pp. 114–117, January 2014.
21. P. Kumar and S. Selvakumar, "Distributed denial-of-service (DDoS) threat in collaborative environment - A survey on DDoS attack tools and traceback mech-

anisms," in *Advance Computing Conference, 2009. IACC 2009. IEEE International*, March 2009, pp. 1275–1280.
22. T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," *ACM Comput. Surv.*, vol. 39, no. 1, Apr. 2007.
23. V. Thing, M. Sloman, and N. Dulay, "A survey of bots used for distributed denial of service attacks," in *New Approaches for Security, Privacy and Trust in Complex Environments*, ser. IFIP International Federation for Information Processing, H. Venter, M. Eloff, L. Labuschagne, J. Eloff, and R. Solms, Eds. Springer US, 2007, vol. 232, pp. 229–240.
24. R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker, "Controlling high bandwidth aggregates in the network," *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 3, pp. 62–73, Jul. 2002.
25. R. Chen, J.-M. Park, and R. Marchany, "Track: A novel approach for defending against distributed denial-of-service attacks," Technical Report TR-ECE-06-02, Dept. of Electrical and Computer Engineering, Virginia Tech, Tech. Rep., 2006.
26. T. Anderson, T. Roscoe, and D. Wetherall, "Preventing internet denial-of-service with capabilities," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 1, pp. 39–44, Jan. 2004.
27. K. Argyraki and D. R. Cheriton, "Scalable network-layer defense against internet bandwidth-flooding attacks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 4, pp. 1284–1297, Aug. 2009.
28. X. Liu, X. Yang, and Y. Lu, "To filter or to authorize: Network-layer DoS defense against multimillion-node botnets," in *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*, ser. SIGCOMM '08. New York, NY, USA: ACM, 2008, pp. 195–206.
29. X. Liu, A. Li, X. Yang, and D. Wetherall, "Passport: Secure and adoptable source authentication," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, ser. NSDI'08. Berkeley, CA, USA: USENIX Association, 2008, pp. 365–378.
30. S. Sendra, P. A. Fernández, M. A. Quilez, and J. Lloret, "Study and performance of interior gateway ip routing protocols," *Network Protocols and Algorithms*, vol. 2, no. 4, pp. 88–117, 2010.
31. D. Medhi and K. Ramasamy, *Network Routing: algorithms, protocols and architectures*, M. Kaufmann, Ed., 2007.

Part II

DDoS: Simulation Experiments

3

Modeling and Simulation of Internet and Application Layer DDoS Attacks

Abstract

Nowadays, many kinds of transactions are made through the Internet, and this establishes strong requirements in order to warranty the security of information systems against different types of threats and attacks. DDoS is one of the most effective attack techniques, which is achieved by exploiting a network of controlled hosts (botnet). The attacker remotely governs these computers and instructs them to send malicious traffic towards the victim. A DDoS can be performed an internet, transport and/or application layer. DDoS attacks that target the application layer are very sophisticated and dangerous. An application layer DDoS attack is difficult to detect and mitigate because it attacks specific resources mimicking the normal use of an application. This chapter analyzes the results obtained through the simulation of internet and application layer DDoS attacks, targeted to a Domain Name System (DNS) server, which has been achieved with NeSSi^2.

3.1 Introduction

Due to the advancement in technology and excessive growing networks, homes, schools, companies, and/or governments are exposed to frequent attacks around the world, which evidently indicates the necessity to protect the information in a network. While the advances in technology permit more efficient types of hardware and software, simultaneously hackers have enhanced their skills to threaten networks, this fact establishes security as a crucial aspect to have in mind. Today, DDoS attacks are one of the greatest and most sophisticated threats to cyber security; this technique is a coordinated attack on the availability of services of a single or multiple victim systems through a network of controlled hosts (botnet) [1, 2]. These attacks can be countered at different locations in the data network. Each attack stream flows out of a

machine, through a server or router into the net, into the server, router, or firewall that controls access to the target machine's net; and finally to the target itself [3]. DDoS attacks can be classified into two categories: flooding attacks and vulnerability attacks. Flooding DDoS attacks consume resources such as network bandwidth by overwhelming a bottleneck link with a huge volume of packets. Vulnerability attacks use the expected behavior of protocols such as UDP to the attackers advantage [4]. According to Arbor Networks enterprise recorded network attacks and stated that during the first three months of 2015 there were 25 over 100 Gps through the Internet; from April to June 2015, DDoS attacks were reported over UDP ports with 196GB/s; Arbor also reported that 21% attacks consisted of at least 1 GB/s. This draws complicated scenarios for detecting and mitigating attacks in smalls and mediums enterprises. The average cost of these kind of attacks is $ 40,000 per hour. As reported by Akamais [state of the internet], in Q1 2015, China is the country that generated 23.45% of DDoS attacks in the world, Germany is in second place with 17.39%, and the United States recorded 12.18% of total threat traffic.

The TCP/IP reference model defines four conceptual layers in a communications network [5]. DDoS attacks mainly exploit three of these layers: internet, transport, and application. In the case of internet layer DDoS attacks, which are called IntDDoS attacks in this chapter; the majority of DDoS attacks target the internet and transport layers. Such attacks occur when the amount of data packets and other traffic overloads a victim system and consumes all available upstream and downstream bandwidth to prevent access to websites. These attacks generally include UDP floods, which are used frequently for larger bandwidth DDoS attacks because they are connectionless and it is easy to generate protocol 17 (UDP) messages from many different scripting and compiled languages. In the case of application layer DDoS attacks, which are called AppDDoS attacks in this chapter; there is vulnerability in an application. By exploiting it, the attackers overwhelm the server or database powering a web application. Such attacks mimic legitimate user traffic, making them harder to detect. Application requests overwhelm the web server or database causing it to crash, then the website subsequently becomes unavailable [6, 7]. These attacks may cause more severe problems in high-speed Internet than in the past [8]. IntDDoS and AppDDoS attacks, which are legitimate in protocols and packets, have become a big problem for the cyberworld; these attacks may be stealthier and most dangerous for the data network, hence, this chapter analyzes the results obtained through the modeling and simulating of internet and Application layer DDoS attack, targeted to a Domain Name System (DNS) server, which has been achieved with NeSSi^2.

The organization of the chapter is as follows: Section 3.2 describes internet and application layer DDoS attacks. Section 3.4 describe brief NeSSi^2 Network Simulator. Section 3.5 provides a detail of the network attack model.

Section 3.6 shows the obtained experiments results. Finally, the conclusions are presented together with an indication of future work in Section 3.7.

3.2 IntDDoS and AppDDoS Attacks

Internet and transport layer DDoS attacks are types of volumetric DDoS attacks on a network infrastructure. IntDDoS attacks rely on extremely high volumes (floods) of data to slow down server performance, consume the network bandwidth and loss of access for legitimate users of the victim systems. IntDDoS attack types generally include SYN flooding, ICMP flooding, and UDP flooding [8, 9]. Figure 3.1 shows a internet and transport layer DDoS attack.

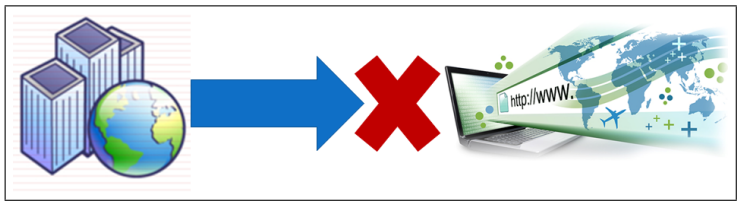


Fig. 3.1. internet Layer Attack

User Datagram Protocol (UDP) is a connectionless protocol used for a variety of applications such as DNS, SNMP, VOIP, video streaming, and DHCP. Attackers are able to send a great number of packets since the UDP does not have a congestion control mechanism [10]. "UDP-based attack occurs when the attackers send a large number of UDP packets to random ports to saturate the target. The attackers can also spoof the IP of the attack packets for two purposes: first to ensure that the excessive reply packets do not reach them; second to keep their network location anonymous. When the victim receives the UDP packets with spoofed source address, it checks for the application listening at that port. Since there is no application listens at that port, it replies with an ICMP destination unreachable which amplifies the attack. These attacks are potentially severe and can dramatically bring down the business of companies"[10]. Flooding attacks are very similar to internet Layer DDoS attacks in that the attacker and his zombies send a huge number of packets. However, instead of consuming the resources of a whole server as in internet layer DDoS, the target is an application running on the server, such as a DNS Server. When the attack is carried out, the application is overwhelmed and is no longer available to genuine clients [11]. The vulnerability is caused by a malicious user who utilizes the data without the genuine users prior permission [12].

DDoS attacks, generally are carried out at the internet layer. However, there is evidence to suggest that AppDDoS attacks can be more effective and dangerous than traditional ones [13].

AppDDoS attacks are the most dangerous as they can be used to attack a particular application of the server, such as DNS Servers, routers, or web clusters [14]. This attack is an attack structured to overload specific elements of an application server infrastructure; these attacks are especially complex, stealthy, and difficult to detect because they resemble legitimate website traffic. Even simple AppDDoS attacks, for example those targeting login pages with random user IDs and passwords, or repetitive random searches on dynamic websites can critically overload CPUs and databases. Also, AppDDoS attackers can randomize or repeatedly change the signatures of an attack, making it more difficult to detect and mitigate [9]. The CyberSlam and the MyDoom worm are instances of AppDDoS attacks [8]. Figure 3.2 shows a application layer DDoS attack.

These attacks may focus on exhausting the server resources by consuming the network bandwidth, the CPU cycles, by consuming the RAM memory, Sockets, disk/database bandwidth, and I/O bandwidth. Research has found, with increasing computational complexity in Internet applications and larger network bandwidth, that server resources may become the bottleneck of those applications [8].

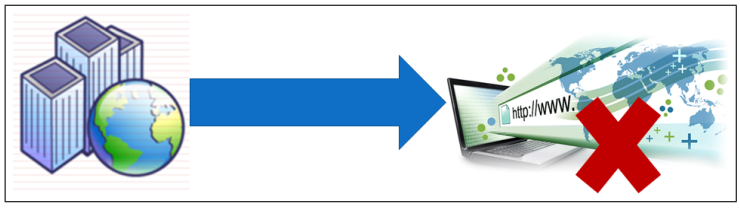


Fig. 3.2. Application Layer Attack

3.3 Domain Name System (DNS) server

According to the TCP/IP reference model, the application layer includes the Domain Name System (DNS), for mapping host names onto their network addresses [5]. DNS is a distributed database for storing information on domain names [15], and can be used to perform refreshing of IP addresses and to convert between an IP address and a domain name [16].

A Finite State Automata (FSA) was used to model the behavior of the DNS server.

3.4 NeSSi² Network Simulator

Definition 3.1 (NeSSi²). *Network Security Simulator [17] is a discrete event and agent-based network simulator with a focus on security-based research.*

A simulation environment simplifies the study of a problem and allows researchers and developers to concentrate on the most critical issues [18]. NeSSi² has been structured on a three-tier plugin architecture: the Graphical Frontend, the agent-based simulation Backend and the Result Database.

The graphical frontend of NeSSi² allows to create and edit the components of a network simulation. It can be used to create and modify network topologies and specify the behavior of network elements. It also allows to start a simulation. Network modeling is achieved very simply and friendly because it is done by dragging and dropping the components. The GUI allows the real-time inspection of the network during the simulation.

The NeSSi² Backend is the component that actually carries out the simulations. It is responsible to create and setup the simulation environment. It is directly connected with underlying database layer and takes care of recording the relevant simulation data.

Simulation results are stored in a database so that them can be used for further evaluation. It is also possible to track the traffic generated on the network by creating a suitable `pcap` file for each node.

The topology describes the static information contained in the network but the most important feature of NeSSi² is the possibility to define the network behavior properly. The first step to realize a simulation consists in the association of a profile to each node. It is possible to use the pre-built profiles (such as server, client, router, firewall, etc.) or to introduce new custom applications. The second step consists in the setup of a scenario where behavior definitions are tied to a particular network topology. This feature is important because one could be interested to evaluate the same network under different circumstances. This can be done by creating several simulation sessions involving the same network where different scenario are deployed. In the third step a simulation session, which is characterized by additional and specific information for the selected scenario (e.g., the times a simulation should be repeated, the amount of ticks, etc.), is created and executed. NeSSi² offers an extensive support for constructing complex applications-level scenarios based on a simulated TCP/IP protocol stack (it also works with UDP).

3.5 Modeling and Simulation DDoS Attacks

The attack network model includes aspects of the network configuration relevant to IntDDoS and AppDDoS layer attack, as well as a set of attributes of the configuration. It is assumed that the attacker has commandeered a

number n of hosts distributed across a wide-range of geographical areas, organized into server farms popularly known as malicious networks (botnets) (see Section 1.3). Besides, was considered session-oriented connectionless to the server, session on a UDP connection with the DNS server. This model includes elements and attributes for modeling a network attack.

Figure 3.3, displays the network topology that was implemented using by NeSSi^2 as network simulator tool, in details Section 3.4. The attack was conducted with UDP flood to the DNS server, where the attacker overwhelmed ports random of a specific host with IP packets that contained UDP datagrams. To evaluate the behaviors of the attack have been introduced, 1 DNS Server and 11 subnets, with link bandwidths of 1Gbit. On the other hand, the goal of the botnet was to saturate network bandwidth to its maximum limit so that the DNS server capacity for serving clients was diminished.

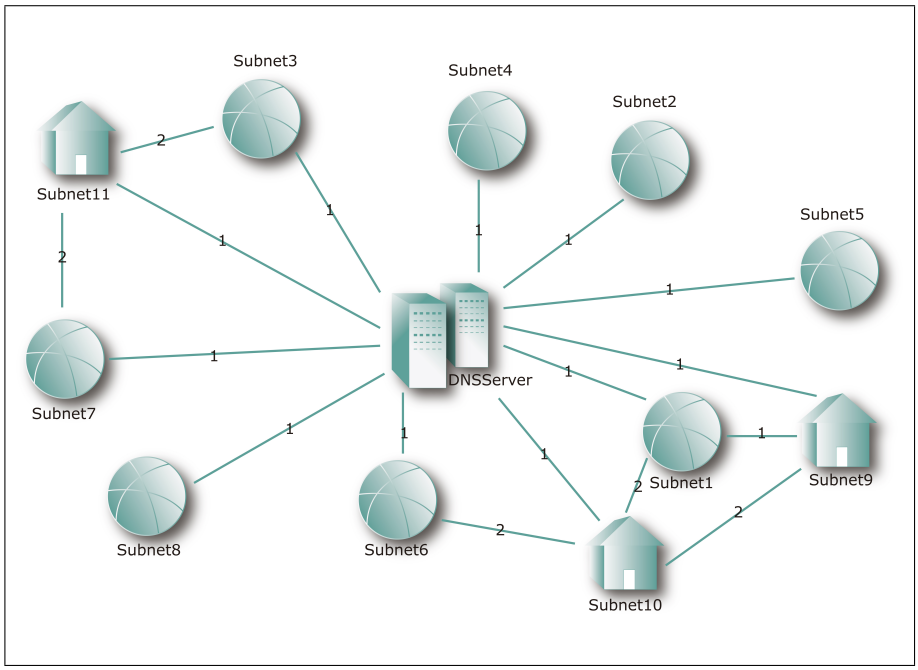


Fig. 3.3. Network Topology

Four new entities were introduced to model the attack scenario: *Client, Bot, Zombie*, and *DNS Server*.

3.5.1 Client

The *Client* entity was implemented to fulfill basically two tasks:

1. Send and receive "echo" requests , and
2. Send requests to the DNS server.

These behaviors are reflected in two profiles. For the purposes of the first task, it is necessary to provide general information, which often appears in almost all network entities, one of which is the creation of an IP address, which can be requested in two fundamental ways: One way is through the user interface of NeSSi^2, implementing the code as shown in Figure 3.4, and finally, it is created through `IPFactory` (see Figure 3.5).

```
@StringOption(value = "Destination IP")
@Description(value = "IP address of the attacked host")
private String destinationAddr;
destinationIP = IPFactory.
createIPv4Address(destinationAddr);
```

Fig. 3.4. Creation of an IP address

```
destinationIP = IPFactory.
createIPv4Address(destinationAddr);
```

Fig. 3.5. `IPFactory`

Another way is by invoking the `getNetworkMetaInf() method`, which provides a list of all the addresses of a certain type of network through `getIPLIstForApplication (ApplicationType.class)`, or in turn, depending on their needs, which is provided with random `getRandomIPAddressForApplication (ApplicationType.class)`.

The second task, i.e. requests sent by the client to DNS server were implemented through the following methods:

- `handleEvent(Event e)`, and
- `start()`.

The event type is the default on NeSSi^2 is `TickedEvent`, the number of successive ticks through the code is added to the `start()` method for the purposes of the simulation, as shown in Figure 3.6.

Additionally, the start() method must registers ports, in this case, UDP, in order to ensure that the application is able to receive packets it was also necessary to configure the next tick, the same that was implemented in the code shown in Figure 3.7.

```
getLayerHandler().
addEvent(new TickedEvent(this, 5));
```

Fig. 3.6. TicketEvent

```
TickedEventtickedEvent=(TickedEvent) e;
tickedEvent.setTick(tickedEvent.getTick()+30);
handler.addEvent(tickedEvent);
```

Fig. 3.7. Next tick

Another important aspect is to record events in the database to evaluate further the statistics offered by the tool. The code implemented for this purpose is shown in Figure 3.8.

```
int tick = handler.getCurrentTick();
IPDevice device = handler.getDevice();
HbmRecordingInteventRec = EventFactory.
getIntegerEvent(tick, device,
EventLabels.REQUESTS_SERVICE, 1);
IRecorder recorder =
getRecorder(EventLabels.class);
recorder.record(eventRec);
```

Fig. 3.8. Event Record

3.5.2 Bot

The task of the *Bot* entity was to recruit zombie computers in a few ticks to start the attack. The attacker remotely governs these computers and instructs them to send malicious traffic towards the victim (DNS Server). The created zombie botnet then simply attacks the victim with a large amount of traffic. The higher the number of zombies, the greater the impact of the attack. The `Bot` class reproduces the behavior of a Bot, its was extended from the `AbstractIPApplication` class as depicted in Figure 3.9.

3.5.3 Zombie

The activity of the *Zombie* is to recognize a packet containing special code or a specific number, which will be interpreted as a sign of attack to start sending requests to the destination port of the DNS server, that is, port

```
public class Bot extends AbstractIPApplication {
@StringOption(value="IP Victim")
@Description(value="IP Address")
private String destinationAddr;
private IPv4Address destinationIP;
@IntegerOption("Start")
@Description("Tick")
private int tickStart ;

public Bot(){
  destinationAddr = "0.0.0.0";
  tickStart=10;
}//Bot

@Override
public boolean start()
{
  destinationIP = IPFactory
    .createIPv4Address(destinationAddr);
  this.getLayerHandler().getTransportLayer()
    .registerUDPPort(8, this);
  getLayerHandler().addEvent(new TickedEvent(this,3));
  return super.start();
}//start

@Override
public void handleEvent(Event event) {
  if (!(event instanceof TickedEvent)) {
    return;
  }//if
  String start="VR "+tickStart;
  getLayerHandler().getTransportLayer().sendUDPPacket(7, 7,
  destinationIP, start.getBytes());
  int tick= getLayerHandler().getCurrentTick();
  IPDevice device=getLayerHandler().getDevice();
  HbmRecordingInt eventRec = EventFactory
    .getIntegerEvent(tick,
  device, ddos.ddos.EventLabels.RESEARCH, 1);
  IRecorder recorder= getRecorder(EventLabels.class);
  recorder.record(eventRec);
}//handleEvent
}//class
```

Fig. 3.9. `Bot` class implemented

53. Zombie hosts were innocently recruited from hundreds of unprotected computers accessing the Internet through high-bandwidth and always avail-

able connections. The attack takes place every time the event is planned and when the packet is received. The `Zombie` class was extended from the `AbstractIPApplication` class as depicted from Figure 3.10 to Figure 3.12. The menacing zombie activity was coded in the `handleIncomingPacket(IPacket p)` method, as shown in Figure 3.11.

```
public class Zombie extends AbstractIPApplication {
@IntegerOption("Client Port")
@Description("Port")
private int echoClientPort = 7;
@StringOption(value = "Destination IP")
@Description(value = "IP Address")
private String destinationAddr;
private IPv4Address destinationIP;
private int destinationPort;
private int tickAttack;
@IntegerOption("Frequency")
@Description("Frequency")
private int frequency = 3;

public Zombie() {
  destinationAddr = "0.0.0.0";
}//Zombie

public boolean start() {
  this.getLayerHandler().getTransportLayer()
    .registerUDPPort(this.echoClientPort, this);
  getLayerHandler().addEvent(new TickedEvent(this, 10));
  destinationIP = IPFactory
    .createIPv4Address(destinationAddr);
  IPDevice device = getLayerHandler().getDevice();
  HbmRecordingInt eventRec = EventFactory
    .getIntegerEvent(0, device,
  ddos.ddos.EventLabels.START_CLIENT, 1);
  IRecorder recorder = getRecorder(EventLabels.class);
  recorder.record(eventRec);
  return super.start();
}//start
...
}//class
```

Fig. 3.10. `Zombie` class implemented

```
@Override
public void handleIncomingPacket(IPacket p) {
  if (!(p instanceof Packet)) {
    return;}
  Packet packet = (Packet) p;
  IPv4Header ipHeader = packet.findHeader(IPv4Header.class);
  UDPHeader udpHeader = packet.findHeader(UDPHeader.class);
  if (ipHeader == null || udpHeader == null) {
    return;}
  String attack = new String(packet.getPayload());
  String match = "[VR]+\\s[0-9]*";
  if (Pattern.matches(match, attack)) {
    StringTokenizer st = new StringTokenizer(attack);
    st.nextToken();
    attack = st.nextToken();
    int att = Integer.parseInt(attack);
    if (att != 0) {
      destinationPort = 53;
      tickAttack = att;}//if
  }//if
  int tick = getLayerHandler().getCurrentTick();
  if (tickAttack <= tick) {
    destinationPort = 53;
    ITransportLayer transportLayer = this.getLayerHandler()
      .getTransportLayer();
    transportLayer.sendUDPPacket(echoClientPort,
      destinationPort,
    destinationIP, "VRZombie.it".getBytes());
    IPDevice device = getLayerHandler().getDevice();
    HbmRecordingInt eventRec = EventFactory.
      getIntegerEvent(tick,
    device, ddos.ddos.EventLabels.ATTACK_ZOMBIE, 1);
    IRecorder recorder = getRecorder(EventLabels.class);
    recorder.record(eventRec);
  }//if
}//handleIncomingPacket
```

Fig. 3.11. `handleIncomingPacket` function implemented

```
@Override
public void handleEvent(Event event) {
  if (!(event instanceof TickedEvent)) {
    return;
  }//if
  TickedEvent e = (TickedEvent) event;
  int tick = getLayerHandler().getCurrentTick();
  if (tickAttack <= tick) {
    destinationPort=53;
    ITransportLayer transportLayer = this.getLayerHandler()
      .getTransportLayer();
    transportLayer.sendUDPPacket(echoClientPort,
      destinationPort,
    destinationIP, "VRZombie.it".getBytes());
    IPDevice device = getLayerHandler().getDevice();
    HbmRecordingInt eventRec = EventFactory
      .getIntegerEvent(tick,
    device, ddos.ddos.EventLabels.ATTACCO_ZOMBIE, 1);
    IRecorder recorder = getRecorder(EventLabels.class);
    recorder.record(eventRec);
    e.setTick(getLayerHandler()
      .getCurrentTick() + frequenza);
    getLayerHandler().addEvent(e);
  }//if
}//handleEvent
```

Fig. 3.12. `handleEvent` function implemented

3.5.4 DNS Server

The DNS server, described in Section 3.3, whose behavior is specified by the finite state automaton shown in Fig. 3.13, is modelled by the `DNSServer` class as a multithreaded application which is able to process up to n requests in parallel. In particular, if the DNS server is in the *Available* state, it handles incoming requests as soon as they arrive; on the contrary, when there are no more available processing resources, the server switches to the *Busy* state and stores the incoming requests into a limited buffer. If the buffer reaches capacity, incoming requests are dropped. The service time is assumed to follow an exponential distribution (3.1) with mean 5 ms.

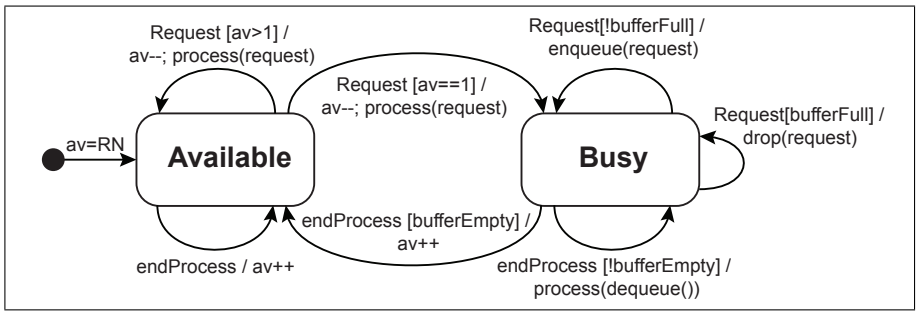


Fig. 3.13. Server behavior

$$f(x,\lambda) = \begin{cases} \lambda e^{-\lambda x} & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases} \tag{3.1}$$

3.6 Results

The statistics were generated by JFreeChart, which is part of the NeSSi^2 network simulator.

Figures from 3.14 to 3.16 show that the client sends requests to the DNS server regularly, and the answer is normally received thus demonstrating the servers' ability to perform their functions. Figure 3.17 depicts the attack executed by the botnet to the DNS server. In this figure, it is possible to observe when a Bot has attacked the DNS server without the latter having direct contact with the server. The number of networks that the botnet recruited to carry out such an attack corresponds to a total of 11 subnets. On the other hand, the attack was exploited protocol inconsistencies to render DNS server inaccessible.

The client has also sent its requests to the DNS server that was controlled by Zombie for excessive packets forwarding. Figure 3.18 shows the effects of

the attack made by the Zombies to the DNS Server; the black fringe represents client's requests, and the pink fringe represents the attack in progress.

Intddos and Appddos layer attacks mimic legitimate clients and overwhelm the network bandwidth, thereby, substantially delaying or denying service to the genuine/authorized clients.

In the first period normal operation of the DNS server was observed, but after that period the server was saturated and began to discard and block packets. The DNS server was implemented so that it can block services for a moment until the buffer is released, but the buffer remains full, and then the server is out of service.

Additionally, it was observed that the server was saturated then its could not continue providing services so that packets were blocked and discarded, as shown Figure 3.19.

Besides, Figure 3.20 allows to determine the type and number of packets that arrived at the DNS server, so it was possible to distinguish regular packets and malicious packets, the last used to saturate the server and leave the server out of service.

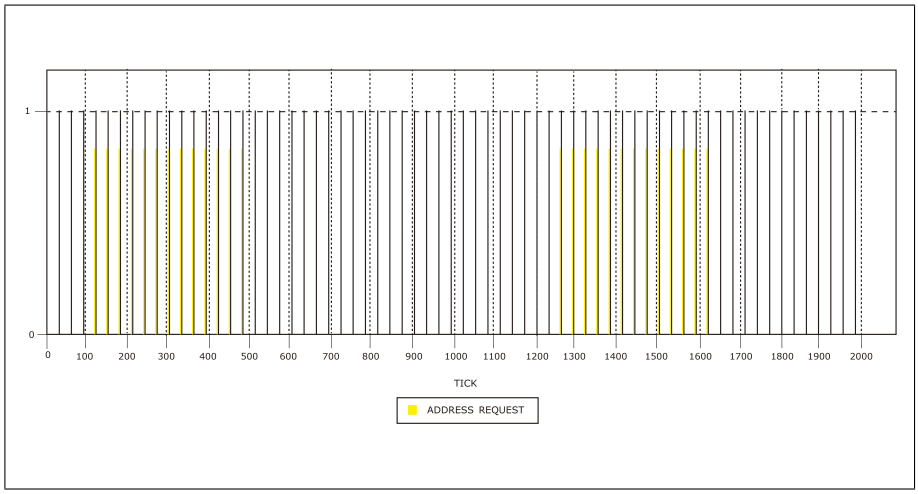


Fig. 3.14. Request client

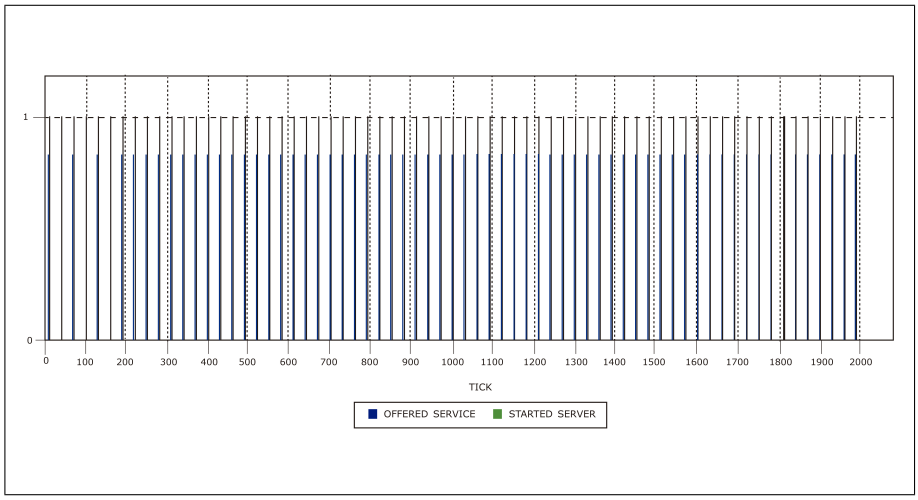


Fig. 3.15. Server responses

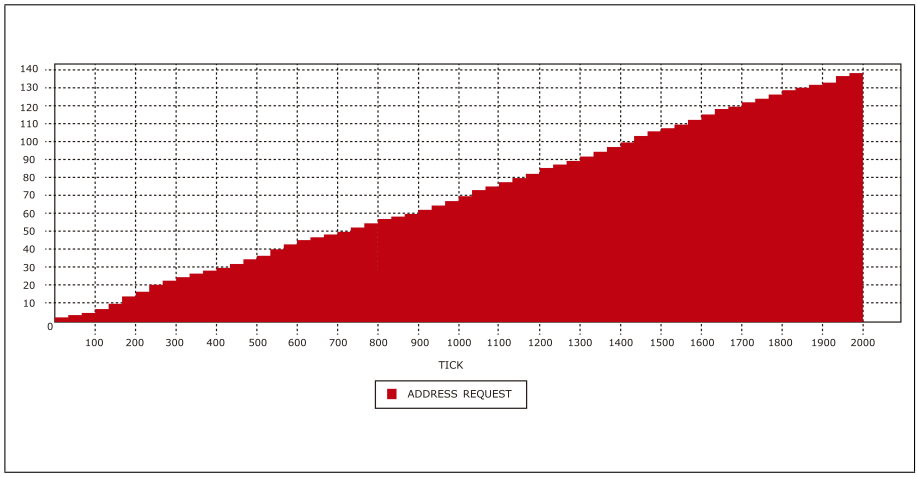


Fig. 3.16. Cumulative requests.

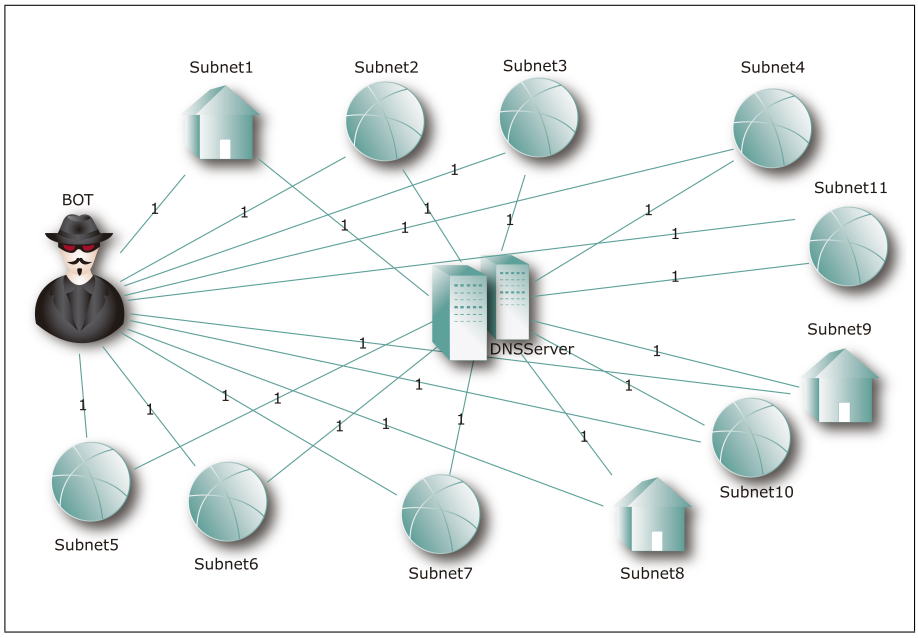


Fig. 3.17. Attacked Network Topology

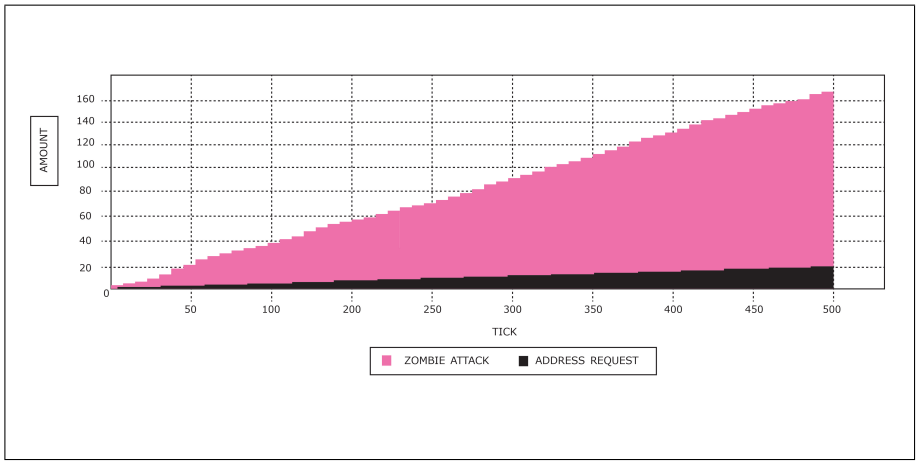


Fig. 3.18. Requests sent to DNS server

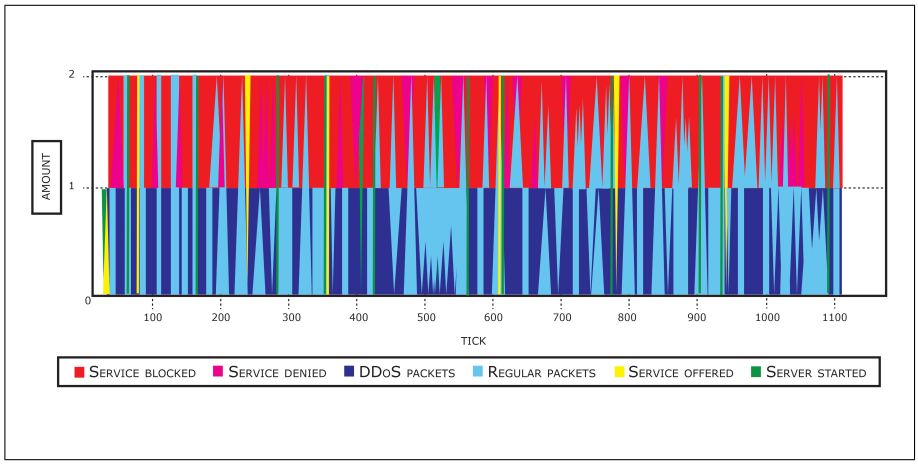


Fig. 3.19. Sever under attack

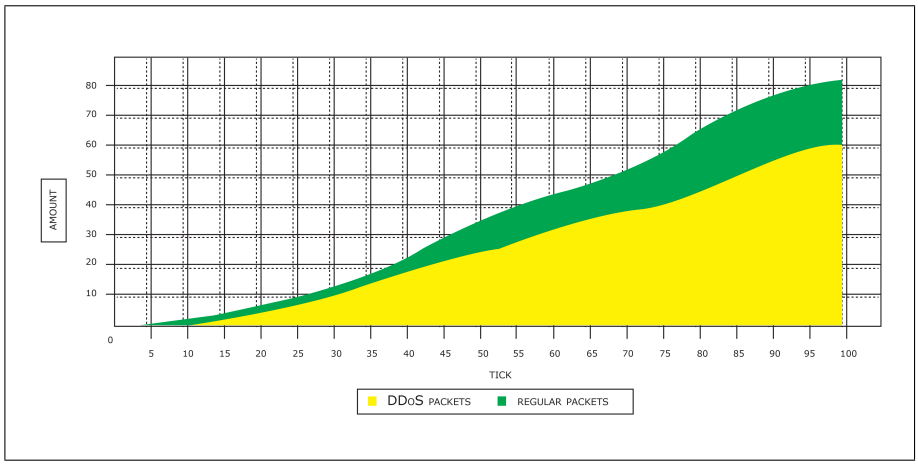


Fig. 3.20. DDoS Packets and Regular Packets

3.7 Conclusions

NeSSi^2 is a tool that can simulate scenarios using TCP/IP stack protocols, in this context, a topology developed in the form of a UDP Flood together with the worm attack technique. The latter had the task of bombing the victim (DNS server). Application layer DDoS attacks (AppDDoS) are very dangerous and extremely difficult to detect due to the huge number of hosts participating in the attack; AppDDos attacks generated too much traffic toward the victim and as a consequence services remain out of the order. Having modeled and simulated scenarios of DDoS attacks using a botnet, the results allowed for quantification of the amount of User Datagram encapsulated in IP packets addressed to the DNS server, bringing the dysfunctional state in which he found himself immersed is checked the server when it was under the DDoS attack.

It is necessary to analyze the behavior of data network components through modeling real detections and defense with their respective simulations; how to prevent and mitigate DDos attacks will be described in Chapter 6.

References

1. S. Specht and R. Lee, "Taxonomies of distributed denial of service networks, attacks, tools and countermeasures," Technical Report CE-L2003-03, Princeton University, http://www. princeton. edu/˜ rblee/ELE572_ F04Readings. html, Tech. Rep., 2003.
2. L. Qin, L. Hong, K. Songlin, and L. Chuchu, "Feature extraction and construction of application layer ddos attack based on user behavior," in *Control Conference (CCC), 2014 33rd Chinese*, July 2014, pp. 5492–5497.
3. J. Yu, Z. Li, H. Chen, and X. Chen, "A detection and offense mechanism to defend against application layer ddos attacks," in *Networking and Services, 2007. ICNS. Third International Conference on*, June 2007, pp. 54–54.
4. S. B. Ankali and D. Ashoka, "Detection architecture of application layer ddos attack for internet," *Int. J. Advanced Networking and Applications*, vol. 3, no. 01, pp. 984–990, 2011.
5. A. S. Tanenbaum and D. J. Wetherall, *Computer Networks: Pearson New International Edition: University of Hertfordshire.* Pearson Higher Ed, 2013.
6. IMPERVA, "The top 10 ddos attack trends," 2015. [Online]. Available: https://www.imperva.com/docs/DS_Incapsula_The_Top_10_DDoS_Attack_Trends_ebook.pdf
7. N. Cybersecurity and C. I. Center, "Ddos quick guide," 2014. [Online]. Available: https://www.us-cert.gov/sites/default/files/publications/DDoS%20Quick%20Guide.pdf
8. Y. Xie and S. zheng Yu, "Monitoring the application-layer ddos attacks for popular websites," *Networking, IEEE/ACM Transactions on*, vol. 17, no. 1, pp. 15–25, Feb 2009.
9. Akamai, "Types of ddos attacks. an explanation of many types of ddos attacks." [Online]. Available: https://www.stateoftheinternet.com/types-of-ddos-attacks.html

10. M. Malekzadeh, M. Beiruti, and M. S. Abadi, "Assessment of high and low rate protocol-based attacks on ethernet networks," *Assessment*, vol. 6, no. 7, 2015.
11. Y. Dantas, V. Nigam, and I. Fonseca, "A selective defense for application layer ddos attacks," in *Intelligence and Security Informatics Conference (JISIC), 2014 IEEE Joint*, Sept 2014, pp. 75–82.
12. S. Khan and A. Saxena, "Detecting input validation attacks in web application," *International Journal of Computer Applications*, vol. 109, no. 6, pp. 1–4, 2015.
13. V. Durcekova, L. Schwartz, and N. Shahmehri, "Sophisticated denial of service attacks aimed at application layer," in *ELEKTRO, 2012*, May 2012, pp. 55–60.
14. S. Ranjan, R. Swaminathan, M. Uysal, and E. W. Knightly, "Ddos-resilient scheduling to counter application layer attacks under imperfect detection." in *INFOCOM*. Citeseer, 2006.
15. M. Radwan and R. Heckel, "Detecting and refactoring operational smells within the domain name system," *arXiv preprint arXiv:1504.02615*, 2015.
16. S. Bao, J. Chen, W. Jiang, Z. Su, X. Yang, and J. Zhang, "Providing a virtual domain name system (dns) in a local area network (lan)," Mar. 24 2015, uS Patent 8,989,182. [Online]. Available: https://www.google.com/patents/US8989182
17. NeSSi^2, "Network security simulator," 2015. [Online]. Available: http://www.nessi2.de/
18. E. Lochin, T. Prennou, and L. Dairaine, "When should I use network emulation?" *annals of telecommunications - annales des télécommunications*, vol. 67, no. 5-6, pp. 247–255, 2012.

4

A Simulation Model of DDoS Amplification Attacks

Abstract

DDoS attacks are increasing both in magnitude and sophistication. These types of attacks reached a new high-water mark in terms of scale by using amplification and reflection techniques. Amplification attacks involve an answer that is bigger than the original request, and Reflection attacks involve sending requests with spoofed source IP addresses to servers causing the unwanted response traffic to the spoofed addresses instead of the real senders. This chapter proposes a simulation model for the analysis two techniques adopted together in order to increase the effect of a DDoS attack, i.e. amplification and reflection, that can exploit Network Time Protocol (NTP) or Domain Name System (DNS) protocols. The developed model, which has been implemented within NeSSi^2, allowed a comparison of the effects of the attack under the two scenarios.

4.1 Introduction

Nowadays network security is threatened by various types of cyber attacks. Among these, the most common and effective one is the Distributed Denial of Service (DDoS) [1]. A DDoS attack is a malicious attempt to coordinate the behaviors of several hosts in order for them to flood a victim host with a high number of packets, so that its bandwidth and/or its computational resources are exhausted [2]. To perform the first task, the attacker tries to generate a huge volume of network traffic in order to flood the link that connects the victim with the network. This type of DDoS operates at the internet level of the TCP/IP stack. The second type of attack is instead performed at application or session levels. During a DDoS attack, the aggressor orchestrates the behavior of a network (*botnet*) of slave hosts (*zombies*), which have previously been infected by a malicious code, through messages that indicate a specific victim to be flooded [3].

This chapter is focused on two techniques adopted together in order to increase the effect of a DDoS attack, i.e. *reflection* and *amplification*. A reflection-based DDoS attack exploits the IP spoofing vulnerability. The attacker sends a network request, with a forged IP source address (i.e. the victim's address) to a network entity (usually a server) which is exploited as a *reflector* [4], i.e. it will direct its responses to the fake source address. This technique can be performed both on TCP and UDP protocols. In the case of the TCP this reflection attack is also known as SYN-ACK or SYN-flood [5]. When executed over UDP, a typical attack employs a DNS server as reflector.

The impact of reflection attacks can be augmented when the size of the response generated by the reflector is much bigger than that of the request. Such an effect is referred to as *amplification*. Usually, DDoS amplification attacks are performed on connectionless protocols.

A particular amplification technique, the so called *smurf* attack [6], exploits a misconfiguration of a local network that allows the use of broadcast messages and can be used with any protocol. DNS and Network Time Protocol (NTP) are among the most frequently used UDP-based protocols employed to perform amplification/reflection DDoS attacks [7]

This experiment describes an agent-based simulation model whose aim is to provide a tool for the evaluation of the effects of DDoS amplification attacks under different scenarios. The model has been developed within a NeSSi^2 simulator [8] and can be exploited for assessing the effectiveness of defense/reaction techniques against such type of cyber attacks.

The chapter is organized in four parts. Section 4.2 describes DDoS amplification attacks and in particular those based on DNS and NTP. Section 4.3 presents the simulation model that was developed. Section 4.4 shows the evaluation of two attack scenarios considering respectively DNS and NTP protocols. Finally, Section 4.5 concludes the chapter.

4.2 DDoS Amplification/Reflection Attacks

In its basic form, a DDoS attack generates a huge volume of traffic by exploiting a botnet of zombies controlled by one master node. However this scheme can be improved by using reflection and amplification techniques.

The aim of a distributed reflective DDoS attack is to overwhelm the victim's bandwidth. The attacker resorts to a false identity to send traffic to the victim by exploiting several amplifiers that work in parallel and this leads to a highly distributed denial-of-service attack. As a consequence, a high volume of traffic is generated from the reflectors towards the victim because they believe that the victim is the host that asked for it [7, 9]. The basic scheme of an amplification/reflection DDoS attack is shown in Fig. 4.1.

Two indices have been identified for evaluating the impact of the amplification: Bandwidth Amplification Factor (BAF) and Packet Amplification Factor (PAF) [7]. BAF is the ratio between the size of the payload sent from

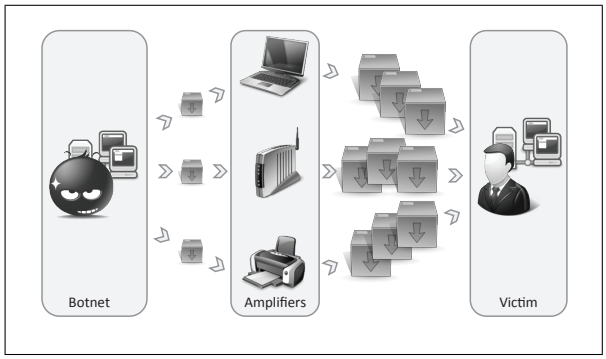


Fig. 4.1. Amplification/reflection DDoS attack schema [7]

the amplifier to the victim and the size of the payload of the packet sent to the amplifier by the attacker. Analogously, PAF is the ratio between the number of packets sent from the amplifier to the victim and the number of packets sent to the amplifier by the attacker.

4.2.1 DNS Amplification Attack

Domain Name System (DNS), is a hierarchical distributed system devoted to resolve the names of network hosts. This protocol is defined in RFC 1034 [10] and RFC 1035 [11] and uses UDP as its transport layer over the default port 53. The packet format is the same for both requests and responses and it is formed by four sections: `question`, `answer`, `authority` and `additional`.

There are two types of DNS queries: *recursive* and *non-recursive*. The first kind of query returns a record without contacting other DNS servers, while the second forwards the request to other servers to obtain a portion of the answer.

A particular query is the `ANY` request that asks for all known informations about a DNS zone in a single request. Usually the response to this query is a packet with a total size of about 4 MByte.

Open Resolver DNS, which are recursive DNS resolvers that accept requests from anyone, are frequently used as amplifiers to perform DDoS amplification attacks. These servers allow recursive ANY queries that make it possible to generate a large response by sending a single packet of 64 bytes. The BAF for this amplification attack is 40x.

A DNS amplification attack relies on the use of publically accessible open DNS servers to flood a victim system with DNS excessive response traffic, this attack is a common form of DDoS. Figure 4.2 shows an answer that is bigger than the original request. An alert [12] to this kind of attack was released in 2013 by US-CERT. To date, no effective solution to defeat this attack has been developed and only mitigation approaches have been devised.

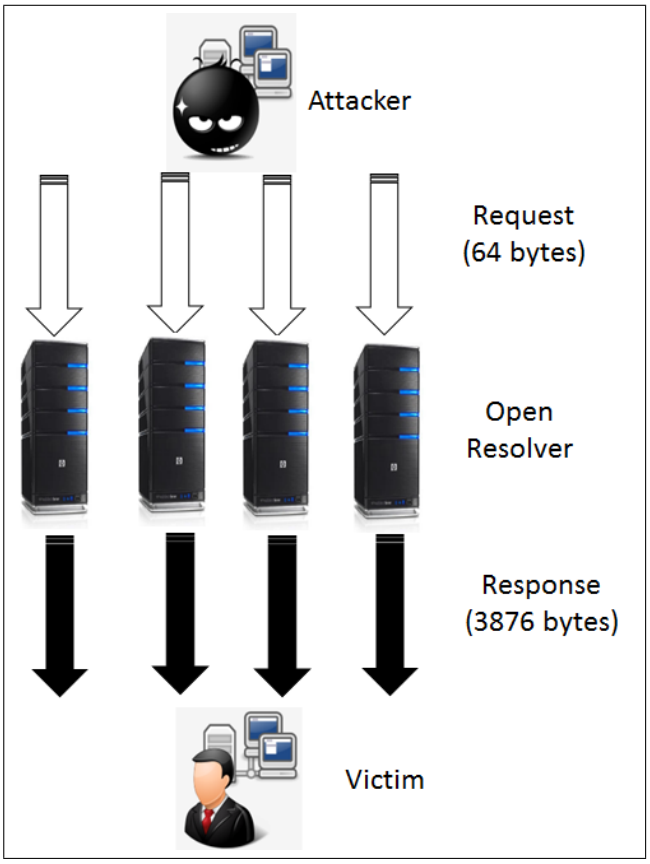


Fig. 4.2. DNS Amplification Attack

4.2.2 NTP Amplification Attack

Network Time Protocol (NTP), is an Internet protocol that allows the clock synchronization of computer systems. It is able to increase the local time precision over packet routing networks with variable latency. NTP was defined in RFC 958 [13] and uses UDP as its transport layer over the default port 123.

This protocol has been designed to distribute time information in a large, diverse system. NTP uses a symmetric architecture in which a distributed subnet of time servers operating in a self-organizing, hierarchical configuration synchronizes local clocks within the subnet and to national time standards via wire, radio, or calibrated atomic clock [14]. Usually the hierarchical structure is consists of four levels.

The first level is directly connected to and synchronized with a time source, such as an atomic clock. Each server belonging to a layer synchronizes its clock

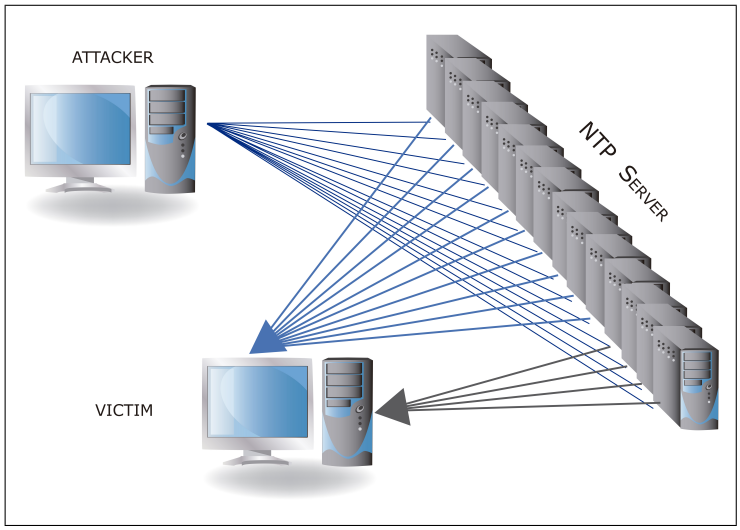


Fig. 4.3. NTP reflection DDoS Attack

with the others server of the same layer and with those of the upper layer. NTP clients belong to the fourth level.

NTP servers allow to query the number of client connected with them. This function can be performed with the command `monlist`. An NTP server, which allows this functionality, replies to a `monlist` request by sending back to the source the list of the last 600 clients IP addresses connected with it. The response is made by a series of packets, each containing, in addition to the clients IP addresses, other information. Each response packet has a payload size of 440 bytes containing a total of 6 IP addresses. The server could generate a response of 100 packets, with a total size of 43 KB. As a DNS server, also an NTP server could be exploited to perform a DDoS using it as amplifier. The attacker sends a `monlist` request to the server with the IP source address spoofed to the victim address. The server answers to the spoofed address with a much larger response. Using the formula of Bandwidth Amplification Factor, we estimate a factor of 205x. A DNS server has a BAF of 40x, i.e. 500 percent less than that of an NTP server. In addition, an NTP server sends more packets and thus facilitates bandwidth exhaustion. During 2014 the US-CERT published an alert to notify this attack [15, 16]. The vulnerability is described in cve-2013-5211 and a solution to the problem is specified, involving patching of the server or disabling of the `monlist` functionality. Fig. 4.3 shows abuse of an NTP to perform a massive reflection DDoS attack.

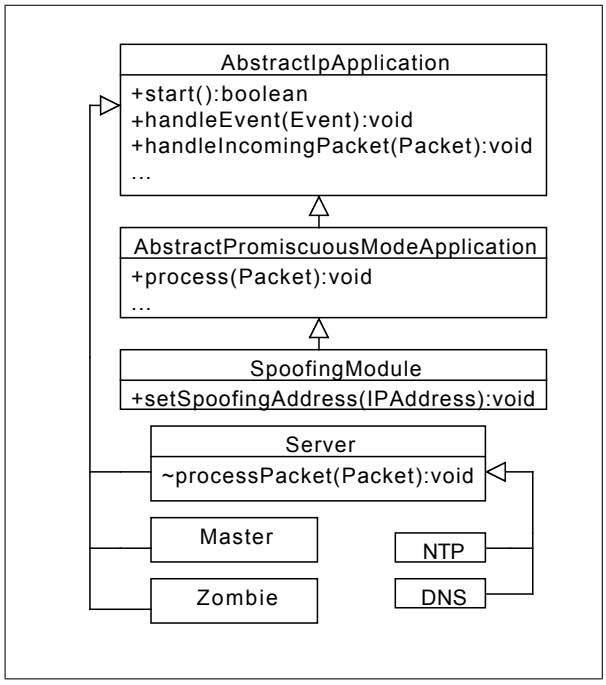


Fig. 4.4. Class diagram

4.3 Simulation Model

To model the attack scenario we introduced three new entities (master, zombie, server) and a custom NeSSi^2 application was implemented for each of them (see class diagram of Fig. 4.4).

The abstract class `AbstractIPerApplication` is the base class that should be extended to introduce a new NeSSi^2 application. It has three abstract methods: `start`, which allows setting up of the application parameters (e.g. registering the UDP port which has to be used); `handleEvent` which is the point where discrete events are handled and new ones are generated and scheduled; `handleIncomingPacket`, which defines how incoming packets are dealt with.

In addition to the above mentioned applications, another support module was developed to enable source address spoofing. To achieve this, the class `AbstractPromicuousModeApplication`, that allows the pre-processing of all incoming and outgoing packets of a given host, was properly extended. In this case, only the method `process` has been redefined for overriding the source address field with the address specified by the newly introduced `setSpoofedAddress` method.

The `Master` class models the behavior of the attacker which controls a botnet in order to flood a host without exposing himself. At the beginning of the simulation, the master agent searches automatically all zombie agents

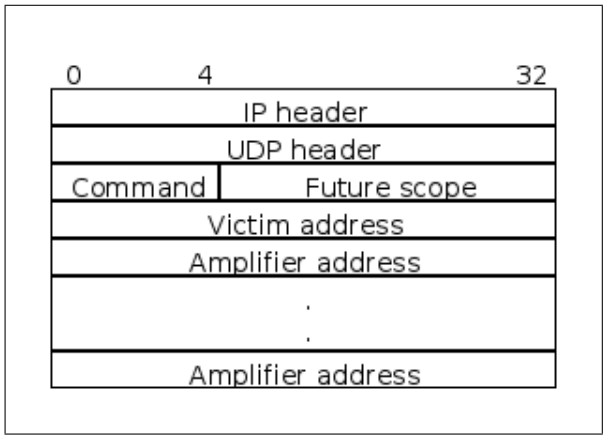


Fig. 4.5. Master packet format

and, after a certain amount of time, it sends each of them a UDP packet with a specific command. The packet source address may be the real one or it can be changed by exploiting the spoofing module. The packet payload (see Fig. 4.5) contains the information necessary to perform the DDoS attack: the type of amplification attack (i.e. DNS or NTP), the victim address and the list of the server addresses to be used as amplifiers.

The `Zombie` class models the behavior of the malicious code which infects a zombie machine. When this application is installed on a given host, it starts listening on the port specified as parameter and waits for a command from the master. The zombie application can be customized by the following parameters:

- `interval`: the amount of time that elapses between two server requests;
- `backdoor`: the port over which master commands are received;
- `snumber`: the number of servers contacted per each issued request.

When the zombie receives a packet from the master, it extracts the contained information to configure the attack. If the received list of amplifiers contains more than `snumber` addresses, a suitable set of them is randomly chosen on each request.

Currently, only DNS and NTP amplification attacks are supported. In particular, in the first case the zombie sends `ANY` requests to DNS servers while in the other case it sends `monlist` requests to NTP servers. More protocols can be easily added.

The `Server` base class implements the general behavior of a server which is specified by the finite state automaton shown in Fig. 3.13 (see Section 3.5.4) [17]. A server is able to process up to n requests in parallel. In particular, if it found itself in the `Available` state, it handles incoming requests as soon as they arrive; on the contrary, when there are no more available processing

resources, it switches to the `Busy` state and stores the incoming requests into a limited buffer. If the buffer gets filled, incoming requests are dropped. Service time is assumed to follow an exponential distribution with the mean value specified as parameter.

The server application can be configured by the following parameters

- `RN`: the number of server resources (zero is interpreted as infinite);
- `port`: the port over which requests are received;
- `mean`: the expected value of the negative exponential distribution.

The `Server` class provides an abstract method, `processPacket`, which is the callback that must be overridden to define how requests are processed according to the specific service provided.

The `DNSServer` and `NTPServer` classes extend the `Server` base class and suitably implement the `processPacket` method. The `Victim` class models a a simple UDP client.

In addition to the above NeSSi^2 applications, it is also possible to install a packet sniffer (`PacketSnifferApplication`) which is able to collect traffic data and to store them in a PCAP file for statistical analysis of these applications

The network topology which we considered is shown in Fig. 5.7. There are six subnets. The first subnet contains only one host running the `Master` application. The second, third and fourth subnets are composed of fifty hosts each, which are used by the master as zombies. The fifth subnet contains three servers used as amplifiers. The last subnet has only one host having the victim profile. Link bandwidths are summarized in Table 4.1 and Table 4.2.

Simulation parameters are the same in both simulation scenarios. In the next section, results of the simulation will be shown.

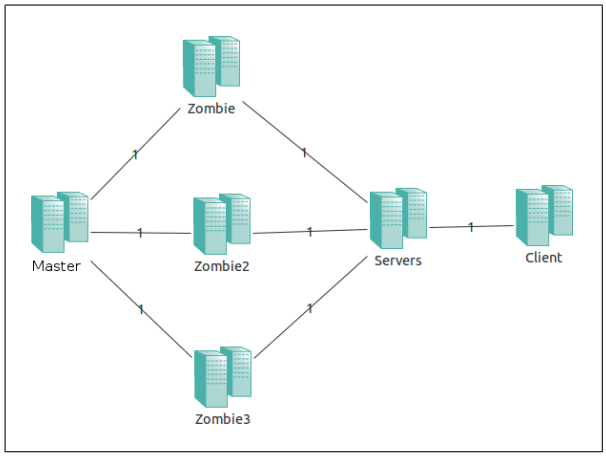


Fig. 4.6. Topology

Table 4.1. Bandwidth of links between subnets

	Master	Zombie	Server	Victim
Master	-	1Mbit	-	-
Zombie	1Mbit	-	10Mbit	-
Server		10Mbit	-	1Gbit
Victim	-	-	1Gbit	-

Table 4.2. Subnet internal link bandwidths

	Master	Zombie	Server	Victim
Link size	1Mbit	10Mbit	200Mbit	1Gbit

4.4 Simulation Results

The developed simulation model has been used to evaluate and compare the impacts of both DNS and NTP amplification attacks. The considered scenarios use a botnet of 150 zombies that exploits 3 vulnerable servers with an available bandwidth of 200 Mbit/s. The effect of the attacks were monitored by measuring the bandwidth utilization on three points:

- on a link inside a zombie subnet to evaluate the bandwidth needed to a zombie which cooperates in the DDoS attack;
- on a link belonging to the server subnet to evaluate the traffic generated by each single server;
- on link between servers and the victim subnet to observe the power of the attack.

In the first considered scenario we installed on zombies and servers the suitable applications to reproduce an NTP amplification attack, instead, in the second scenario we installed the profiles for the DNS amplification attack.

Fig. 4.7 shows the evolution of the bandwidth occupation on the link between the server and victim subnet during 20 seconds of attack. The graph displays two lines, the continuous line represents the traffic generated by the NTP servers and the dashed line the traffic of the DNS servers. The bandwidth used by the NTP attack is the 300 percent much bigger than that of the DNS attack. Both attacks have the same growth speed, but the DNS has a more stable behavior compared to NTP that oscillates between 400 Mbit/s and 500 Mbit/s. The reason is that NTP generates more packets than DNS as shown in Table 4.3 with a ratio of 23:1. This peculiarity of the NTP protocol makes the attack much more effective.

Fig. 4.8 and Fig. 4.9 show the comparison between incoming and outgoing packets from the servers under the two scenarios. The amplification effects are clearly observable in both graphics where the continuous line indicates the responses and the dashed line the malicious queries. The huge difference

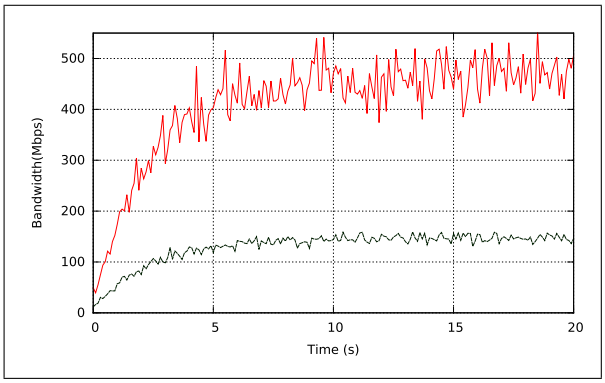


Fig. 4.7. Comparison between DNS and NTP impact on victim link

Table 4.3. Statistics for the victim's link

Statistics		
Protocol	NTP	DNS
Packets	1863889	79840
Interval (s)	20,00	20,00
Avg. packets/sec	105304	4090
Avg. Mbit/sec	406	130
Max. peak Bandwidth (Mbit)	500	200

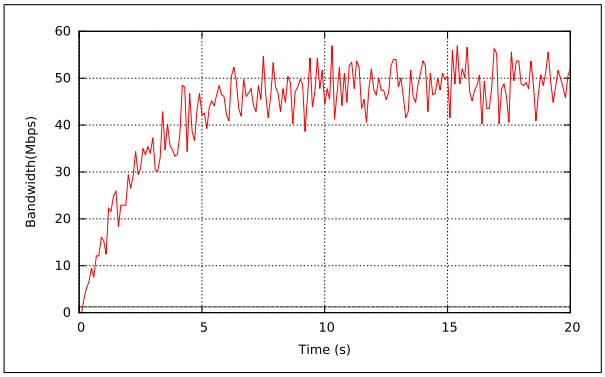


Fig. 4.8. DNS server in-out

between the two directions makes it difficult understand the bandwidth generated from the botnet. Table 4.4 reports the average value of bandwidth on the server link. It is possible to see that the two attacks produce a very small and similar traffic flow toward the servers.

Regarding the zombies, a single host is able to join with the attack using 25 Kbit/sec for the DNS attack and 67 Kbit/sec for the NTP attack.

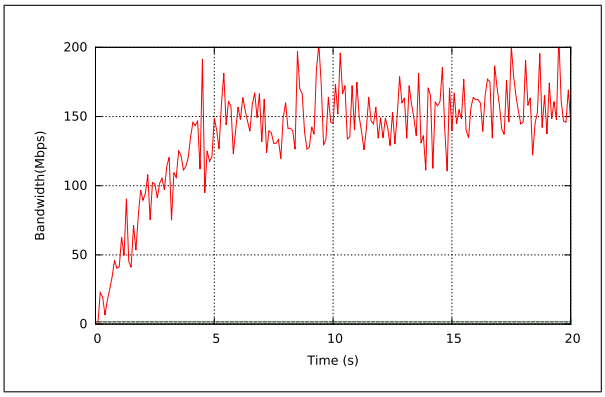


Fig. 4.9. NTP server in-out

Table 4.4. Incoming and outgoing packets for NTP and DNS server

	NTP	DNS
Ingoing AVG. Mbit/sec	1,6	1,2
Outoing AVG. Mbit/sec	142,7	45,6

4.5 Conclusions

This work proposes a simulation model suitable for the evaluation of DDoS amplification attacks. The obtained simulation results demonstrate the effective dangerousness of the amplification attacks that permit a single host to generate a huge volume of traffic by exploiting infected hosts (botnet) and vulnerable servers. The developed model constitutes a valuable tool to easy the study of the effects of such types of attacks, as discussed in this chapter, or to develop and test appropriate defense techniques/mechanisms which will be discussed in the next chapter.

References

1. F. Lau, S. Rubin, M. Smith, and L. Trajkovic, "Distributed denial of service attacks," in *Proc. of IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, 2000, pp. 2275–2280.
2. C. Patrikakis, M. Masikos, and O. Zouraraki, "Distributed denial of service attacks," *The Internet Protocol Journal*, vol. 7, no. 4, pp. 13–35, 2004.
3. C. Douligeris and A. Mitrokotsa, "{DDoS} attacks and defense mechanisms: classification and state-of-the-art," *Computer Networks*, vol. 44, no. 5, pp. 643 – 666, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128603004250
4. V. Paxson, "An analysis of using reflectors for distributed denial-of-service attacks," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 3, pp. 38–47, Jul. 2001.

5. S. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 4, pp. 2046–2069, 2013.
6. S. Kumar, "Smurf-based distributed denial of service (DDoS) attack amplification in Internet," in *Internet Monitoring and Protection, 2007. ICIMP 2007. Second International Conference on*, July 2007, pp. 25–25.
7. C. Rossow, "Amplification hell: Revisiting network protocols for DDoS abuse," in *Proc. of Symposium on Network and Distributed System Security (NDSS)*. San Diego, CA, USA: Internet Society, 23-26 February 2014.
8. NeSSi^2, "Network security simulator," 2015. [Online]. Available: http://www.nessi2.de/
9. S. Yu, "An overview of DDoS attacks," in *Distributed Denial of Service Attack and Defense*. Springer, 2014, pp. 1–14.
10. P. Mockapetris, "Domain Names - concept and facilities," Network Working Group, 1987. [Online]. Available: https://tools.ietf.org/html/rfc1034
11. ——, "Domain Names - implementation and specification," Network Working Group, 1987. [Online]. Available: https://tools.ietf.org/html/rfc1035
12. US-CERT, "NTP amplification attacks using CVE-2013-5211," 2014. [Online]. Available: https://www.us-cert.gov/ncas/alerts/TA14-013A
13. D. L. Mills, "Network Time Protocol (NTP)," Network Working Group, 1985. [Online]. Available: https://tools.ietf.org/html/rfc958
14. ——, "Internet time synchronization: the network time protocol," *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1482–1493, 1991.
15. US-CERT, "Dns amplification attacks," 2013. [Online]. Available: https://www.us-cert.gov/ncas/alerts/TA13-088A
16. M. Orlando, "NTP can be abused to amplify denial-of-service attack traffic," 2014. [Online]. Available: http://www.kb.cert.org/vuls/id/348126
17. A. Furfaro, P. Pace, A. Parise, and L. M. Valdiviezo, "Modelling and simulation of a defense strategy to face indirect DDoS flooding attacks," in *Internet and Distributed Computing Systems*. Springer, 2014, pp. 263–274.

5

A Defense Technique Against Indirect DDoS Flooding Attacks

Abstract

Computer simulations allow to model and reproduce DDoS attack scenarios and to gain more insights about how attacks may evolve and on their effects. This allows to develop detection, reaction and defense techniques whose effectiveness can be, in turn, evaluated into a simulated environment before deploying them in real settings. This chapter introduces an extension of the *StopIt* technique which makes it able to cooperate with capability-based mechanisms for defeating indirect attacks. The enhanced version of the protocol has been implemented into the ns-3 simulator, and its effectiveness has been evaluated under different scenarios.

5.1 Introduction

Currently, there is a wide variety of attacks, among all possible security attack types, one of the most interesting and sophisticated attack technique, that still is not easily contained, is represented by the DDoS which is very difficult to detect and to deal with because of its distributed nature. DDoS attacks are often launched by a network (*botnet*) of remotely controlled, well organized, and widely scattered computers (*zombies*) that are simultaneously and continuously sending a large amount of traffic and/or service requests to the target system. In particular, the attack can be launched both to internet/transport (*i.e.* by exhausting bandwidth, router processing capacity or network resources) or application level (*i.e.* by exhausting the server resources such as memory, CPU, sockets) thus causing a considerable slowdown of responses from the target system or even a complete crash.

Starting from this challenging and modern communication scenario, the chapter shows how to configure and extend the well known ns-3 network simulator [1, 2] for testing different network communication scenarios during DDoS attacks; moreover, a new strategy to improve the performance of a standard

hybrid defence mechanism is also proposed and validated. The main contributions can be summarized as follows:

1. we recall the operation of the standard *StopIt* mechanism to face DDoS security attacks and we show its performances and limitations by implementing a specific simulation model;
2. we propose an extension of that technique by taking advantage from the integration of the *DiffServ* architecture, typically used to guarantee specific quality of service levels within a communication network;
3. we have implemented our proposal and we have tested it throughout simulations in order to demonstrate the extreme reactivity and adaptability to different network traffic conditions.

The chapter is structured in three parts. The first one (Section 5.2) presents a novel solution, to overcome some of those limitations. The second part (Section 5.3) validates the goodness of the proposal by showing results obtained throughout several simulation campaigns in different network conditions. Third part presents (Section 5.4) concludes the chapter.

5.2 The Proposal

According to the studies conducted in ([3]b), *StopIt* (Section 2.3.3) outperforms filter-based designs such as AITF, and it is effective in providing non-interrupted communication under a wide range of DDoS attacks. However, *StopIt* does not always outperform capability-based mechanisms if the attack traffic does not directly reach a victim, but congests a link shared by the victim. In such cases, a capability-based mechanism (*e.g.* *TVA* [4], *TVA+*([3]b)) is more effective but it presents several disadvantages mainly due to *i)* the accuracy of the attack detection mechanism that each receiver can employ, *ii)* the significant amount of per flow state information to be maintained at each router turning into high processing and memory costs. In particular, in case of indirect attack, the behaviours of *TVA+* and *StopIt* are opposite: *StopIt* experiences an increase of the data transfer time with the growth of data traffic and *TVA+* does not complete the data transfer if the bandwidth is not high enough. The best solution, to get the advantages of both defense mechanisms, would consist into the merge of the two hybrid techniques so as to obtain the benefits of both. However, the merging of those two techniques is not trivial because it would consist into the creation, from scratch, of a new hybrid mechanism having an high complexity with consequent impacts on the actual feasibility of the system; thus we propose to make the *StopIt* system able to activate a priority-based strategy, already available within the AS routers, with the aim of offering privileges to traffic with destination towards the server that is experiencing a degradation of its performance.

Starting from this intuition, we propose an *"uncommon"* use of the *DiffServ* [5] model for IP traffic differentiation as a simple, but effective, technique to make the *StopIt* more robust against the indirect flooding attacks.

In this way, the *StopIt* mechanism will face an indirect DDoS attack by using a technology already supported by the access routes and designed to provide a specific QoS level within the network.

DiffServ has been already used against DDoS attacks. In particular, it was employed to attenuate the effect of DDoS flooding attacks [6] by adopting a bandwidth allocation policy which assigns to the normal traffic a priority higher than the one assigned to suspicious traffic. Incoming packets are analysed and classified through a suitable anomaly detection module. On the contrary, the solution proposed in this work does not extend the *DiffServ* model but simply exploits it as a standard tool to implement an effective capability-based mechanism.

In the next subsections first, we briefly recall the main features of the *DiffServ* model, then we explain the dynamic integration of that model with the *StopIt* mechanism.

5.2.1 DiffServ for traffic management

The *DiffServ* is a *coarse-grained*, *class-based* mechanism for traffic management and QoS differentiation. It was developed to differentiate IP traffic so that traffic relative priorities could be determined on a per-hop basis. By using *DiffServ*, traffic is classified based on priority; then the traffic is forwarded using one of the three IETF-defined per-hop behavior (PHB) mechanisms corresponding to different priority queues.

This approach allows traffic with similar service characteristics to be passed with similar traffic guarantees across multiple networks, even if the multiple networks do not provide the same service at the same way.

DiffServ replaces the first bits in the standard ToS (Type of Service) byte with a differentiated services code point (DSCP) that is then mapped to the PHB. This technique allows service providers to control how the DSCP code points are mapped to PHBs, and each time a packet enters a network domain it may be re-marked.

Most common networks use the following generally defined PHBs:

- *Assured Forwarding (AF)* that gives assurance of delivery under prescribed and stringent conditions (Premium Service)
- *Expedited Forwarding (EF)* dedicated to low-loss, low-latency traffic
- *Default Behaviour (BE)* typically used for best-effort traffic

StopIt-DiffServ cooperation

In case of an indirect DDoS attack where the flooding traffic is sent to a normal host H_u sharing a link with a DNS Server H_d, the dynamic activation of the *DiffServ* support by the server H_d, experiencing a performance degradation, is implemented according to the following assumptions within the communication network:

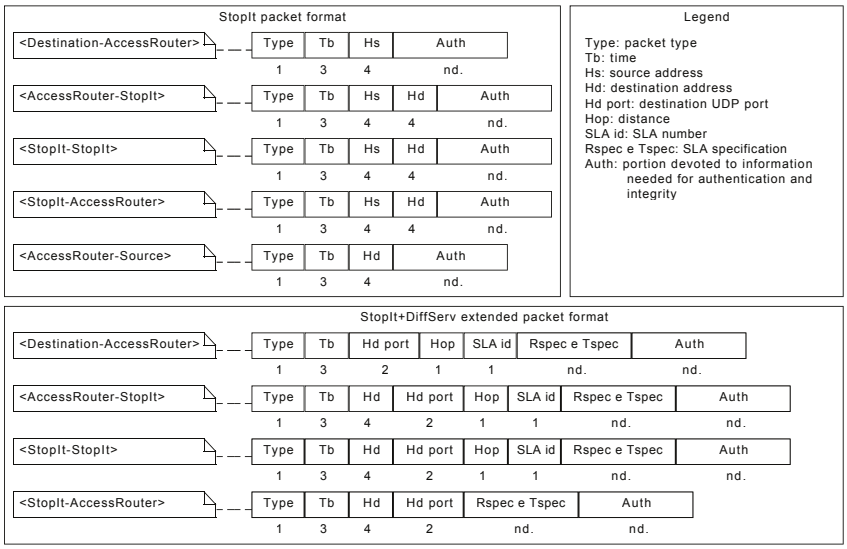


Fig. 5.1. *StopIt* packet format for *DiffServ* activation

- At least one *StopIt* server is present within each AS;
- Each AS corresponds to a *DiffServ* domain;
- In each *DiffServ* domain, the packets coming from the *StopIt* server are managed throughout the highest priority Assured Forwarding (AF) queue;
- The *DiffServ* system is able to install new Service Level Agreements (SLAs) at run time;
- The server H_d experiencing a performance degradation is able to detect anomalous traffic conditions by using a specific detection algorithm that is out of the scope of this work.

To support the integration between *StopIt* and *DiffServ*, we extended the standard packet format exchanged among different network entities by adding specific fields such as the Hop counter, the SLA identifier, the Rspec and Tspec representing traffic and request specifications respectively. Figure 5.1 shows the comparison between the standard *StopIt* packet format and the extended version.

Once the server H_d detects a decrease in its performance, mostly due to traffic anomalies, it starts the activation of the jointly *StopIt-DiffServ* defence mechanism by executing the following steps:

1. H_d sends a temporary *DiffServ* activation request toward the access router R_d within its AS by respecting the packet format defined in Figure 5.1;
2. R_d forwards the request to the *StopIt* server after filling the packet with the information about all the interfaces connected to the AS;

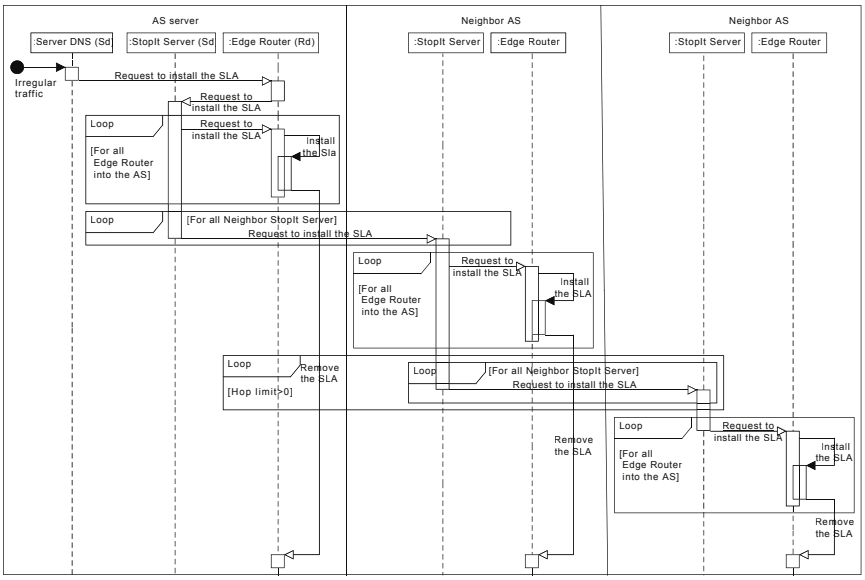


Fig. 5.2. *StopIt* and *DiffServ* routers interactions

3. The *StopIt* server installs the specific SLA for a certain time T_b, then it decreases by one the hop limit field and forwards the request to all the neighbour ASs
4. The other *StopIt* servers, once received the request packet, repeat the actions from point 2 until the hop limit field reaches zero.

At the end of this procedure, the ASs in which the SLA has been installed, will give priority to traffic destined to H_d, so that it is not affected by the attack any more. It is worth to note that, the proposed technique for packet diffusion is similar to the selective flooding strategy used by the OSPF [7] routing protocol; thus it can be easily implemented as well.

Figure 5.2 shows the interactions between the routers involved into the activation of the *StopIt-DiffServ* cooperation.

5.3 Simulation Analysis and Results

In this section, the modelling and simulation of the *StopIt* mechanism is described, together with the proposed enhancements obtained by using the ns-3 discrete-event network simulator specifically designed to test and validate the performances of wireless and wired IP network systems. In particular, ns-3 is one of the fastest and efficient network simulators freely available on the

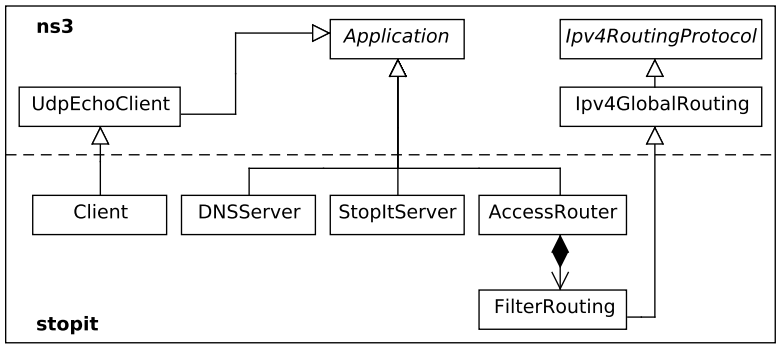


Fig. 5.3. *StopIt* class hierarchy

web in which the simulation time discretely moves from one event to another. Events are scheduled at a specific simulation time and they will wait until that time to be executed. To perform a simulation, the network topology is first setup and configured with the desired attributes such as link bandwidths, propagation delays and traffic sources. For the present study, the proposed strategy has been developed by integrating a freely downloadable *DiffServ* model for ns-3 simulator [8] with the standard operation offered by the *StopIt* mechanism in order to generate a unified simulation framework. In particular the *StopIt* implementation has been made under the following assumptions:

- IP addresses cannot be spoofed because *StopIt* deployment in a real network subsumes the use of Passport;
- The only network elements corrupted are the hosts belonging to the botnet;
- Strategic attacks directed to filter exhaustion are not taken into account;
- Only IPv4 networks are considered;
- *StopIt* servers are already aware of their peers at startup;
- Access routers play also the role of edge routers and are able to install/remove *DiffServ* SLAs at runtime.

5.3.1 Modelling with ns-3

In order to simulate in ns-3 a DDoS attack scenario in presence of the *StopIt* defense mechanism, the needed components have been implemented by introducing suitable classes,that inherit from the ns-3 `Application` base class, which respectively reproduce the behavior of a DNS server (the victim), *StopIt* servers, routers supporting packet filtering and DNS clients (see Fig. 5.3). The DNS server, whose behavior is specified by the finite state automaton shown in Fig. 3.13 (see subsection 3.5.4), is modelled by the `DNSServer` class as a multithreaded application which is able to process up to n requests in

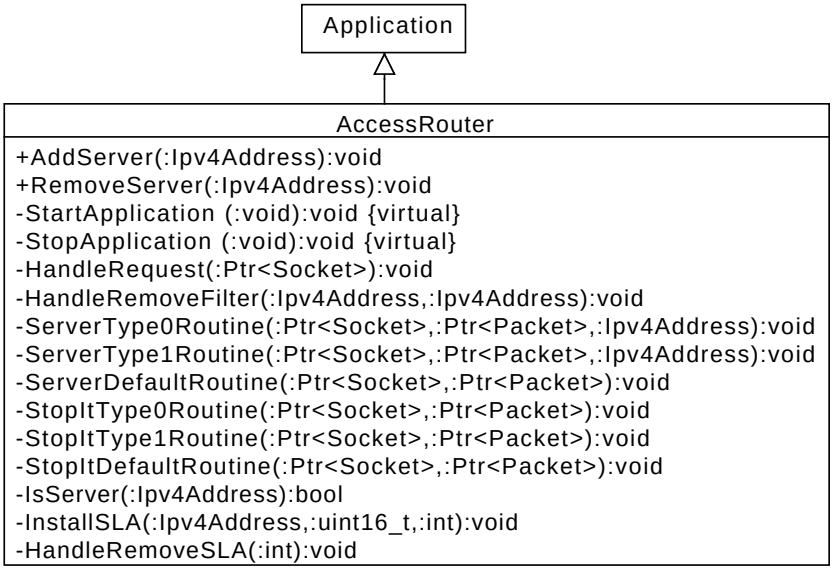


Fig. 5.4. AccessRouter class

parallel. Service time is assumed to follow an exponential distribution with mean 5 ms.

As stated before, *StopIt* needs the support of an attack detection algorithm. For simulation purposes, the presence of such an algorithm has been modelled by using a detection function which is parametrized with respect to the probability of detecting a malicious address and the time employed to recognize an attack since the arrival of the first malicious packet. A *StopIt* server is modelled by the `StopItServer` class which reproduces the behavior described in Sections 2.3.3 and 5.2. The `AccessRouter` class implements the router application which is in charge of packet filtering, dispatching of *StopIt* requests and *DiffServ* policy enforcement. Figure 5.4 depicts the `AccessRouter` class diagram.

Figure 5.5 shows the `StopItServer` class diagram. The C++ code which defines the `StopItServer` interface is shown in Fig. 5.6.

5.3.2 Simulation Scenario

In order to evaluate the effectiveness of the proposed mechanism we simulated both direct and indirect DDoS attacks against a DNS server. The network topology that has been considerd is represented in Figure 5.7. The network is split up into three zones as in [9]. The first zone contains ten ASs, each made up of 50 hosts, where the traffic sources are located in. The 50% of such ASs is

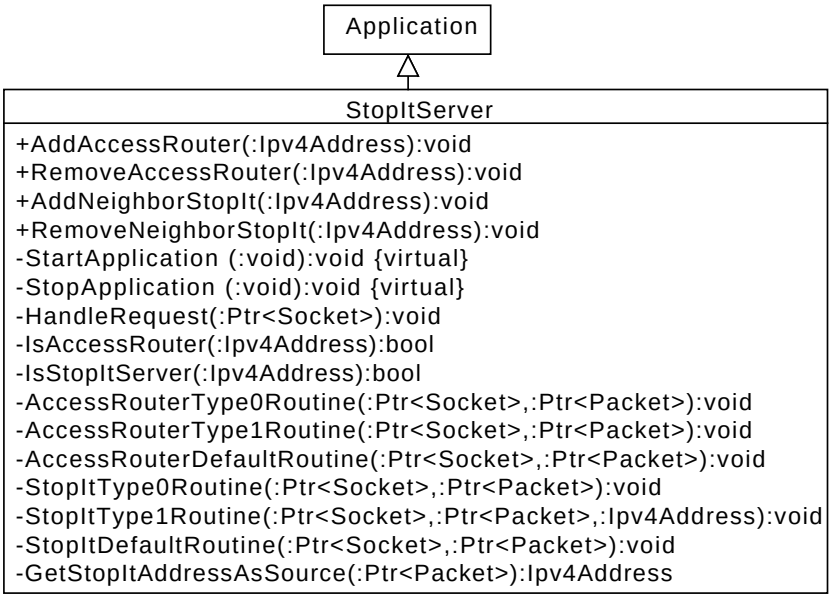


Fig. 5.5. StopItServer Class

corrupted and belongs to the botnet. The second zone represents the intermediate network and the third zone contains only the victim's AS. To evaluate the behaviors of the defense mechanisms in a realistic scenario we introduced different types of traffic sources have been introduced. We included 24 VoIP sources (using the ilbc_mode_30 codec at 13.33kbps), 230 HTTP sources and 230 DNS sources. During normal network operation, *DiffServ* handles both DNS and HTTP sources as Best Effort traffic and VoIP sources as Assured Forwarding. To face an indirect DDoS attack, DNS traffic is handled as coming from high priority sources. Since the simulated attacks aimed at exhausting the available capacity of the link L_d, we measured its bandwith occupation.

Links		**DNS Service**		**Legal DNS traffic**		**Malicious traffic**	
Bandwidth	10 Mbps	Resources	8	Packet size	26 bytes	Packet size	78 bytes
Delay	1 ms	Buffer size	200	Packet rate	1 pkt/s	Packet rate	100 pkt/s
		Mean service time	5 ms				

Table 5.1. Simulation parameters

```
#ifndef STOPIT_SERVER_H
#define STOPIT_SERVER_H

#include <vector>
#include "ns3/application.h"
#include "ns3/event-id.h"
#include "ns3/ptr.h"
#include "ns3/address.h"
#include "ns3/ipv4-address.h"
#include "ns3/diff-serv-sla.h"

namespace ns3 {

class StopItServer: public Application
{
public:
        static TypeId GetTypeId (void);
        StopItServer ();
        virtual ~StopItServer ();
protected:
        virtual void DoDispose (void);
private:
        virtual void StartApplication (void);
        virtual void StopApplication (void);

void HandleRequest (Ptr<Socket> socket);
bool IsStopItServer(Ipv4Address source);
void AccessRouterType1Routine(Ptr<Socket> socket,
        Ptr<Packet> packet);
void StopItType0Routine(Ptr<Socket> socket,
        Ptr<Packet> packet);
void StopItType1Routine(Ptr<Socket> socket,
        Ptr<Packet> packet);
uint16_t m_port;
Ptr<Socket> m_socket;
Address m_local;
Ipv4Address m_router;
vector< Ipv4Address > m_stopit;
int const m_initialCodePoint = 0;
int const m_nonConformantActionI=0;
int const m_nonConformantActionII=0;

};
} // namespace ns3
#endif /* STOPIT_SERVER_H */
```

Fig. 5.6. `StopItServer` interface definition

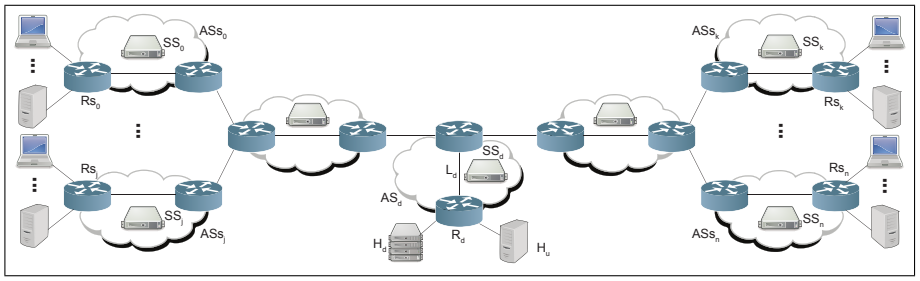


Fig. 5.7. Network topology

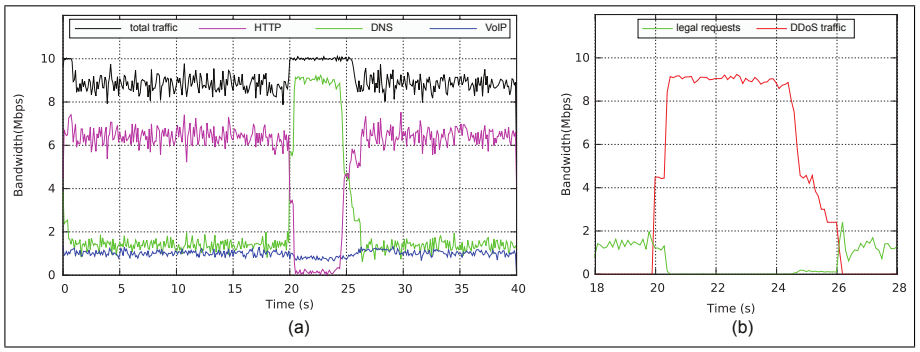


Fig. 5.8. (a) Direct DNS DDoS attack (b) Detail of legal and malicious DNS traffic

5.3.3 Direct Flooding Attack

Figure 5.8(a) shows the effectiveness of *StopIt* in the case of direct attack. The black curve is the total used bandwidth, the purple line represents the HTTP traffic, the blue line is the VoIP traffic and the green one the DNS traffic (both legal and illegal). In the simulated scenario, the attack begins at $t = 20$ s and it is detected after 3 s. During the time interval between the begin of the attack and the *StopIt* response, the DNS traffic increases and saturates the available bandwidth at expense of the HTTP packets. As can be seen in the details reported in Figure 5.8(b), during this period the DNS traffic is almost totally made of malicious packets. After the filters are successfully installed, the botnet traffic is blocked (100% of malicious sources identified) and the normal situation is restored. It is worth noting that the VoIP traffic remains almost unaffected due to the priority ensured by the *DiffServ* policy.

5.3.4 Shared Link Flooding Attack

Figure 5.9 depicts a scenario where the attack is achieved indirectly by flooding a host that belongs to the same AS of the victim with the aim of exhaust the available bandwidth of a shared link. In this case, the victim does not

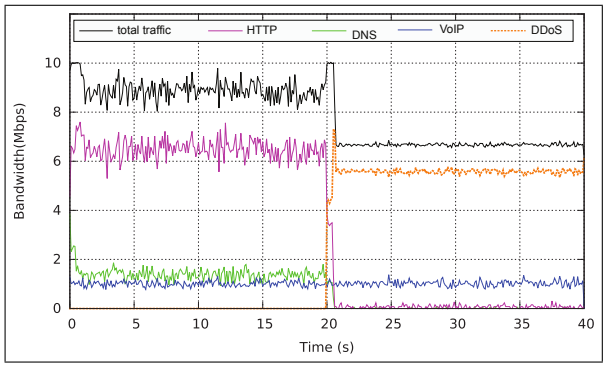


Fig. 5.9. Indirect DDoS attack (StopIt)

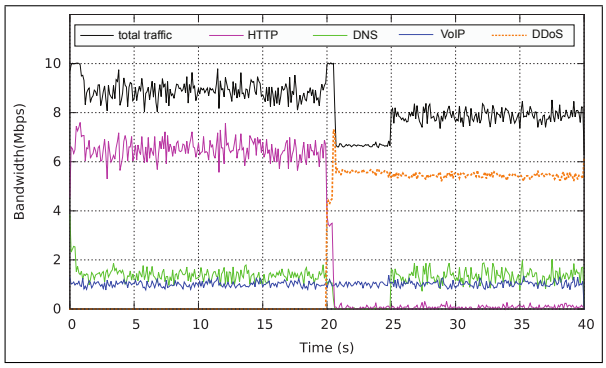


Fig. 5.10. Indirect DDoS attack (*StopIt* + *DiffServ*)

known the IP addresses of the zombies, so it can only become aware of the attack because its incoming traffic falls under the expected value for a long time. It can be seen that *StopIt* is not able to face the attack while the VoIP traffic remains unaffected as in the previous scenario. Figure 5.10 shows how the proposed defence strategy, by exploiting the cooperation of *StopIt* and *DiffServ*, re-insures the necessary bandwidth to the DNS server after the time needed to detect the anomalous behaviour and to dispatch the *StopIt* requests. Of course, the data traffic generated by the DDoS attack (orange curve) is still present in the network causing a decrease in terms of performances to the HTTP connections that experience a very low throughput. This issue is out of the scope of this work and it will investigated in a future research.

5.4 Conclusion

This chapter proposed and validated the integration of the *DiffServ* model within the *StopIt* mechanism to overcome the main limitation of this stan-

dard filter-based technique with the aim of facing both direct and indirect DDoS attacks. The cooperation between *DiffServ* and *StopIt* has the great advantage to be easily implemented in common routers since it is based on widely available technologies. The results, obtained throughout a self developed extension of the ns-3 simulator, have confirmed this initial intuition. However, the proposed solution cannot be considered as final because, even if the main services are guaranteed, the illegal sources still continue to overload the network.

References

1. nsnam, "Ns-3 documentation," http://www.nsnam.org/ns-3-19/documentation/.
2. G. Carneiro, H. Fontes, and M. Ricardo, "Fast prototyping of network protocols through ns-3 simulation model reuse," *Simulation Modelling Practice and Theory*, vol. 19, no. 9, pp. 2063 – 2075, 2011.
3. X. Liu, X. Yang, and Y. Lu, "To filter or to authorize: Network-layer DoS defense against multimillion-node botnets," in *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*, ser. SIGCOMM '08. New York, NY, USA: ACM, 2008, pp. 195–206.
4. X. Yang, D. Wetherall, and T. Anderson, "A dos-limiting network architecture," vol. 35, no. 4. New York, NY, USA: ACM, Aug. 2005, pp. 241–252. [Online]. Available: http://doi.acm.org/10.1145/1090191.1080120
5. IETF, "Rfc 2475: An architecture for differentiated services," http://www.ietf.org/rfc/rfc2475.txt, Tech. Rep., 1998.
6. W.-S. Lai, C.-H. Lin, J.-C. Liu, H.-C. Huang, and T.-C. Yang, "Using adaptive bandwidth allocation approach to defend ddos attacks," *International Journal of Software Engineering and Its Applications*, vol. 2, no. 4, pp. 61–72, 2008.
7. IETF, "Rfc 2328: Ospf - open shortest path first," https://www.ietf.org/rfc/rfc2328.txt, Tech. Rep., 1998.
8. S. Ramroop, "A diffserv model for the ns-3 simulator," http://www.eng.uwi.tt/depts/elec/staff/rvadams/sramroop/index.htm, 2011.
9. P. Kumar and S. Selvakumar, "Distributed denial-of-service (DDoS) threat in collaborative environment - A survey on DDoS attack tools and traceback mechanisms," in *Advance Computing Conference, 2009. IACC 2009. IEEE International*, March 2009, pp. 1275–1280.

6

Conclusions and Outlook

Cyber Security is a critical area of innovation for the development of current and future information systems. Nowadays, Distributed Denial of Service (DDoS) is one of the most sophisticated and effective attack techniques. In view of this, it is fundamental that DDoS prevention/defense algorithms be evaluated through suitable discrete-event simulation models in order to assess their effectiveness before being deployed in production systems.

The research activity described in this Ph.D. thesis was devoted to a study cyber attacks, specifically, Distributed Denial of service (DDoS) attacks. In particular the work mainly consisted in the design, modelling and implementation of an extension of the *StopIt* defense mechanism to face DDoS attacks, using NeSSi^2 and ns-3 as network simulator tools. This mechanism of defense against DDoS attacks has proven to be effective even in the case of indirect attacks, in which malicious traffic is not routed to the victim but other hosts on the same subnet in order to saturate the available bandwidth and hence to hinder the provision of the services offered by the victim.

As future work, an extension of the present research is planned: this will be developed by designing suitable detection algorithms that may directly run on edge network devices and exploit *StopIt* features to block illegal sources also in the case of indirect attacks.

References

1. A. Furfaro, P. Pace, A. Parise, and L. M. Valdiviezo, "Modelling and simulation of a defense strategy to face indirect DDoS flooding attacks," in *Internet and Distributed Computing Systems*. Springer, 2014, pp. 263–274.
2. Y. Xiang, K. Li, and W. Zhou, "Low-rate ddos attacks detection and traceback by using new information metrics," *Information Forensics and Security, IEEE Transactions on*, vol. 6, no. 2, pp. 426–437, 2011.
3. C. Rossow, "Amplification hell: Revisiting network protocols for DDoS abuse," in *Proc. of Symposium on Network and Distributed System Security (NDSS)*. San Diego, CA, USA: Internet Society, 23-26 February 2014.
4. M. Bailey, E. Cooke, F. Jahanian, Y. Xu, and M. Karir, "A survey of botnet technology and defenses," in *Conference For Homeland Security, 2009. CATCH'09. Cybersecurity Applications & Technology*. IEEE, 2009, pp. 299–304.
5. D. Radovanovic, T. Radojevic, D. Lucic, and M. Sarac, "It audit in accordance with cobit standard," in *MIPRO, 2010 Proceedings of the 33rd International Convention*, May 2010, pp. 1137–1141.
6. T. Mahmood and U. Afzal, "Security analytics: Big data analytics for cybersecurity: A review of trends, techniques and tools," in *Information Assurance (NCIA), 2013 2nd National Conference on*, Dec 2013, pp. 129–134.
7. PwC, "UK cyber security standards research," Department for Business, Innovation & Skills Cabinet Office, 2013.
8. M. Angelini, M. C. Arcuri, R. Baldoni, C. Ciccotelli, G. A. Di Luna, L. Montanari, I. C. Panetta, L. Querzoni, and N. V. Verde, "Italian cyber security report: Critical infrastructure and other sensitive sectors readiness," Department for Business, Innovation & Skills Cabinet Office, 2013.
9. D. Bisson, "The 5 most common attack patterns of 2014," 2015. [Online]. Available: http://www.tripwire.com/state-of-security/security-data-protection/cyber-security/the-5-most-common-attack-patterns-of-2014/
10. ISO, "Iso/iec 27001 - information security management." [Online]. Available: http://www.iso.org/iso/home/standards/management-standards/iso27001.htm
11. K. A. N.V., "The five most common cyber security mistakes," 2013. [Online]. Available: https://www.kpmg.com/Global/en/IssuesAndInsights/ArticlesPublications/Documents/five-most-common-cyber-security-mistakes.PDF

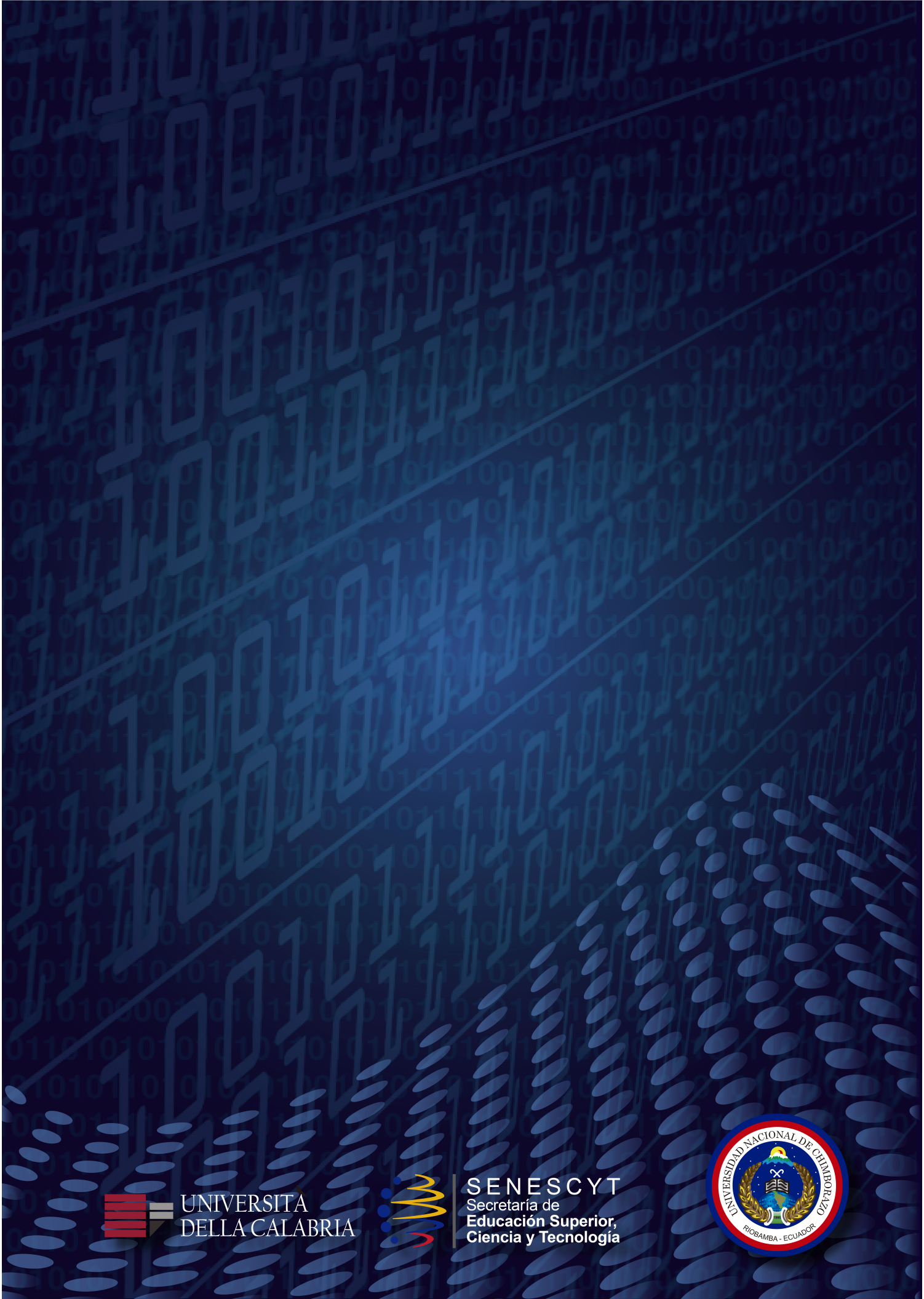
12. K. LLP, "Cyber security: its not just about technology," 2014. [Online]. Available: http://www.kpmg.com/Global/en/IssuesAndInsights/ArticlesPublications/Documents/cyber-security-not-just-technology.pdf
13. M. C. Waxman, "Cyber-attacks and the use of force: back to the future of article 2 (4)," *Yale Journal of International Law*, vol. 36, 2011.
14. ARN, "Top 10 most notorious cyber attacks in history." [Online]. Available: http://www.arnnet.com.au/slideshow/341113/top-10-most-notorious-cyber-attacks-history/
15. M. Sullivan, "8 types of cyber attacks your business needs to avoid," 2015. [Online]. Available: http://quickbooks.intuit.com/r/technology-and-security/8-types-of-cyber-attacks-your-business-needs-to-avoid
16. A. Ghorbani, W. Lu, and M. Tavallaee, "Network attacks," in *Network Intrusion Detection and Prevention*, ser. Advances in Information Security. Springer US, 2010, vol. 47, pp. 1–25. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-88771-5_1
17. P. Djalaliev, M. Jamshed, N. Farnan, and J. Brustoloni, "Sentinel: Hardware-accelerated mitigation of bot-based ddos attacks," in *Computer Communications and Networks, 2008. ICCCN '08. Proceedings of 17th International Conference on*, Aug 2008, pp. 1–8.
18. S. Stankovic and D. Simic, "Defense strategies against modern botnets," *arXiv preprint arXiv:0906.3768*, 2009.
19. C. Schiller and J. Binkley, *Botnets: The Killer Web Applications*. Elsevier Science, 2011. [Online]. Available: https://books.google.it/books?id=4MAuVjOx6kIC
20. C. Davis, "Iso 14001: What is it? why should i care?" 2015. [Online]. Available: http://blog.claytonhomes.com/blog/iso-14001-what-is-it-why-should-i-care
21. G. of the HKSAR, "An overview of information security standards," 2008. [Online]. Available: http://www.infosec.gov.hk/english/technical/files/overview.pdf
22. Y.-W. Shin, "It governance: Focusing on cobit model," 2014. [Online]. Available: http://www.lgcnsblog.com/inside-it/it-governance-focusing-on-cobit-model/
23. P. Charuenporn and S. Intakosum, "Qos-security metrics based on itil and cobit standard for measurement web services." *J. UCS*, vol. 18, no. 6, pp. 775–797, 2012.
24. R. Frost, "International organization for standardization (iso)," *The Quality Assurance Journal*, vol. 8, no. 3, pp. 198–206, 2004.
25. S. Donaldson, S. Siegel, C. Williams, and A. Aslam, "Cybersecurity frameworks," in *Enterprise Cybersecurity*. Apress, 2015, pp. 297–309. [Online]. Available: http://dx.doi.org/10.1007/978-1-4302-6083-7_17
26. ISO, "Iso/iec 27040:2015," 2015. [Online]. Available: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=44404
27. A. Sharifi, M. A. Beheshtizad, H. Sharifi, A. Nourollahi, J. Sharifi, and R. Asadi, "Designing an information technology model for audit in banks," *IJCER*, vol. 4, no. 1, pp. 17–22, 2015.
28. D. data, "What is a ddos attack?" 2013. [Online]. Available: http://www.digitalattackmap.com/understanding-ddos/
29. S. Yu, "An Overview of DDoS Attacks," in *Distributed Denial of Service Attack and Defense*. Springer, 2014, pp. 1–12.

30. T. Bienkowski, "5 misconceptions about the modern ddos attack," 2015. [Online]. Available: http://www.slideshare.net/Arbor_Networks/5-d-do-s-misconceptionswebinarslides
31. M.-S. Kim, H.-J. Kong, S.-C. Hong, S.-H. Chung, and J. Hong, "A flow-based method for abnormal network traffic detection," in *Network Operations and Management Symposium, 2004. NOMS 2004. IEEE/IFIP*, vol. 1, April 2004, pp. 599–612 Vol.1.
32. C. Weenk, "Ddos defense mechanisms: a state of the art research," 2007. [Online]. Available: http://referaat.cs.utwente.nl/conference/6/paper/6796/ddos-defense-mechanisms-a-state-of-the-art-research.pdf
33. J. L. Darte, *Identificación de ataques Inforáticos a través de Reyes Bayesianas*. Editorial Académica Española, 2012.
34. S. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 4, pp. 2046–2069, 2013.
35. S. Ranjan, R. Swaminathan, M. Uysal, and E. W. Knightly, "Ddos-resilient scheduling to counter application layer attacks under imperfect detection." in *INFOCOM*. Citeseer, 2006.
36. Neustar, "The danger deepens," 2014. [Online]. Available: https://www.neustar.biz/
37. C. Douligeris and A. Mitrokotsa, "{DDoS} attacks and defense mechanisms: classification and state-of-the-art," *Computer Networks*, vol. 44, no. 5, pp. 643 – 666, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128603004250
38. 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, 2004.
39. S. A. Sunitha M.S, "A low rate ddos: A security threat in mobile adhoc networks," *IJCSNS International Journal of Computer Science and Network Security*, vol. 12, no. 11, pp. 1–3, 2012. [Online]. Available: http://paper.ijcsns.org/07_book/201211/20121101.pdf
40. Y. Chen, K. Hwang, and Y.-K. Kwok, "Filtering of shrew ddos attacks in frequency domain," in *Local Computer Networks, 2005. 30th Anniversary. The IEEE Conference on*, Nov 2005, pp. 8 pp.–793.
41. S. Yu, "DDoS Attack Detection," in *Distributed Denial of Service Attack and Defense*. Springer, 2014, pp. 31–40.
42. Y. Chen and K. Hwang, "Collaborative detection and filtering of shrew ddos attacks using spectral analysis," *Journal of Parallel and Distributed Computing*, vol. 66, no. 9, pp. 1137–1151, 2006.
43. P. N. Jadhav and B. Patil, "Low-rate ddos attack detection using optimal objective entropy method," *International Journal of Computer Applications (0975 8887)*, vol. 78, no. 3, pp. 33–38, 2013. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.402.6134&rep=rep1&type=pdf
44. A. S. Tanenbaum and D. J. Wetherall, *Computer Networks: Pearson New International Edition: University of Hertfordshire*. Pearson Higher Ed, 2013.
45. S. Natarajan, X. Huang, and T. Wolf, "Efficient conflict detection in flow-based virtualized networks," in *Computing, Networking and Communications (ICNC), 2012 International Conference on*, Jan 2012, pp. 690–696.

46. X. Ma and Y. Chen, "Ddos detection method based on chaos analysis of network traffic entropy," *Communications Letters, IEEE*, vol. 18, no. 1, pp. 114–117, January 2014.
47. P. Kumar and S. Selvakumar, "Distributed denial-of-service (DDoS) threat in collaborative environment - A survey on DDoS attack tools and traceback mechanisms," in *Advance Computing Conference, 2009. IACC 2009. IEEE International*, March 2009, pp. 1275–1280.
48. T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," *ACM Comput. Surv.*, vol. 39, no. 1, Apr. 2007.
49. V. Thing, M. Sloman, and N. Dulay, "A survey of bots used for distributed denial of service attacks," in *New Approaches for Security, Privacy and Trust in Complex Environments*, ser. IFIP International Federation for Information Processing, H. Venter, M. Eloff, L. Labuschagne, J. Eloff, and R. Solms, Eds. Springer US, 2007, vol. 232, pp. 229–240.
50. R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker, "Controlling high bandwidth aggregates in the network," *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 3, pp. 62–73, Jul. 2002.
51. R. Chen, J.-M. Park, and R. Marchany, "Track: A novel approach for defending against distributed denial-of-service attacks," Technical Report TR-ECE-06-02, Dept. of Electrical and Computer Engineering, Virginia Tech, Tech. Rep., 2006.
52. T. Anderson, T. Roscoe, and D. Wetherall, "Preventing internet denial-of-service with capabilities," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 1, pp. 39–44, Jan. 2004.
53. K. Argyraki and D. R. Cheriton, "Scalable network-layer defense against internet bandwidth-flooding attacks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 4, pp. 1284–1297, Aug. 2009.
54. X. Liu, X. Yang, and Y. Lu, "To filter or to authorize: Network-layer DoS defense against multimillion-node botnets," in *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*, ser. SIGCOMM '08. New York, NY, USA: ACM, 2008, pp. 195–206.
55. X. Liu, A. Li, X. Yang, and D. Wetherall, "Passport: Secure and adoptable source authentication," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, ser. NSDI'08. Berkeley, CA, USA: USENIX Association, 2008, pp. 365–378.
56. S. Sendra, P. A. Fernández, M. A. Quilez, and J. Lloret, "Study and performance of interior gateway ip routing protocols," *Network Protocols and Algorithms*, vol. 2, no. 4, pp. 88–117, 2010.
57. D. Medhi and K. Ramasamy, *Network Routing: algorithms, protocols and architectures*, M. Kaufmann, Ed., 2007.
58. S. Specht and R. Lee, "Taxonomies of distributed denial of service networks, attacks, tools and countermeasures," Technical Report CE-L2003-03, Princeton University, http://www. princeton. edu/˜ rblee/ELE572_ F04Readings. html, Tech. Rep., 2003.
59. L. Qin, L. Hong, K. Songlin, and L. Chuchu, "Feature extraction and construction of application layer ddos attack based on user behavior," in *Control Conference (CCC), 2014 33rd Chinese*, July 2014, pp. 5492–5497.
60. J. Yu, Z. Li, H. Chen, and X. Chen, "A detection and offense mechanism to defend against application layer ddos attacks," in *Networking and Services, 2007. ICNS. Third International Conference on*, June 2007, pp. 54–54.

61. S. B. Ankali and D. Ashoka, "Detection architecture of application layer ddos attack for internet," *Int. J. Advanced Networking and Applications*, vol. 3, no. 01, pp. 984–990, 2011.
62. IMPERVA, "The top 10 ddos attack trends," 2015. [Online]. Available: https://www.imperva.com/docs/DS_Incapsula_The_Top_10_DDoS_Attack_Trends_ebook.pdf
63. N. Cybersecurity and C. I. Center, "Ddos quick guide," 2014. [Online]. Available: https://www.us-cert.gov/sites/default/files/publications/DDoS%20Quick%20Guide.pdf
64. Y. Xie and S. zheng Yu, "Monitoring the application-layer ddos attacks for popular websites," *Networking, IEEE/ACM Transactions on*, vol. 17, no. 1, pp. 15–25, Feb 2009.
65. Akamai, "Types of ddos attacks. an explanation of many types of ddos attacks." [Online]. Available: https://www.stateoftheinternet.com/types-of-ddos-attacks.html
66. M. Malekzadeh, M. Beiruti, and M. S. Abadi, "Assessment of high and low rate protocol-based attacks on ethernet networks," *Assessment*, vol. 6, no. 7, 2015.
67. Y. Dantas, V. Nigam, and I. Fonseca, "A selective defense for application layer ddos attacks," in *Intelligence and Security Informatics Conference (JISIC), 2014 IEEE Joint*, Sept 2014, pp. 75–82.
68. S. Khan and A. Saxena, "Detecting input validation attacks in web application," *International Journal of Computer Applications*, vol. 109, no. 6, pp. 1–4, 2015.
69. V. Durcekova, L. Schwartz, and N. Shahmehri, "Sophisticated denial of service attacks aimed at application layer," in *ELEKTRO, 2012*, May 2012, pp. 55–60.
70. M. Radwan and R. Heckel, "Detecting and refactoring operational smells within the domain name system," *arXiv preprint arXiv:1504.02615*, 2015.
71. S. Bao, J. Chen, W. Jiang, Z. Su, X. Yang, and J. Zhang, "Providing a virtual domain name system (dns) in a local area network (lan)," Mar. 24 2015, uS Patent 8,989,182. [Online]. Available: https://www.google.com/patents/US8989182
72. NeSSi^2, "Network security simulator," 2015. [Online]. Available: http://www.nessi2.de/
73. E. Lochin, T. Prennou, and L. Dairaine, "When should I use network emulation?" *annals of telecommunications - annales des télécommunications*, vol. 67, no. 5-6, pp. 247–255, 2012.
74. F. Lau, S. Rubin, M. Smith, and L. Trajkovic, "Distributed denial of service attacks," in *Proc. of IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, 2000, pp. 2275–2280.
75. C. Patrikakis, M. Masikos, and O. Zouraraki, "Distributed denial of service attacks," *The Internet Protocol Journal*, vol. 7, no. 4, pp. 13–35, 2004.
76. V. Paxson, "An analysis of using reflectors for distributed denial-of-service attacks," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 3, pp. 38–47, Jul. 2001.
77. S. Kumar, "Smurf-based distributed denial of service (DDoS) attack amplification in Internet," in *Internet Monitoring and Protection, 2007. ICIMP 2007. Second International Conference on*, July 2007, pp. 25–25.
78. S. Yu, "An overview of DDoS attacks," in *Distributed Denial of Service Attack and Defense*. Springer, 2014, pp. 1–14.
79. P. Mockapetris, "Domain Names - concept and facilities," Network Working Group, 1987. [Online]. Available: https://tools.ietf.org/html/rfc1034

80. ——, “Domain Names - implementation and specification,” Network Working Group, 1987. [Online]. Available: https://tools.ietf.org/html/rfc1035
81. US-CERT, “NTP amplification attacks using CVE-2013-5211,” 2014. [Online]. Available: https://www.us-cert.gov/ncas/alerts/TA14-013A
82. D. L. Mills, “Network Time Protocol (NTP),” Network Working Group, 1985. [Online]. Available: https://tools.ietf.org/html/rfc958
83. ——, “Internet time synchronization: the network time protocol,” *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1482–1493, 1991.
84. US-CERT, “Dns amplification attacks,” 2013. [Online]. Available: https://www.us-cert.gov/ncas/alerts/TA13-088A
85. M. Orlando, “NTP can be abused to amplify denial-of-service attack traffic,” 2014. [Online]. Available: http://www.kb.cert.org/vuls/id/348126
86. nsnam, “Ns-3 documentation,” http://www.nsnam.org/ns-3-19/documentation/.
87. G. Carneiro, H. Fontes, and M. Ricardo, “Fast prototyping of network protocols through ns-3 simulation model reuse,” *Simulation Modelling Practice and Theory*, vol. 19, no. 9, pp. 2063 – 2075, 2011.
88. X. Yang, D. Wetherall, and T. Anderson, “A dos-limiting network architecture,” vol. 35, no. 4. New York, NY, USA: ACM, Aug. 2005, pp. 241–252. [Online]. Available: http://doi.acm.org/10.1145/1090191.1080120
89. IETF, “Rfc 2475: An architecture for differentiated services,” http://www.ietf.org/rfc/rfc2475.txt, Tech. Rep., 1998.
90. W.-S. Lai, C.-H. Lin, J.-C. Liu, H.-C. Huang, and T.-C. Yang, “Using adaptive bandwidth allocation approach to defend ddos attacks,” *International Journal of Software Engineering and Its Applications*, vol. 2, no. 4, pp. 61–72, 2008.
91. IETF, “Rfc 2328: Ospf - open shortest path first,” https://www.ietf.org/rfc/rfc2328.txt, Tech. Rep., 1998.
92. S. Ramroop, “A diffserv model for the ns-3 simulator,” http://www.eng.uwi.tt/depts/elec/staff/rvadams/sramroop/index.htm, 2011.


UNIVERSITÀ
DELLA CALABRIA
SENESCYT
Secretaría de
Educación Superior,
Ciencia y Tecnología
UNIVERSIDAD NACIONAL DE CHIMBORAZO
RIOBAMBA - ECUADOR