VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

Simona RAMANAUSKAITĖ

MODELLING AND RESEARCH OF DISTRIBUTED DENIAL OF SERVICE ATTACKS

DOCTORAL DISSERTATION

TECHNOLOGICAL SCIENCES,
INFORMATICS ENGINEERING (07T)

LEIDYKLA
Vilnius TECHNIKA 2012

Doctoral dissertation was prepared at Vilnius Gediminas Technical University in 2008–2012.

Scientific Supervisor

Prof Dr Habil Antanas ČENYS (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering – 07T).

VGTU leidyklos TECHNIKA 1997-M mokslo literatūros knyga
<http://leidykla.vgtu.lt>

ISBN 978-609-457-141-1

© VGTU leidykla TECHNIKA, 2012
© Simona, Ramanauskaitė, 2012
simram@it.su.lt

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Simona Ramanauskaitė

SRAUTINIŲ ATAKŲ ĮTAKOS INTERNĖTINĖS PASLAUGOS SUTRIKDYMUI MODELIAVIMAS IR TYRIMAS

DAKTARO DISERTACIJA

TECHNOLOGIJOS MOKSLAI,
INFORMATIKOS INŽINERIJA (07T)

LEIDYKLA
Vilnius TECHNIKA 2012

Disertacija rengta 2008–2012 metais Vilniaus Gedimino technikos universitete.

Mokslinis vadovas

prof. habil. dr. Antanas ČENYS (Vilniaus Gedimino technikos universitetas,
technologijos mokslai, informatikos inžinerija – 07T).

Abstract

In the dissertation the Denial of Service (DoS) attacks and their models are investigated. DoS attack is a type of cyber attacks when an attacker tries to deny a service in the network machine. There are many types of DoS attacks, and therefore the main object of the dissertation is specified as distributed denial of service (DDoS) attacks. DDoS uses multiple agents at the same time to exhaust certain resources of network machine and make it unavailable.

The importance of DDoS attacks can be explained on the basis of the following facts: nowadays there are no countermeasures which can ensure full resistance to DDoS; DoS effect can be created even by legitimate users of the systems; internet services become more popular therefore the denial of such a service or diminishing of its quality can cause undesired impact on the other systems or their users.

The main objective of this dissertation is creation of model for the estimation of the composite DDoS attack success. This model would allow estimating of network machine resistance to different type and power DDoS attacks.

The dissertation consists of eight parts including Introduction, 5 chapters, Conclusions and References.

In the introduction, the investigated problem, importance of the thesis and the object of research are defined and the purpose and tasks of the thesis, scientific novelty are described together with the practical significance of results and defended statements. At the end of introduction, author's publications on the subject of the thesis are listed. They introduce and reflect the material of presentations at the conferences and justify the structure of the dissertation:

Chapter 1 – revision of DoS attacks, analysis of existing DoS attack and their countermeasure taxonomies. In this chapter, review of the specificities of the computer network modelling and analysis of the existing DoS and DDoS attack models is presented as well.

Chapter 2 – introduction and description of the new taxonomies for DoS attack and their countermeasure.

Chapter 3 – description of proposed models for different types of DDoS attacks as well as composite DDoS attack model.

Chapter 4 and 5 – investigation of the results of proposed models. They are compared with the results of the OPNET modelling tool, and different situations are analysed to represent characteristics of the DDoS attacks.

Conclusion is made that the proposed composite DDoS attack model is easily adaptable for different types of DDoS attacks and might be used to define probability of attack success.

Reziumė

Disertacijoje nagrinėjamos internetinės paslaugos sutrikdymo (*angl.* DoS – Denial of Service) atakos ir jų modeliavimo priemonės. Tai kibernetinių atakų tipas, kurių metu siekiama tam tikro tinkle veikiančio mazgo teikiamas paslaugas padaryti neprieinamas jų teisėtiems klientams. DoS atakos gali turėti daug skirtingų tipų, todėl šio darbo pagrindinis tyrimų objektas yra srautinė internetinės paslaugos sutrikdymo (*angl.* DDoS – Distributed Denial of Service) ataka, kuri paslaugos sutrikdymo siekia naudodama bent kelis atakuojančiuosius kompiuterius vienu metu ir kuri siekia išnaudoti visus pasirinkto tipo resursus tą paslaugą teikiančiame mazge. DDoS atakos aktualios dėl šių priežasčių: šiuo metu nėra apsaugos priemonių, leidžiančių patikimai ir užtikrintai apsisaugoti nuo jų keliamos grėsmės; jas gali sukelti net ir teisėti vartotojai, netinkamai elgdamiesi ar esant netinkamai paruoštai sistemai; internete teikiamos paslaugos vis dažniau naudojamos kasdieniniame gyvenime ir jų blokavimas ar kokybės suprastėjimas gali neigiamai paveikti kitų sistemų ar jų vartotojų darbą.

Pagrindinis šios disertacijos tikslas – sukurti jungtinį DDoS atakos sėkmės tikimybės vertinimo modelį. Šis modelis leistų įvertinti kompiuterinės technikos sugebėjimą atlaikyti skirtingo tipo ir galingumo DDoS atakas, todėl galėtų būti taikomas prevencijai bei paslaugų tiekėjų kokybės vertinimui.

Disertaciją sudaro įvadas, penki skyriai, rezultatų apibendrinimas, naudotos literatūros ir autoriaus publikacijų disertacijos tema sąrašai.

Įvadiniamе skyriuje aptariama tiriamoji problema, darbo aktualumas, aprašomas tyrimų objektas, formuluojamas darbo tikslas bei uždaviniai, aprašomas darbo mokslinis naujumas, darbo rezultatų praktinė reikšmė, ginamieji teiginiai. Įvado pabaigoje pristatomos disertacijos tema autoriaus paskelbtos publikacijos ir pranešimai konferencijose bei disertacijos struktūra.

Pirmajame skyriuje pateikiama DoS atakų specifika, analizuojamos egzistuojančios DoS atakų ir jų kontrapriemonių taksonomijos, šiame skyriuje taip pat apžvelgiama kompiuterių tinklų modeliavimo specifika ir analizuojami egzistuojantys DoS ar DDoS atakų modeliai.

Antrajame skyriuje aprašomos siūlomos DoS atakų ir jų kontrapriemonių taksonomijos, o trečiajame pateikiami siūlomi modeliai skirtingo tipo DDoS atakoms ir vienas jungtinis DDoS atakų modelis.

Ketvirtajame ir penktajame skyriuose pateikiami atliktų tyrimų rezultatai, kurie lyginami su OPNET modeliavimo įrankio rezultatais ir analizuojamos DDoS atakos savybės esant skirtingoms situacijoms.

Pasiūlytas jungtinis DDoS atakų modelis yra lengvai plečiamas ir gali būti taikomas skirtingo tipo DDoS atakų išankstiniame sėkmės vertinimui.

Notations

Symbols

c_a – average number of cycles for all attack bots sent packets processing;

c_b – average number of cycles for bot sent packets processing;

c_g – average cycle number for general job processing;

c_n – average number of cycles for legitimate user sent packets processing;

c_s – average cycle number needed for its processing;

f – CPU frequency;

K – number of active channels;

L – average time a job spends in the system;

M – size of additional buffer (bytes);

N – memory buffer size (number of queries in the buffer);

n – number of attack bots;

$n(t)$ – number of agents participating in DDoS attack at time moment t ;

$n_a(t)$ – number of agents, participating in DDoS attack at time moment t ;

n_{in} – initial number of agents in attack;

n_n – average number of legitimate users during the DDoS attack;

$n_r(t)$ – additional agent requirement function (describes how many agents are added to the attack every time moment);

$n_{t < t_w}(t)$ – number of active agents in Botnet in time moment t after the DDoS attack started;
 $n_{t > t_w}(t)$ – number of shut down agents in Botnet at time moment t after the DDoS attack started;
 P – composite DDoS attack success probability;
 P_B – bandwidth depletion DDoS attack success probability;
 P_C – CPU work exhaustion DDoS attack success probability;
 P_{fmr} – probability of filter false negative in router;
 P_{fns} – probability of filter false negative in server;
 P_{fpr} – probability of filter false positive in router;
 P_{fps} – probability of filter false positive in server;
 P_{im} – packet blocking probability because of insufficient memory;
 P_M – memory exhaustion DDoS attack success probability;
 P_{mt} – DDoS attack mitigation probability;
 P_{pr} – DDoS attack prevention probability;
 P_{pv} – DDoS attack postvention probability;
 P_S – composite DDoS attack survivor probability;
 P_{TA} – attack organizer trace back probability;
 p_{ta} – attack organizer traceability probability analysing active agent;
 $p_{Ta}(t)$ – attack organizer traceability probability function for one Botnet agent;
 p_{tn} – attack organizer traceability probability analysing not otherwise existing agent;
 P_{ub} – packet blocking probability because of unavailable bandwidth;
 r_b – number of packets in bot session to set up the session;
 r_n – number of packets in legitimate user session to set up the session;
 S – number of CPU's in the system;
 s_b – average number of packets in one bot session;
 s_n – average number of packets in one legitimate user session;
 T – channel throughput (bits/s);
 T_a – attack traffic;
 t_{aar} – attack agent recruitment time;
 t_{alt} – average life time for one agent;
 t_{as} – average time (s) of bot session data storage in the memory buffer;
 t_{gs} – average time (s) of general (both legitimate user and bot) session data storage in the memory buffer;
 T_n – legitimate users traffic;
 t_{ns} – average time (s) of legitimate user session data storage in the memory buffer;
 t_{tr} – time, required to execute a successful trace back operation (s);

t_w – time limit which the user is ready to wait for a job will be served;
 v_a – attack traffic average packet size (bits);
 v_b – average packet size (bits), generated by attack bot;
 v_g – general (both attack and normal) traffic average packet size (bits) after filtering;
 v_n – normal traffic average packet size (bits);
 β – Pareto distribution probability density function parameter;
 λ – query arrival rate;
 λ_{ac} – average attack packet arrival rate;
 λ_{ap} – attack traffic average packet arrival rate (packets/s);
 λ_{ap}' – attack traffic average packet arrival rate (packets/s) after filtering;
 λ_{ap}' – attack traffic average session arrival rate (sessions/s) after filtering;
 λ_{as} – attack traffic average session arrival rate (sessions/s);
 λ_{bc} – bot average packet arrival rates;
 λ_{bp} – average packet arrival rate (packets/s), generated by one attack bot;
 λ_{bs} – average session arrival rate (sessions/s), generated by one attack bot;
 λ_{gc} – general job arrival rate;
 λ_{gp} – general (both attack and normal) traffic average packet arrival rate (packets/s) after filtering;
 λ_{gs} – general (both attack and normal) traffic average session arrival rate (sessions/s) after filtering;
 λ_{nc} – legitimate user average packet arrival rates;
 λ_{np} – normal traffic average packet arrival rate (packets/s);
 λ_{np}' – normal traffic average packet arrival rate (packets/s) after filtering;
 λ_{np}' – normal traffic average session arrival rate (sessions/s) after filtering;
 λ_{ns} – normal traffic average session arrival rate (sessions/s);
 λ_s – average service job arrival rate in the system;
 μ – query service rate;
 μ_{cc} – average CPU service rate;
 μ_{cp} – average channel service rate (packets/s);
 μ_{gp} – average link (combining all channels together) service rate (packets/s);
 $\rho_n(t)$ – proportion of attack query volume in all incoming queries at time moment t ;

Abbreviations

BGP – Border Gateway Protocol;

C&C – Command-and-control;

CAPTCHA – Completely Automated Public Turing test to tell Computers and Humans Apart;

CCITT – Consultative Committee on International Telegraphy and Telephony;

CPU – Central Processing Unit;

DDoS – Distributed Denial of Service;

DoS – Denial of Service;

fBm – Fractional Brownian Motion;

FCFS – First Come First Serve;

GPRS – General Packet Radio Service;

IP – Internet Protocol;

IRC – Internet Relay Chat;

IS – Information System;

ISDN – Integrated Services Digital Network;

ITC – Information Technology and Communication;

JFK – Just Fast Keying;

LAN – Local Area Network;

NS – Network Simulator;

NS-2 – Network Simulator;

P2P – Peer-to-peer;

PSTN – Public Switched Telephone Network;

SPPA – Security Protocols Process Algebra;

SQL – Structured Query Language;

TCP – Transmission Control Protocol;

UMTS – Universal Mobile Telecommunications System.

VaR – Value at Risk;

VBR – Variable Bit Rate;

WAN – Wide Area Network;

X.509 – Standard for a public key and privilege management infrastructure;

X.800 – Security architecture for Open Systems Interconnection for CCITT applications;

Contents

INTRODUCTION	1
The Investigated Problem.....	1
Importance of the Thesis	2
The Object of Research	3
The Goal of the Thesis	3
The Tasks of the Thesis	3
Research Methodology.....	3
Importance of Scientific Novelty	4
Practical Significance of Achieved Results	4
The Defended Statements	4
Approval of the Results	5
Dissertation Structure	5
Acknowledgements	5
1. ANALYSIS OF DDOS ATTACKS, THEIR COUNTERMEASURES AND MODELLING	7
1.1. DDoS Attack	8
1.1.1. Botnet.....	8
1.1.2. TCP SYN Attack.....	12
1.1.3. Smurf Attack	13
1.1.4. Nuke Attack	14
1.1.5. SQL Wildcard Attack.....	14

1.1.6. Taxonomies of DoS Attacks	15
1.2. Taxonomies of DoS Attack Countermeasures	19
1.3. Computer Network Modelling	23
1.3.1. Traffic Modelling	23
1.3.2. Types of Models	27
1.4. DoS and DDoS Attack Modelling	28
1.4.1. Cost-based Models	28
1.4.2. Game-based Models	29
1.4.3. Rewrite Theory Models	30
1.4.4. Mathematical Models	31
1.5. Research on Internet Security in Lithuania	33
1.6. Conclusions for Chapter 1	34
 2. NEW TAXONOMIES OF DOS ATTACKS AND THEIR COUNTERMEASURES..	
.....	37
2.1. Proposed DoS Attacks Taxonomy	37
2.2. Proposed DoS Attack Countermeasures Taxonomy	42
2.3. Conclusions for Chapter 2	46
 3. COMPOSITE DDOS ATTACK MODEL	47
3.1. Bandwidth Depletion DDoS Attack Model	47
3.2. Memory Exhaustion DDoS Attack Model	51
3.3. CPU Work Exhaustion DDoS Attack Model	53
3.4. Composite DDoS Attack Model	56
3.4.1. CPU Work Exhaustion Model Modifications to be Used with Bandwidth Depletion DDoS Attack Model	57
3.4.2. Memory Exhaustion Modifications to be Used with Bandwidth Depletion DDoS Attack Model	57
3.5. Conclusions for Chapter 3	58
 4. EXPERIMENTAL RESULTS OF PROPOSED TAXONOMIES AND MODELS .	59
4.1. Comparison of Characteristics of DoS Attack and their Countermeasure Taxonomies	59
4.1.1. Analysis of DoS Attack Taxonomies	60
4.1.2. Experiment Results of DoS Attack Taxonomy Usage	61
4.1.3. Analysis of DoS Attack Countermeasure Taxonomies	63
4.1.4. Experiment Results of DoS Attack Countermeasure Taxonomy Usage	64
4.2. DoS Attack Model Validation Experiments	66
4.2.1. Experimental Situation and Validation Environment	66
4.2.2. Bandwidth Depletion Model VS Bandwidth Utilization	69
4.2.3. CPU Work Exhaustion Model VS Processing Time	72
4.2.4. Memory Exhaustion Model VS Calculated Comparison Value	73
4.3. Conclusions for Chapter 4	74

5. EXPERIMENTS WITH COMPOSITE DDOS ATTACK MODEL	77
5.1. Experiments with Memory and CPU Work Depletion DDoS Attack Model ..	77
5.2. Experiments with Composite Bandwidth and Memory Exhaustion DDoS Attack Model	81
5.3. Experiments with Bandwidth Exhaustion DDoS Attack Survive Model	85
5.3.1. Attack Postvention Probability.....	85
5.3.2. Attack Mitigation Probability.....	86
5.3.3. DDoS Prevention Probability.....	88
5.3.4. Research of Botnet Agent Allocation Strategies in DDoS Attack	88
5.4. Conclusions for Chapter 5	93
GENERAL CONCLUSIONS	95
REFERENCES	97
THE LIST OF SCIENTIFIC AUTHOR'S PUBLICATIONS ON THE SUBJECT OF THE DISSERTATION.....	107

List of Figures

Fig. 1.1. Ways of malware download separate from the exploit delivery, when the malware code is downloaded from the attacker (Mirkovich <i>et al.</i> 2005)	10
Fig. 1.2. Ways of malware download separate from the exploit delivery when the malware code is downloaded from central source (Mirkovich <i>et al.</i> 2005).....	10
Fig. 1.3. Command and control agent management scheme (Juknius and Goranin 2010)11	
Fig. 1.4. Peer-to-peer agent management scheme (Juknius and Goranin 2010)	12
Fig. 1.5. Hybrid peer-to-peer agent management scheme (Wang <i>et al.</i> 2007).....	12
Fig. 1.6. Package flow in three-way handshake (a) and TCP SYN attack (b) (Thing <i>et al.</i> 2007).....	13
Fig. 1.7. A Smurf Attack underway against a network (Pastore and Dulaney 2006).....	14
Fig. 1.8. DoS attack taxonomy proposed by D. Karig (Karig and Lee 2001).....	16
Fig. 1.9. DoS attack taxonomy proposed by A. Fadlallah (Fadlallah and Serhrouchi 2005).....	16
Fig. 1.10. DoS attack taxonomy proposed by C. Douligeris (Douligeris and Mitrokots 2004).....	17
Fig. 1.11. DoS attack taxonomy proposed by J. Mirkovic (Mirkovic and Reiher 2002) .	18
Fig. 1.12. DoS attack taxonomy proposed by S. Specht (Specht and Lee 2004).....	18
Fig. 1.13. DoS attack countermeasure taxonomy proposed by D. Kagir (Karig and Lee 2001).....	19
Fig. 1.14. DoS attack countermeasure taxonomy proposed by A. Asosheh (Asosheh and Ranezani 2008)	20

Fig. 1.15. DoS attack countermeasure taxonomy proposed by M. Specht (Specht and Lee 2004).....	21
Fig. 1.16. DoS attack countermeasure taxonomy proposed by C. Douligeris (Douligeris and Mitrokots 2004)	21
Fig. 1.17. DoS attack countermeasure taxonomy proposed by A. Fadlallah (Fadlallah and Serhrouhmi 2005).....	22
Fig. 1.18. DoS attack countermeasure taxonomy proposed by J. Mirkovic (Mirkovic and Reiher 2002)	23
Fig. 1.19. Traffic modelling levels (Pearson <i>et al.</i> 1996)	24
Fig. 1.20. Poisson process and real traffic comparison in different time periods (Leland <i>et al.</i> 1994).....	25
Fig. 2.1. New proposed DoS attack taxonomy	38
Fig. 2.2. New proposed DoS attack countermeasure taxonomy	43
Fig. 3.1. Illustration of bandwidth depletion DDoS attack formal model	48
Fig. 3.2. Example model of memory exhaustion DDoS attack	52
Fig. 3.3. Principe of FCFS scheduling algorithm	53
Fig. 3.4. Principle of Round-Robin scheduling algorithm	54
Fig. 3.5. Principle scheme of the composite DDoS attack	56
Fig. 4.1. Statistical data for the analysed DoS attack taxonomies	60
Fig. 4.2. Category minuteness level for analysed DoS attack taxonomies	61
Fig. 4.3. Number (a) and percentage (b) of assigned categories in certain DoS attack taxonomy for different type of DoS attack	62
Fig. 4.4. Statistical data for analysed DoS attack countermeasure taxonomies	63
Fig. 4.5. Category minuteness level for DoS attack countermeasure taxonomies	64
Fig. 4.6. Number (a) and percentage (b) of assigned categories in certain DoS attack countermeasure taxonomy for different type of DoS attack	64
Fig. 4.7. Modelled situation map in OPNET IT Guru tool	66
Fig. 4.8. DDoS attack success probability and link utilization comparison in “No attack” situation	70
Fig. 4.9. DDoS attack success probability and link utilization comparison in “Attack 1” situation	70
Fig. 4.10. DDoS attack success probability and link utilization comparison in “Attack 2” situation	71
Fig. 4.11. Bandwidth depletion success vs link utilization	71
Fig. 4.12. Request processing time comparison in our and OPNET model.....	73
Fig. 4.13. Dependency of memory exhaustion probability and calculated value.....	74
Fig. 5.1. Dependency of number of queries in the buffer in time, when incoming traffic is bigger than the maximum buffer size.....	78
Fig. 5.2. Average request service time changes in time, when the incoming traffic is a little bit bigger than the buffer size.....	79
Fig. 5.3. Changes of the composite resource depletion DDoS attack success probability and its components in time, when the attack incoming rate is constant.....	79
Fig. 5.4. Comparison of the queue size and incoming traffic rate in time using the increasing incoming traffic rate	80

Fig. 5.5. Changes of the average service time in time, using the increasing incoming traffic rate	80
Fig. 5.6. Changes of the composite resource depletion DDoS attack success probability and its components in time, when the attack incoming rate is increasing	81
Fig. 5.7. Conceptual model of the composite bandwidth and memory exhaustion DDoS attack.....	82
Fig. 5.8. Attack success dependency on the attack query filtering percentage	83
Fig. 5.9. Attack success probability depending on the attack query service time	83
Fig. 5.10. Attack success probability dependency on the query arrival rate with the average service time	84
Fig. 5.11. Attack success probability dependency on the query arrival rate with very short service time	84
Fig. 5.12. Changes of number of the bots in DDoS attack when different agent recruitment strategies are used.....	89
Fig. 5.13. DDoS attack survival probability and its component probabilities changes in time when no agent recruitment is used.....	91
Fig. 5.14. DDoS attack survival probability and its component probabilities changes in time when attack adds constant number of agents every second	91
Fig. 5.15. DDoS attack survival probability and its component probabilities changes in time when every time of moment the increasing number of agents is added to the attack.....	92
Fig. 5.16. DDoS attack survival probability and its component probabilities changes in time when every time of moment the decreasing number of agents is added to the attack.....	92

List of Tables

Table 1.1. Scan strategies used by different worms.	9
Table 1.2. Relationship between different modelling techniques.....	28
Table 4.1. Classified attack statistics for different DoS attack type and taxonomy	62
Table 4.2. Suggested attack countermeasures statistics for different DoS attack type and taxonomy.....	65
Table 4.3. Parameters of three experimental model situations	67
Table 4.4. Request processing time: comparison of the proposed model and OPNET tool	72
Table 4.5. Examples of the memory exhaustion DoS attack experiment data	74
Table 5.1. Comparison of DDoS effectiveness in different time intervals and using different agent allocation strategies.....	90

Introduction

The Investigated Problem

Information technology and communication (ITC) provides new possibilities, however, it also brings new concerns, which must be taken into account. One of the main ITC issues is information security. ITC provides the easier data exchange, search opportunities and makes the information more accessible and public to the world. This is why the information secrecy is not enough to ensure its security.

Modern ITC services and systems should ensure realisation of all the components of information security (confidentiality, integrity, accessibility, authenticity, non repudiation, access control). However, in particular situation some security components can be more important than the other ones. In network system case the accessibility can inflict the quality of service and even the customer opinion on the service provider. Therefore for service quality insurance and service provider's reputation maintenance one of the most important security issue should be accessibility.

System accessibility estimation models could help to prepare for potential threads. In this work Distributed Denial of Service (DDoS) attacks are analysed

and a new model for estimation of the computer system accessibility is proposed. This model should allow predicting the probability of different characteristic DDoS attack success and victim's probability to resist certain DDoS attacks.

Importance of the Thesis

A saying "If You are not in online, You are not at all" becomes very popular in society and business field. Computer networks allow proliferation of information, as well as using and providing different services for other computer systems. Such distributed systems or services supply fast and convenient way to access information, remotely manage processes or objects. Therefore business marketing becomes more oriented to internet users and services.

The other tendency can also be noticed. The main goal of hackers is not just recognition, cyber attacks become a way of making money. One of ways to make money from cyber attack is DDoS attack. During it a certain service is made unavailable for its legitimate users. Promising to cancel (sometimes even not starting at all) the attack, a hacker requires certain amount of money from the victim or competitors of a certain internet service, and it agrees to pay money for elimination of the competitor for a certain time of moment. There are many ways to make money from the cyber attack. However, accessibility can be one of the most expensive properties to ensure.

The other accessibility assurance problem is the opportunity to execute DDoS attacks easily. It is so because of variety of DDoS attack tools, Botnet existence and borrowing opportunity etc. The DDoS attack nature and popularity of its execution tools allows execution of such a type of cyber attacks even by amateurs. This is why system administrators should prepare carefully to ensure the systems accessibility on different levels.

However there are no unified measures to show systems resistance to DDoS attack. System administrators should rely on its experience and own knowledge to decide the effectiveness of different countermeasures and DDoS attack sizes. When client wants to check what the service accessibility rate is, the client must rely on service provider's opinion which sometimes can be subjective and not reliable.

Having model for network system DDoS attack success measurement there would be possible to estimate potential threats, losses, countermeasure effectiveness. It would allow modelling different situations before the attack appears and optimize the systems accessibility.

The Object of Research

Object of this dissertation is estimation of the probability of denial of service attack success.

The Goal of the Thesis

The main objective is creation of the model for composite DDoS attack success estimation.

The Tasks of the Thesis

In order to achieve the objective, it is necessary to deal with the following issues:

1. To analyze existing DoS attacks and their countermeasure taxonomies.
2. To analyze existing DoS and DDoS attack models.
3. To derive models for different type DDoS attacks.
4. To combine the created models in order to represent composite DDoS attack success probability.
5. To validate the proposed model.
6. To develop the prototype and perform experiments on several DDoS attacks.

Research Methodology

In order to achieve the objective, the following research methods were used:

- Comparative research and literature research for DoS attacks and their countermeasure properties, taxonomies, model types, comparison of existing models and judgement.
- The analysis results were summarized and the approach was expounded using the research generalization and logical induction methods.
- Mathematical modelling methods were used for DDoS attack property estimation.
- Proposed DDoS attack models relay on queuing theory and probability theory was used to derive the composite DDoS attack probability too.

Importance of Scientific Novelty

In preparation of this thesis these aspects of scientific novelty appeared:

1. New DoS attack and their countermeasure taxonomies where proposed.
2. Central processing unit (CPU) performance depletion model was proposed.
3. Composite DDoS attack model was proposed to estimate different type DDoS attack success probability and/or victim capacity to resist a certain size of DDoS attack.

Practical Significance of Achieved Results

Network system accessibility assessment is usually based on the expert opinion. There is no complete methodology to estimate the resistance to different type of DoS attacks. In this work, new DoS attack and their countermeasure taxonomies are presented for classifying the new type of DoS attacks or their countermeasure.

Proposed composite DDoS model can be used to analyse and predict DDoS attack success probability for different attack and victim properties. It can be as well used to find the optimal system configuration to ensure its accessibility. In case of wide acceptance of this model defined additional certification procedures the proposed model could serve as a tool for unified service accessibility rate estimation.

The research presented in the thesis leads to some practical recommendations. It describes which DDoS attack properties can pose the biggest threat, and which Botnet agent allocation strategies can cause the biggest effectiveness in different attack stages, etc.

The Defended Statements

1. Probabilistic modelling of DDoS attacks belonging to different types gives the basis to create the consolidated DDoS attack model generalizing the probability of the success attack.
2. Incorrect system configuration or unpredicted system usage by legitimate users of the system can result as DoS attack. Therefore DDoS attack success probability can be used creating prevention measures to improve internet service availability, despite the probability of real DDoS attack threat.

3. Multiple types of resources are exhausted during DDoS attack. Therefore, to achieve more precise DDoS attack probability, composite DDoS attack model should be used.
4. Composite DDoS attack success probability does not depend linearly on attack parameters and cannot be modelled by linear equations.

Approval of the Results

7 scientific publications were published: 6 in reviewed periodical journals (Ramanauskaitė and Čenys 2009, 2011a, 2011b, 2012; Ramanauskaitė 2010; Ramanauskaitė and Juknius 2010) and 1 in the reviewed international conference proceedings (Ramanauskaitė and Čenys 2011c).

The author has made 5 presentations at 5 scientific conferences (2 international):

- The 17th International Conference on Information and Software Technologies „Information Technologies 2011“, Kaunas, Lithuania.
- The 14th Conference for Lithuania Junior Researchers SCIENCE FOR FUTURE, 2011, Vilnius, Lithuania.
- International Young Scientists Conference and Erasmus Staff Mobility Event 2010, Šiauliai, Lithuania.
- The 13th Conference for Lithuania Junior Researchers SCIENCE FOR FUTURE, 2010, Vilnius, Lithuania.
- The 12th Conference for Lithuania Junior Researchers SCIENCE FOR FUTURE, 2009, Vilnius, Lithuania.

Dissertation Structure

In general, the dissertation is composed of introduction, five main chapters and general conclusions. The total scope of dissertation is 109 pages, including 42 equations, 55 pictures and 8 tables.

Acknowledgements

I am grateful to my scientific supervisor Prof Dr Habil Antanas Čenys for guiding and helping me in my studies and preparation of the thesis. I would also like to thank Dr Nikolaj Goranin for helping me in my doctoral studies.

I am grateful to the academic staff of Vilnius Gediminas Technical University and many of my colleagues for help and valuable ideas. I would also like to thank Šiauliai University and the European Commission FP7 project

INstitutional Transformation for Effecting Gender Equality in Research (INTEGER), Project No. 266638) for the financial assistance and stimulation.

This thesis would not have been possible unless support from my family and friends.

Analysis of DDoS Attacks, Their Countermeasures and Modelling

Information society is progressing every day and becomes more relying on information systems. However, this dependency can cause huge problems if a certain ITC system suddenly would stop providing the necessary services. If some component of this service would be a part of the other distributed system, the denial of service would influence the other systems and users. For a while DoS was not considered as separate security issue and was treated as a result of unavailability to manage confidentiality or integrity problem (Gligor *et al.* 2001). However, the rate and scope of the damage of DoS attacks show that DoS attacks are important and should be treated as a security problem. Understanding of DoS attacks as separate security problem rose with the popularity of distributed systems and after the paper of Gligor was published in 1986 (Gligor 1986).

In this chapter, DDoS attack definitions, Botnet characteristics, existing DDoS and their countermeasure taxonomies, the workload characteristics in communication systems and techniques for modelling performance are analyzed.

Analysis presented in this Chapter was published in (Ramanauskaitė and Čenys 2009, 2011b)*.

* The references are given in the list of publications by the author on the topic of the dissertation

1.1. DDoS Attack

Denial of Service attack has many different definitions. The International Telegraph and Telephone Consultative Committee (CCITT) in recommendation X.800 (CCITT X.800, 1991) describes denial of service as “The prevention of authorized access to resources or the delaying of time-critical operations”. The Committee on National Security Systems (Committee on National Security Systems 2010) adds a note to this definition that “Time-critical may be milliseconds or it may be hours, depending upon the service provided”.

Meanwhile Software Engineering Institute (CERT 1999) accents the goal of the attack and defines that “The primary goal of the attack is to deny the victim(s) access to a particular resource”. The common security term dictionary of Bright Hub (Bright Hub 2010) in the definition of Denial of Service also states the goal of this attack, however, it in addition mentions that “A DoS attack is a method of overloading a network with connection requests to the point that it cannot function normally.”

However, in practice service can be denied not only by overloading system resources, but also by using virus, worm assaults (Mirkovich *et al.* 2005). In this work, there will be no analysis of this part of denial of the service and the focus will be concentrated on Denial of Service attacks where attacker lies on a huge amount of sent traffic to attack victim causing its Denial of Service because of the sent traffic amount.

To achieve a bigger attack traffic size, multiple attack sources can be used (Fadlallah and Serhrouchi 2005). A Denial of Service technique that uses numerous hosts to perform the attack is called Distributed Denial of Service (DDoS) (Committee on National Security Systems 2010). It does not mean that an attacker owns a lot of computers to execute DDoS attack. Usually this type of attack uses an army of corrupted and controlled hosts – Botnets. Therefore to understand DDoS attacks better, Botnet principles should be known.

1.1.1. Botnet

Botnet is a collection of hosts, infected by malicious software (called robots, bots or agents) that runs autonomously and automatically without knowledge of the machine owner (Mahathi 2008). The infected machine gives its control to the bot creator (called bothead or botmaster) so the creator can use all controlled machines for a desired malicious purposes even without controlled machine user knowledge (Stone-Gross *et al.* 2009). It is safer and cheaper to use Botnet for attacker than own the same number of real host.

Analysing the life cycle of Botnet (Mirkovich *et al.* 2005; Lin and Chiueh 2006; Morparia 2008; Feily *et al.* 2009; Naseem *et al.* 2010) there are a few main stages:

- Scanning for potentially vulnerable machines.
- Compromising of chosen machines.
- Delivery of malware.
- Connection establishment with Botnet.

The scanning usually is done by already existing bots in the Botnet, not the botmaster itself. Its purpose is to find out as much as possible potentially vulnerable machines. Therefore couple of scanning techniques are used (Lin and Chiueh 2006; Gu and Liu 2008; Staniford *et al.* 2002):

- Random scan – randomly chosen IP addresses.
- Hitlist scan – existing list of vulnerable IP addresses.
- Rout-based scan – gathering of existing IP addresses information using BGP routing prefixes.
- Divide-and-conquer scan – many bots scan different IP space.
- Local preference scan – all bots scan IP addresses from its local network.
- Topological scan – information gathering from bots data left by its user and using its machine.
- Permutation – all bots share a common pseudo random permutation of the IP address space.

The scanning is one of the most important phases in Botnet recruitment so different worms use different scanning strategies (Zou *et al.* 2006). Scan strategies used by different worms are summarized in Table 1.1.

Table 1.1. Scan strategies used by different worms.

Worm \ Scan strategy	Random	Hitlist	Rout-based	Divide-and-conquer	Local preference	Topological	Permutation
Slammer (Moore <i>et al.</i> 2003)	✓						
CodeRed II (Moore <i>et al.</i> 2002)	✓				✓		
Witty (Kuma <i>et al.</i> 2005)	✓	✓					
Warhol (Weave 2002)		✓					✓
Flash (Zou <i>et al.</i> 2006)		✓		✓			
Routing slammer worm (Zou <i>et al.</i> 2006)			✓				
Morris (Eichin and Rochlis 1989)						✓	

When the IP address list is gathered (or even on the fly of IP address search) all IP addresses are tested for vulnerabilities they could be exposed to. If vulnerabilities are found and IP addresses are exposed to them, the next step is to download the bots malware.

Usually exploit and encoded malware are packaged in a single script and delivered similarly like viruses, worms or Troy horses (Aycock 2006). However, Agabot started using separation between exploit and malware delivery. First of all the exploit is sent, then it opens a shell on the remote host and downloads the encoded malware over HTTP or FTP.

J. Mirkovich *et al.* (Mirkovich *et al.* 2005) distinguishes two basic types how malware can be downloaded using separated exploit and malware delivery. The first method would be to require the code from the original machine, which is infecting the target. However, if the exploit source was obtained, the infecting machine can be traced. On the other hand, just one bot would be traced back if the malware code download place information is deleted immediately after malware code delivery. (see Fig. 1.1).

Fig. 1.1. Ways of malware download separate from the exploit delivery, when the malware code is downloaded from the attacker (Mirkovich *et al.* 2005)

The second method suggests using a central source, used just for malware storage. This allows to secure the attackers information, however, if the information about the central source could be extracted the central source would be eliminated and the spread of Botnet would be stopped (see Fig. 1.2).

Fig. 1.2. Ways of malware download separate from the exploit delivery when the malware code is downloaded from central source (Mirkovich *et al.* 2005)

The main Botnet feature is ability to control all the bots to make a certain task together (Wang *et al.* 2007). Therefore the control network must be composed. Most common agent control mechanism is called command and control (C&C) (Morparia 2008) (see Fig. 1.3). It uses one or many hosts as central point to communicate directly with bots. These C&C servers receive commands from their botmaster and forward them to the other bots in the network (Wang *et al.* 2007).

Fig. 1.3. Command and control agent management scheme (Juknius and Goranin 2010)

Usually IRC (Internet Relay Chat) protocol is used to control the bots (Juknius and Čenys 2008). Botmaster sends commands to the C&C server, which transmits the commands to the chat room, while all bots always wait for commands in this room. This control network has a defined structure. However, it is quite vulnerable because of the centralized control mechanism. If the C&C servers were found and neutralized, all Botnet communication network would be lost.

To eliminate this weakness other communication architectures can be used – P2P (Peer to Peer). In this architecture, there is no centralized point for C&C because all nodes act both as client and server (Grizzard *et al.* 2007)(see Fig. 1.4). Therefore if one bot is neutralized impact to other bots is very small and Botnet stays working properly.

However, cyber attacks do not stop to evolve, as well as Botnet communication network architectures. Wang *et al.* proposes hybrid P2P Botnet communication architecture (Wang *et al.* 2007). It uses P2P for control bots communication, while each control bot communicates with certain number of the client bots. (see Fig. 1.5).

Fig. 1.4. Peer-to-peer agent management scheme (Juknius and Goranin 2010)

Fig. 1.5. Hybrid peer-to-peer agent management scheme (Wang *et al.* 2007)

Usually attacker uses multiple stepping stones to make the traceability more difficult. Therefore more complex agent management schemes should be used in practice.

1.1.2. TCP SYN Attack

TCP SYN attack is one of the most known and used resource depletion attacks. A SYN flood attack occurs during the three-way handshake that marks the onset of a TCP connection. In the three-way handshake, a client requests a new connection by sending a SYN packet to a server. After that, the server sends a

SYN/ACK packet back to the client and places the connection request in a queue. Finally, the client acknowledges the SYN/ACK packet.

If an attack occurs, however, the attacker sends an abundance of SYN packets to the victim, obliging it both to open a lot of TCP connections and to respond to them. Then the attacker does not execute the third step of the three-way handshake that follows, rendering the victim unable to accept any new incoming connections, because its queue is full of half-open TCP connections (Chang 2002). Mostly the attacker sends a spoofed package to victim, what causes that the SYN/ACK package is send completely to other host, which do not respond because did not sent any SYN packets to the victim.

Fig. 1.6. Package flow in three-way handshake (a) and TCP SYN attack (b) (Thing *et al.* 2007)

The standard actions to protect a system is to increase the buffer size, decrease the time period of keeping a pending connection in the SYN RECEIVED state in the queue, enable build-in protection mechanisms (Burdach 2010). The most known protection mechanisms are SYN cache (Lemon 2002), SYN cookie (Bernstein 1999), Synkill (Schuba *et al.* 1997) etc. However, all these countermeasures have limitations and must be choosen wisely to maximise the resistance to DDoS attacks.

1.1.3. Smurf Attack

Smurf attack uses a spoofed address. It sends a broadcast PING package with false IP address (which is set to be a victims IP address) to a certain network or even many networks. Accordingly all computers reply to this package to provided IP address, therefore all the flow goes to one host – victim (see Fig. 1.7). The incoming traffic can be so big that the victim (host or even network) cannot be able to serve all the traffic because its bandwidth is exhausted (Pastore and Dulaney 2006; Choudary and Shilpa 2011; Sonic WALL, Inc. 2001).

Fig. 1.7. A Smurf Attack underway against a network (Pastore and Dulaney 2006)

According to the investigation of the Software Engineering Institute (CERT 1998), there can be solutions to Smurf attack on 3 levels for intermediary, victim and the site where attacks originate. To prevent you be a part of Smurf attack IP-direct broadcast and respond to ICMP packets sent to IP broadcast addresses can be disabled. However, for victim and site where attacks originate there are no so effective solutions, and they can basically rely on the package filtering, which cannot be effective enough in many cases.

1.1.4. Nuke Attack

Nuke is one of the Denial of Service attacks that have already been in practice for the longest period. It uses modified ICMP packets and sends to the victim which is not capable to process what is sent correctly. The processing of modified ping utility requires more time, therefore the system slows down and even can stop completely.

To achieve the desired DoS effect with Nuke attack, many modified fragmented data should be sent to the victim. Nuke attacks also can be executed on instant messenger or online gaming systems (SecuringMyNetwork 2011).

1.1.5. SQL Wildcard Attack

Some of DoS attacks can be executed in the application layer and do not require protocol or package structure knowledge from the attacker. One of such an example is SQL wildcard usage to consume central processing unit (CPU) in database server. As F. Mavituna shows (Mavituna 2008), if web application uses search form, where LIKE queries with wildcards are used, regular user can search for a string “_aaaaaaaaa[! -z]@\$!_” instead of “foo” and it will cause the

database server to execute a SQL query much longer. If web applications use LIKE queries with multiple search terms, the sql query execution can take even longer.

Concerning this vulnerability, even group of users executing time consuming searches can cause problems for the system, and the same might be said about the huge number of agents striking at the same moment.

This kind of DoS attacks can be eliminated properly designing and implementing web applications: developers should avoid allowing SQL wildcards for regular users or use additional elements as implementing CAPTCHA or puzzle to slow the user down and prove its humanity (Mavituna 2008).

1.1.6. Taxonomies of DoS Attacks

Analysing why the DoS are available, there can be distinguished many DoS attack success reasons. The DoS effect even can be caused by system legitimate users even with no bad purposes. One of the last well known cases when legitimate users caused the systems to be denied of service appeared on the 14-th of March, 2011 during the first day of internet ticket selling to Eurobasket 2011 (Laikas.lt 2011). All users tried to buy the tickets as soon as possible and get the best seats. However, the system was not able to serve such a huge amount of requests. This influenced the decrease of service time and sometimes some users even were not served at all.

This is just one example when denial of service can rise not because of malicious actions. Meanwhile analysing how else DoS can be caused there are a lot of different ways. However, there is no one standard for classifying DoS and DDoS attacks.

D. Karig and R. Lee in their work “Remote Denial of Service Attacks and Countermeasures” (Karig and Lee 2001), classify DoS attacks considering the place where the attack strikes into five categories (see Fig. 1.8):

- Network device level.
- OS level.
- Application level.
- Data flood.
- Protocol feature attack.

Such a classification is simple and uniquely determines the location in classification i.e. protection of the installation object. However, it does not describe all attack properties which are important for the attack analysis and countermeasure selection, and this is because only Data flood and Protocol feature attack are divided into the smaller parts. However, it is not enough in order to represent whole DoS attack situation.

Fig. 1.8. DoS attack taxonomy proposed by D. Karig (Karig and Lee 2001)

A. Fadlallah and A. Serhrouchni (Fadlallah and Serhrouchni 2005) use a DoS attack taxonomy considering number of computers used to execute one attack. They split DoS attacks into two main categories (see Fig. 1.9):

- Single source attacks.
- Multiple source attacks.

Single source attacks are classified into protocol design exploit attacks and software bugs attacks. Meanwhile, the multiple source attacks are classified into chaotic (Worms) and structured (Distributed Denial of Source and Distributed Reflected Denial of Service).

Such a DoS attack taxonomy is simple and clearly defines the limits of DoS and DDoS attacks. However, it does not pay attention to additional attack features, such as what exact vulnerabilities are used.

Fig. 1.9. DoS attack taxonomy proposed by A. Fadlallah (Fadlallah and Serhrouchni 2005)

Works where only DDoS attack taxonomy is described are more comprehensive and classify more attack properties.

C. Douligieris and A. Mitrokotsa (Douligieris and Mitrokots 2004) do not distinguish DDoS attack target but focus on the attack description according to different approaches (see Fig. 1.10):

- Automation degree.
- Vulnerability exploitation.
- Attack rate dynamics.
- Impact.

According to such a taxonomy, each attack is described considering approaches of all four groups. This allows to determine the suitable countermeasure in all these categories and make one united solution to this type of attack.

Fig. 1.10. DoS attack taxonomy proposed by C. Douligieris (Douligieris and Mitrokots 2004)

J. Mirkovic and P. Reiher in their paper “A Taxonomy of DDoS Attack and DDoS Defence Mechanisms” (Mirkovic and Reiher 2002), use a similar DDoS attack taxonomy structure. They classify DDoS attack mechanisms into eight categories (see Fig. 1.11):

- Automation.
- Exploited weakness.
- Source address validity.
- Attack rate dynamics.
- Possibility of characterization.
- Persistence of agent set.
- Victim type.
- Impact on the victim.

This taxonomy is oriented to DDoS attack description from the attacker point of view. It is quite complete and suitable for DDoS attack representation

considering its properties; however, full (by all categories) attack description is complicated. This is because of the fact that the new DDoS attack modifications require deep analysis in order to determine some properties necessary for this classification.

Fig. 1.11. DoS attack taxonomy proposed by J. Mirkovic (Mirkovic and Reiher 2002)

Fig. 1.12. DoS attack taxonomy proposed by S. Specht (Specht and Lee 2004)

More oriented to the comprehensive and instantaneous determination of the characteristics of attack, is the attack taxonomy of S. Specht and R. Lee DDoS,

described in the article “Distributed Denial of Service: Taxonomies of Attacks, Tools and Countermeasures” (Specht and Lee 2004) (see Fig. 1.12). It focuses on how the attack strikes and what kind of victim weaknesses is exploited (bandwidth or resource depletion). But it does not pay attention to the additional attack properties, with the exception of the vulnerability type, which is used to cause DoS effect.

All analyzed taxonomies have specific properties considering it. However, J. Mirkovich (Mirkovic and Reiher 2002) proposed the most detail taxonomy.

1.2. Taxonomies of DoS Attack Countermeasures

Because of a huge DoS wavier in type, accordingly a lot of different type of countermeasures can be used to handle certain type of DoS. The classification of existing DoS attack countermeasures is important because of the possibility to gain the clear view of defence mechanisms, opportunity to choose one united defence mechanism from all the possible methods.

D. Kagir (Karig and Lee 2001) separates five DoS attack countermeasure groups which are oriented to the victim type (see Fig. 1.13):

- Network device level.
- OS level.
- Application level.
- Protocol standard level.
- Replication and load balancing (bandwidth).

More detailed description is possible for the subcategories of these five groups; however, it may be minimal because there are two main subcategories in these groups, i.e. iterative maintenance and usage of the additional systems.

Fig. 1.13. DoS attack countermeasure taxonomy proposed by D. Kagir (Karig and Lee 2001)

A. Asosheh (Asosheh and Ranezani 2008) proposed DDoS attack taxonomy which is also oriented to the victim type but is more detailed. First of all, there

are two main categories representing the possible countermeasure phases (see Fig. 1.14):

- Prevention.
- Detection.

These categories encompass the type of victim (target network and intermediate network for both phases and additional source network for the detection phase). In addition, each subcategory has its own and quite detailed classification.

Fig. 1.14. DoS attack countermeasure taxonomy proposed by A. Asosheh (Asosheh and Ranezani 2008)

The main disadvantage of this classification is not sufficient representation of possible countermeasure phases. M. Specht (Specht and Lee 2004) takes this into account and classifies six phases (see Fig. 1.15):

- Detect and neutralize handlers.
- Detect and prevent secondary victims.
- Detect and prevent potential attacks.
- Mitigate and stop attacks.
- Deflect attacks.
- Post-attack forensics.

Most of these phases have subcategories meant for more detailed DDoS countermeasure description. But this and all the other mentioned DoS countermeasure classifications focus only on one characteristic, i.e. DoS countermeasure may belong to only one category in this classification.

Fig. 1.15. DoS attack countermeasure taxonomy proposed by M. Specht (Specht and Lee 2004)

Fig. 1.16. DoS attack countermeasure taxonomy proposed by C. Douligieris (Douligieris and Mitrokots 2004)

Meanwhile, C. Douligeris (Douligeris and Mitrokots 2004) distinguish two criteria for DDoS countermeasure classification (see Fig. 1.16):

- Classification concerning activity.
- Classification concerning location.

Classification concerning location was not mentioned before and represents the possible location of DoS defence, i.e. in victims, source or intermediate network.

A. Fadlallah (Fadlallah and Serhrouchi 2005) and J. Mirkovich (Mirkovich and Reiher 2002) distinguish also the third criterion to classify DoS countermeasures, and it is classification concerning the cooperation degree (see Fig. 1.17):

- Autonomous.
- Cooperative.
- Interdependent.

Fig. 1.17. DoS attack countermeasure taxonomy proposed by A. Fadlallah (Fadlallah and Serhrouchi 2005)

The main difference between these two taxonomies is the fullness. A. Fadlallah (Fadlallah and Serhrouchi 2005) separates a few subcategories for all criteria. Meanwhile, J. Mirkovich (Mirkovich and Reiher 2002) taxonomy is more detailed (see Fig. 1.18). It even divides the prevention phase into attack and DoS prevention.

Fig. 1.18. DoS attack countermeasure taxonomy proposed by J. Mirkovic (Mirkovic and Reiher 2002)

The DoS, DDoS countermeasure taxonomies are more sensitive for aging than DoS, DDoS taxonomies. It has more specific properties of countermeasure, therefore if new ideas for DoS countermeasure would rise, it may not be represented in current taxonomies.

It is possible to distinguish all possible combinations of DDoS attacks and their countermeasures properties when detailed taxonomy exists. However to get quantitative expressions of analysed properties DDoS attack or their countermeasure should be evaluated in practice or modelled by appropriate tools.

1.3. Computer Network Modelling

Model means “a systematic description of an object or phenomenon that shares important characteristics with the object or phenomenon (Houghton Mifflin Company 2005). Using models we can easier understand the analyzed object by viewing it in different situations and parameters.

Modelling a computer network two main components have to be taken into account. It is the system itself and the incoming traffic, which has to be processed by the system.

1.3.1. Traffic Modelling

Thomas M. Chen (Chen 2007) states that communication system can be viewed focussing on different levels: communication, burst and cell or packet. It demonstrates different views to computer communication process. On the

session level, only the total connection time is important, and no traffic details. Meanwhile, packet level analysis encompasses all packets sent/received during certain time. However (Pearson *et al.* 1996) distinguished more detailed traffic modelling levels (see Fig. 1.19).

Fig. 1.19. Traffic modelling levels (Pearson *et al.* 1996)

The highest abstraction level stays Session level and each session can be composed of many bursts, in this classification called Busy periods. Comparing with Thomas M. Chen (Chen 2007) proposed levels, Pearson *et al.* (Pearson *et al.* 1996) suggests taking into account OS (Operation System) level, which would represent the operating system activity. In each OS level burst several PCT/IP packet should be processed, which in the case that the workstation is attached to an ATM network will generate a burst of ATM cells, not mentioned in T. M. Chen (Chen 2007) classification.

Depending on traffic modelling level, different specification level as well as according traffic pattern is needed.

The Poisson models are the most longstanding traffic models (Frost and Melamed 1994) and have been used for computer network models since 1980 (Marathe and Hawe 1982; Ramakrishnan and Tripathi 1982). It allows representing telephone line models, where arrival process is described by Poisson process and has exponential durations (Paxson and Floyd 1995). However, computer networks evolved and according to many different researches (Puigjaner 2003; Jain and Routhier 1986; Leland *et al.* 1993; Paxson and Floyd 1995), nowadays Erlang models do not represent the Internet traffic precisely and can be used on the session level, meanwhile the burst and packet

levels require more detailed analysis of the traffic activity states and their intrinsic characteristics for model composition.

The traffic scaling behaviour, or scale invariance, in packet data was first observed by Leland *et al.* for Ethernet Local Area Network (LAN) traffic (see Fig. 1.20), and then for Variable Bit Rate (VBR) video (Beran *et al.* 1997), ATM cell traffic (Jerkins and Wang 1997), and Wide Area Network (WAN) traffic (Paxson and Floyd 1995), etc.

Fig. 1.20. Poisson process and real traffic comparison in different time periods (Leland *et al.* 1994)

All these analyses and work done by Riedi and V'ehel (Riedi and Vehel 1997; Vehel and Riedi 1997) showed that on the package level and on time scales smaller than 1s, wide area traffic was consistent with multifractal behaviour, which is inflicted by traffic burstness and package autocorrelation.

For internet traffic generation in package or cell level many solutions are presented. Vern Paxson (Paxson and Floyd 1995) proposed a sage of Pareto

model, where inter-arrival time is with $\beta \approx 1$. Such a model generates a very similar distribution to the real network traffic. However, the Pareto distribution is not self-similar, and therefore this model has its own limitations. It represents self-similarity in short terms and heavy-tailed Pareto model would recognize the possibility of heavy traffic after a period of low traffic.

Ilkka Norros (Norros 1994) in 1994 proposed to use fractional Brownian motion (fBm), which is a case of continuous-time Gaussian process to represent self-similarity in modelled network traffic. This model is continuous, however, it has Hurst parameters for a self-similar flow which makes it difficult to apply in practice.

The estimation of self-similarity parameters can require a big amount of observed data and calculations. Patrice Abry and Darryl Veitch (Abry and Veitch 1998) proposed a wavelet-based tool for the analysis of long-range dependence and a related semi-parametric estimator of the Hurst parameter. This model uses the wavelet based Hurst parameters estimation, because it naturally matches the structure of the fBm process (although fBm is no stationary, its wavelet transform is stationary; even if fBm is long range dependant, its wavelet coefficients are almost uncorrelated) (Jacquet and Harba 2004). Therefore the traffic modelling with fBm becomes easier.

Ashok Erramilli *et al.* (Erramilli *et al.* 2002) proposed to use chaotic maps for network traffic modelling. Chaotic maps can be thought of as a form of continuous-state Markov chains. To represent realistic results, authors suggest using two chaotic maps (one for the window size and one for the ON/OFF state) and do not take into account retransmission of lost packets or the retransmit timeout.

Different network traffic generation approach is proposed by Kashi Venkatesh Vishwanath and Amin Vahda (Vishwanath and Vahdat 2006). In their paper “Realistic and Responsive Network Traffic Generation” a closed-loop, network-responsive traffic generator Swing is described. It accurately captures the packet interactions of a range of applications using a simple structural model, automatically extracts distributions for user, application, and network behaviour and then generates live traffic corresponding to the underlying models in a network emulation environment running commodity network protocol stacks.

NS-2 simulator uses PackMimeHTTP which is a web traffic generator proposed by JinCao *et al.* It uses discrete Weibull distribution, a probability distribution on the positive integers, to model the run length distributions. It uses connection level for network traffic simulation, however, it both captures relationships and dependencies not present in package based models and it is an approach that is more likely to scale to modelling the traffic generated by other application classes such as peer-to-peer file sharing traffic (Cao *et al.* 2004).

1.3.2. Types of Models

Constructing models in telecommunication area, a high-level language, usually Process Algebras, or high-level structure, such as a Queuing Network or a Petri net, can be used to represent the situation and keep it simple enough to be analysed (Field and Bradley 2004).

According to W. Fokkink, process algebra is a mathematical framework in which system behaviour is expressed in the form of algebra terms, enhancing the available techniques for manipulation (Fokkink 2007). In comparison, Petri net and Queuing Networks rely on graphical representation for more detailed understanding.

As R. Puigjaner (Puigjaner 2003) states, the simulation can be used almost universally for its easy and intuitive understanding. However, using Markov chains, it has a limitation of 10^8 states. Then the state number becomes too big to permit efficient modelling without using numerical solutions.

M. Heidari (Heidari 2006) distinguishes four model types: Deterministic, Stochastic, Rule-Based and Multi Agent.

Deterministic models: The processes of this model are often described by differential equations, and with a unique input leading to the unique output for well-defined linear models and with the multiple outputs possible for non-linear models; in these models, equations can be solved by different numerical methods.

Stochastic models: This type is used to model temporal behaviour phenomena with random components. In this model, unique input leads to different output for each model run, due to the random component of the modelled process, and single simulation gives only one possible result. All of the major models in the information system (IS) are stochastic models.

Rule based models: In this model, processes are governed by local rules using cellular automata. This type of models can be uncoupled with non-linear dynamic mathematical systems based on discrete time and space.

Multi-agent models: For modelling complex systems (including multi role, multi platform and multi system aspects) Multi-agent models can be used. In these models group of interacting agents must be developed. Agent is any actor in a system that can generate events that affect it and other agents, a typical agent is modelled as a set of rules.

In these mathematical formalisms, model types differ in purpose of usage. All analysed mathematical formalisms have an associated set of analytical techniques which can be used for their solution of performance metrics, as well as model type. It does not mean that it is tightly connected, but usually a specific mathematical formalism has its own way to possible solution for analysis or model type.

Table 1.2. Relationship between different modelling techniques

Representation	Solution	Model type
Processing algebra	Numerical, Simulation	Deterministic
Queuing network	Numerical, Markov chain, Simulation	Stochastic, Multi-agent
Petri net	Markov chain, Simulation	Rule based, Multi-agent

Analysing the relationship between mentioned modelling techniques and model types, we can state that there is no direct equivalent for each other. There can be noticed just commonly used combinations (see Table 1.2), which are more obvious for representation and model type. However, solution can vary very much.

1.4. DoS and DDoS Attack Modelling

DoS and DDoS attack modelling is one of computer network modelling cases. Different models of DoS or DDoS attack exists in the literature. However, there is no one prime model, which would elevate all DDoS attack issues and characteristics. Despite what is the incoming traffic pattern, representation or model type it uses, there could be different classifications. Here we present one used by us for better understanding of the existing models.

1.4.1. Cost-based Models

One of the first works formally characterizing DoS attacks in literature was Meadows's framework (Meadows 1999). She proposed a way to analyse protocol resistance to denial of service attacks using generic, cost-based approach. This framework requires identifying all actions in a protocol and assigning costs to it. Usually it is a computation cost or resource allocation, which should be combined and compared to the costs incurred by an attacker as a result of participating in the protocol.

For protocol examination, desired simulations should be executed to estimate if the DoS attack require more effort than a given threshold specified by toleration in the framework.

Not minding the difficulties in cost estimation for the appropriate actions (how to represent costs of engaging in individual actions such as computing digital signatures, storing data, impersonating principals, etc.), on the basis of this framework, couple experiments for certain protocol resistance to Denial of Service were published.

One of the usages was published by the framework author (Meadows 2001), where she used the Station to Station (Diffie *et al.* 1992) protocol as an example. This paper included a usage of Fail-stop protocol (Gong and Syverson 1998): “A protocol is fail-stop if any attack interfering with a message sent in one step will cause all causally-after messages in the next step or later not to be sent”, which brings more difficulty to this model.

The first detailed application of Meadows’s framework was represented by Smith *et al.* (Smith *et al.* 2006), where they used it to analyse an Internet key agreement protocol JFK (Aiello *et al.* 2004). Two models were represented to describe two different ways to make a DoS attack against JFK protocol.

S. Lafrance and J. Mullins (Lafrance and Mullins 2003) modified Meadows’s framework by proposing different protocol security estimation method. Meadows evaluate protocol vulnerability to DoS based on a tolerance relation (the amount of resources the designer is willing to spend to provide a given level of security). According to these authors, if the principal behaves differently when it is being attacked, we should conclude that the protocol is unsafe.

S. Lafrance and J. Mullins proposed a new process algebra case – Security Protocols Process Algebra (SPPA) – with extended local function calls, cryptographic primitives and special semantic features in order to handle cryptographic protocols. This notation can be used to illustrate the frameworks usage in Transmission Control Protocol (TCP).

Another framework inspired by Meadows was proposed by Z. Cao *et al.* (Cao *et al.* 2007). These authors disagree that DoS attacks can be represented without taking into account the stochastic nature. Therefore they introduce a probability model into Meadows cost-based framework and adopted a Value at Risk (VaR) model. The CCITT X.509 authentication protocol and its modified version with client puzzles was analysed using the proposed framework and proved the client puzzles effectiveness for DoS attack prevention in CCITT X.509 authentication protocol.

1.4.2. Game-based Models

Game-based models usually are linked with multi-agent solutions as one of its cases, where players of the game are represented as separate agents. Difference is just in agent usage way, because in this case they are treated as concurrent objects, which tries to win against each other. Such an idea is very common to security issues, where two opposite sides (attacker and victim) exist, as well as other situations where one object can cause behaviour of the other objects.

In computer network modelling different game-based approaches exist (Shenker 1994; Altman 1994; Maheswaran and Basar 1998; Hespanha and

Bohacek 2001; Lye and Wing 2002) while for DoS attack modelling one of the first papers was published by Bencsath *et al.* (Bencsath 2003) in 2003. Authors describe a client puzzle approach for DoS attack mitigation and distinguish two game players (attacker and victim) with couple of strategies for both of them and payoffs determined for the attacker only. The choice of strategy for player is determined by a probability distribution over the set of strategies available for the player and allows finding of the optimal strategy for the server in all conceivable cases.

Similar model was also proposed by Ajay Mahimkar and Vitaly Shmatikov (Mahimkar and Shmatikov 2005). They proposed a game-based analysis method and DoS attack prevention protocol, which protects against malicious bandwidth consumption by using adaptive client puzzles and pushing their generation to the intermediate router, and demonstrated it by applying to JFK protocol.

According to Mehran Fallah (Fallah 2010), these mechanisms have not been designed through formal approaches and thereby some important design issues such as effectiveness and optimality have remained unresolved. Therefore he proposed a series of optimal puzzle-based strategies for handling increasingly sophisticated flooding attack scenarios and for interaction between an attacker who launches a flooding attack and a defender who counters the attack using a puzzle-based defence modelled as an infinitely repeated game of discounted payoffs.

The game-based models usually are realized as agents and simulated to represent the modelled situation. In DoS attack models there are two types of agents – victim and attacker. However, Igor Kotenko (Kotenko 2005) suggests an approach to modelling and simulation of cyber-wars in Internet between the teams of software agents (community of agents cloned on various network hosts). The purpose of combining agents into teams is for representation of agent cooperation in the same group and completion between the opposite teams. In represented DDoS attack model, authors use three-level DDoS agent structure making the model more detailed.

1.4.3. Rewrite Theory Models

To represent how specific model object changes its states rewrite theory can be used. Taking into account the stochastic nature of computer networks, G. Agha *et al.* (Agha *et al.* 2005) proposed using probabilistic rewrite theory for DoS attack modelling. They distinguish three objects: attacker, host client and server (victim). Accordingly, each of these objects has its own set of attributes. Model uses seven probabilistic rewrite rules for object actions description to represent a TCP SYN DoS attack scenario representation. However, this model

uses scaled down parameters so that executed experiments might be completed in reasonable amount of time.

M. AlTurki *et al.* (AlTurki *et al.* 2009) used probabilistic rewrite rules to model Adaptive Selective Verification (ASV) protocol. They also used the Maude rewrite engine for performance of Monte Carlo simulations as well as VeStA tool for performance of the statistical model checking for used QuaTEX properties. These existing tools allowed to reduce the properties limitations and to execute more realistic DoS attack modelling.

However, M. Kim *et al.* (Kim *et al.* 2007) states, that these approaches do not provide any procedure by which they can determine the sample size required to achieve normality. Therefore they extended their approach by an on-demand sample generation that can compute the sample size sufficient to guarantee the normality of data, and utilize the normal distribution to obtain the error bound and confidence interval.

1.4.4. Mathematical Models

Both cost-based, game-based and rewrite theory models relay on programming approaches (simulation, multi agent, rule-based models). However, there exist mathematical DoS and DDoS models which can be used separately or combining with programming to produce model results without long lasting simulation or model execution.

One of the first papers mentioning mathematical DoS attack analysis was written by Chang (Chang 2002). He analysed DoS attack defence mechanisms and proposed to use queuing theory for TCP SYN attack representation to calculate the need of attack packets to exhaust the victim memory resources. However, this author uses G/D/ ∞ /N model (general arrival rate, deterministic service time, infinitive service number, queue length N) where threatening of the service time as deterministic is not very realistic.

More detailed models were proposed by Huang *et al.* (Huang *et al.* 2003). These authors in the paper “Analysis of a New Form of Distributed Denial of Service Attack” describe two models for different type of DDoS attack. One model represents TCP SYN attack, and the another – bandwidth exhaustion DDoS attack. For TCP SYN attack model Engset loss model (G/G/N – both arrival and service time is generally distributed, the queue can serve N queries and to save infinitive number of user requests) was proposed. This model allows finding the packet loss probability, and it is possible to use it as a DDoS attack success measurement.

For bandwidth depletion attack model Huang *et al.* (Huang *et al.* 2003) use a fluid model of statistical multiplexing of attack traffic from n agents, which alternate between the burst (on) state and the silence (off) state. Therefore the

proposed formula can be applied both for constant and variable attack traffic rate of one agent and present the approximate agents number needed to exhaust victim bandwidth.

Huang *et al.* also proposed and modified TCP SYN attack model for wireless networks (Huang *et al.* 2003). This model distinguishes legitimate and attack requests therefore use a generalized multi-class Erlang and Engset mixed loss model. For legitimate traffic the arrival is represented as Poisson process, while attack traffic arrives according to finite source model. This modification requires the number of attack agents to be sufficiently large, however, brings more realistic situation representation of resource exhaustion DDoS attack.

Mathematical resource exhaustion DoS attack models that were mentioned used general queries arrival rate distribution and the only Huang *et al.* in model for wireless networks (Huang *et al.* 2003) used Poisson process to represent legitimate users requests. Other authors suggest using exponential arrival rate distribution to represent the attack traffic either.

S. Khan and I. Traor (Khan and Traore 2005) suggest representing the inter-arrival and service time to have exponential distributions during attack as well. The proposed model concentrates on CPU work exhaustion, and therefore uses M/M/1/N model to represent that only one query can be served at one time moment, while N queries can be stored in the buffer. Using this model authors represent desired situation and using numerical methods get queue growth rate and response time metrics. These metrics can be used as a indicator for attack detection therefore are prepared to get by numerical calculations, but not the simulations, which takes more time to be executed.

A. Aissani (Aissani 2008) proposes TCP SYN attack model and distinguished legitimate and attack traffics. However, he also uses exponential distribution (Poisson process) for both legitimate and attack traffic arrival rate representation in this model. The separation of legitimate and attack traffic allows to represent the model more realistically and presents formulas for connection loss probability and the buffer occupancy percentage of half-open connections for both regular and attack traffic. Therefore these metrics can be used for DoS attack detection as well as TCP SYN attack success modelling and analysis.

Trying to denial a service for a long time, the attack optimization becomes more important. Therefore DoS attackers try to calculate what the optimal attack traffic is and choose a specific attack package sending pattern. This kind of DoS attack is called low-rate and modelling of it requires additional attack characteristics, describing the chosen pattern. G. Macia-Fernandez *et al.* (Macia-Fernandez 2006) proposes a model, which allows understanding the dynamics of this kind of attacks. Therefore they are disabled to use simple arrival process and take into account the burst characteristics of the incoming traffic. According to

attack waveform the attack effort, user perceived performance and mean idle time can be calculated using formulas represented in this paper. However, authors were not satisfied with this model because of unconsidered case of iterative servers with superposition among the occurrence probability functions or that of their functioning as a concurrent system. Therefore they extended this model into the new one (Macia-Fernandez *et al.* 2009) where these issues are resolved and new formulas for server availability, client success probability and traffic overhead are proposed.

However, sometimes low-rate DoS attack can be with randomly occurring outages. Therefore N. Chaturvedi *et al.* (Chaturvedi and Mohant 2011) uses alternation between the active and steady states of low-rate DoS attack bursts as a stochastic process. Since the duration of DoS burst is random and hence satisfying the Markovian property of memorylessness, the aforementioned conditional probability distribution is represented as an exponential one.

All mentioned mathematical DoS attack models represent DoS attack in higher aggregation level but transition between specific query states. However, there are mathematical models, which analyse DoS attack in more general form using Markov chains. One of the examples is Y. Wang *et al.* (Wang 2007) proposed DoS attack model where two-dimensional embedded Markov chain is used. It represents TCP SYN attack by using two dimensional Markov chain. Two dimensional Markov chain is used because of two different incoming traffics (attack and normal) and allows finding the stationary probability distribution used to get analytical results for a few security performance metrics: connection loss probability, buffer occupancy percentage of half-open connections for both regular and attack traffic.

K. Salah (Salah 2010) uses the embedded Markov chain to study and analyze the performance of rule-based firewalls when subjected to normal and DoS attacks. The Markov chain represents the transition between states of incoming query comparison with firewall filtering rules. It takes into account the required CPU work time and allows estimation of average packet delay and additional characteristics as packet loss or CPU utilization.

1.5. Research on Internet Security in Lithuania

There are a few academic communities in Lithuania, which are interested in information security researches. In Vilnius Gediminas technical university prof. A. Čenys is leading researches of security of information technologies, prof. A. Kajackas is leading researches of internet infrastructure resilience and quality of service, prof. J. Skudutis works in the field of cyber incidences defensive measures, prof. R. Baušys – in the field of image digital signature area. In Kaunas University of Technology investigations of cyber security is

done through the implemented CERT team and different researches in cryptography area are done by V. Stakėnas, A. Katvickis, E. Kazanavičius, K. Pakalniskis, E. Sakalauskas, D. Burba and A. Venčkauskas.

In internet security area different researches have been implemented. D. Puniškis and R. Laurutis (Puniškis and Laurutis 2007; Puniškis et al. 2006; Puniškis and Laurutis 2005; Laurutis 2003) published articles on neural network application for malware epidemics detection and spam filtering. N. Palauskas, E. Garšva and J. Skudutis performed different researches on computer attack detection, simulation and modelling (Paulauskas and Garsva 2006; Paulauskas 2009; Paulauskas et al. 2009; Paulauskas and Garsva 2008; Paulauskas and Skudutis 2008). N. Goranin investigated genetic algorithm usage for internet worm risk evaluation and evolution forecasting (Goranin and Čenys 2008a, 2008b, 2009). J. Juknius analysed Botnet propagation and detection algorithms (Juknius and Čenys 2008; Juknius and Goranin 2010). R. Rainys analysed the internet network infrastructure dependability. A. Kajackas *et al.* in their researches covered many concerns of internet quality of the service (Kajackas and Rainys 2011, 2010; Kajackas *et al.* 2011).

1.6. Conclusions for Chapter 1

1. There are many ways how to make any service unavailable for its legitimate users. DoS attacks can use communication protocols as well as system architecture vulnerabilities, which makes the combat of DoS attacks very difficult and resources consuming. Sometimes can be no ready suitable countermeasures to be used. As a result it is very important to know what the potential threats are.
2. There is no unique taxonomy for DoS. The review of existing DoS and DDoS taxonomies demonstrated that most of them do not allow full description of DoS attacks or even includes properties which can be known only by the attacker. Therefore none of existing taxonomies can be used carefree and fully classify incoming DoS attacks from the point of view of victim.
3. It is very difficult to model precisely computer network behaviour as well as DoS attacks. The biggest concern in this area is representation of the internet traffic. It is very difficult to describe and model real world situation due to very complex and evolving computer network structure and big variety of communication protocols.
4. The cost-base, game-based, rewrite theory and mathematical models are the most common while modelling DoS attacks. The first three model types usually relay on programming approaches for realisation while the mathematical models can be realised both with programming and

calculus solutions. This type of DoS attack models enable to get modelling results more quickly and therefore might be used in real time for DoS attack detection or countermeasure parameters setting.

5. The deterministic solutions are not sufficient for representation of real computer network traffic and a certain level of traffic randomness should be included in DoS and DDoS attack modelling and simulations.

New Taxonomies of DoS Attacks and Their Countermeasures

The DoS attack and their countermeasure terminology is important in real life for unambiguous communication. Detail DoS attacks and DoS attack countermeasures taxonomies can be used as terminology descriptors as well as learning material to understand all possible types of DoS attack and their countermeasures. In this Chapter new DoS attacks and DoS attack countermeasures taxonomies are proposed.

Analysis presented in this Chapter was published in (Ramanauskaitė and Čenys 2011b)

2.1. Proposed DoS Attacks Taxonomy

Different DoS and DDoS attack taxonomies exist and vary in presented DoS attack properties, detalization level and area of usage. To combine important categories of different taxonomies new DoS attack taxonomy was proposed (see Fig. 2.1). There are five dimensions for exploited vulnerability, number of attacking machines, type of victim and incoming traffic rate and attack source properties representation in it.

Fig. 2.1. New proposed DoS attack taxonomy

Considering exploited vulnerabilities S. Specht (Specht and Lee 2004) separate bandwidth and resource depletion categories while F. Kargl (Karig and Lee 2001) mentions a bug exploitation category too. We take it into account and classify DoS attacks considering vulnerability exploitation into these categories:

- **Bug exploitation attack.** If attacker uses the bugs existing in the victim system for its denial from the service, such an attack falls into the bug exploitation attack category. This category can be split into subcategories:
 - *Attack from outside.* If exploitation of existing bug in victims system directly affects the desired service, this kind of attack is called attack from outside.
 - *Attack from inside.* If attacker exploits a bug in victim system to get some kind of system control and then uses that control to affect the desired service, this attack falls into attack from inside category (service is denied from the same system where it is located).

Resource depletion attack. When all queries require quit a lot of resources, there could be no enough resources in this system for all incoming queries and it stops working properly. Considering the analysed resource type, this type of attack may be divided into such subcategories:

- *Memory depletion attack*. If attacker manages to create situation where all possible memory is allocated for queries and there is no enough memory for the new ones, this service is denied by the memory depletion attack.
- *CPU (Central Processing Unit) work depletion attack*. If attacker manages to supply incoming data which require even more CPU work to analyze or process this query, CPU may give excessive attention to this one job and be unable to do other jobs. If such a situation leads to denial of service, it can be described as CPU work depletion attack.

Resource depletion attacks may be classified into subcategories by attack package modification necessity:

- *Brute-force resource depletion attack*. Attack where sufficiently huge amount of incoming query denies a service is called a brute-force resource depletion attack. Huge amount of package is enough to reach the goal.
- *Semantic resource depletion attack*. If service is denied because of modified incoming data influenced resource depletion, this kind of attack falls into semantic resource depletion attack category.
- ***Bandwidth exhaustion attack***. Bombarding target with huge amount of data, the situation can be created where there is no more opportunity to send other data to the victim because of bandwidth exhaustion. Such a situation is called bandwidth exhaustion DoS attack.

Other property, because of which DoS attacks can vary, is the attacker size. It is important how many packages attacker can generate for its target within a certain period of time. However, it is more important to know what you are fighting with: if it is one computer or many coordinating one attack.

Like A. Fadlallah (Fadlallah and Serhrouchi 2005), we also clearly distinguish differences between single source and multiple source DoS attack:

- ***Single source attack***. If attack is launched from the single attacking machine, it is called single source attack.
- ***Distributed Denial of Service (multiple sources) attack***. Before launching an attack, attacker sets up an attack network (Botnet). He controls the network and uses it for one big and coordinated attack. Such a coordinated attack from the multiple sources is called distributed denial of the service attack.

In many DDoS taxonomies, there is one group for attack automation description (Douligeris and Mitrokots 2004; Mirkovic and Reiher 2002). In order to form the agent army attacker, it is necessary to find vulnerable machines and to install malicious code necessary for control

of that agent. Depending on automation of this action, agent army formation can be divided into four categories:

- *Manual agent army formatting.* When attacker has to find vulnerable machines and install malicious code into them himself, this agent army formatting is called manual.
- *Semi-automatic agent army formatting.* Sometimes attacker uses different tools for finding vulnerable machines and installing malicious code into them or uses tools just for one of these stages and the other installation is done manually. Such an agent army formation strategy is called semi-automatic because in some situations requires human attention.
- *Automatic agent army formatting.* Automatic agent army formatting requires human interference just to launch the tool which will do all necessary tasks to form the agent army.
- *Takeover of existing agent army.* There are agent armies which can be borrowed, rented or even stolen (taken over management). Takeover of the existing agent army does not require finding vulnerable machines but concentrates on the finding of the existing agent army and gaining its control.

C. Douligieris (Douligieris and Mitrokots 2004) classifies DDoS attack tools into Agent-based and IRC-based. It represents the idea of communication between agents and organizer of the attack. On the basis of it, we divide categories considering how communication with agents can be implemented:

- *Direct command assignment.* After turning into an agent, some machines stop obeying to the certain ports because all instructions and information is transmitted to this agent through this port. It is called direct command assignment because commands are transmitted directly to this agent.
- *Indirect command assignment.* Indirect communication can be implemented using IRC (Internet Relay Chat). After becoming an agent, machine goes to the appropriate chat room and monitors the ongoing conversation. Commands are transmitted into that chat but not directly to agent so it is called indirect command assignment.

Attacker practically never has direct access to the communication with agents. He uses stepping stones, proxy servers and other ways to hide his existence. However, there are methods which look like masking techniques but are intended to increase DoS effect (by increasing the amount of traffic or by circumvention of security measures).

Different authors classify these methods as data flood depletion attack (Specht and Lee 2004), type of structured attack (Fadlallah and Serhrouchi

2005), subcategory of vulnerability exploitation attack (Specht and Lee 2004), data flood attack (Karig and Lee 2001). We make these methods separate as the new property of DoS attack, i.e. methods for achieving DoS effect:

- **Direct causing of DoS effect.** If attack traffic from agent is transmitted directly to attack target, this method is called direct.
- **Reflector usage.** Sometimes attack traffic is transmitted not to the direct target but to the intermediate machine. This machine reflects the received traffic not to the original sender but to the real target. Usually it is achieved by spoofing attack packages and is called reflector usage.
- **Amplifier usage.** If attack packages indicate IP address of the target as the broadcast address, these packages will be retransmitted to all machines on the given subnet. Such an opportunity lets the attacker to amplify its attack and facilitates the achievement of DoS effect. This method is called amplifier usage and is used to attack the separate machines and even networks.

D. Karig (Karig and Lee 2001) distinguishes five types of victims in DoS attacks. We separate three categories for clarification of the real objects and in order not to overwrite already existing categories:

- **Node.** Target type is node if attack is targeted just on one internet node and directly does not affect work of the other node.
Impact on the node can be different. Respectively, we distinguish two types of the target nodes:
 - *Denial of node.* If attacker tries to deny service by denying a node (this node do not respond at all or shuts down), this kind of target is treated as all node.
 - *Denial of service in a node.* If attacker denies one or few services in node, but this node can continue other services the attack target is service in a node.
- **Network.** If attacker wants to deny service for certain network or make the network act up, the attack target is the network.
- **Infrastructure.** There are distributed services, which can be denied by denying just one or few nodes in its structure. Internet and DNS importance in this infrastructure is one of the examples. If attacker is targeted into denying of something what can deny the distributed system, this attack target is not just a node but infrastructure.

Attack dynamic is also important to describing of DoS attacks because of possibility to detect an ongoing attack on the early stages. There can be two main categories for attack rate description:

- **Constant rate.** Attacks where agent machines generate attack packets at a steady rate (usually as many as their resources permit) are called constant rate attacks.

- **Variable rate.** Variable rate attacks vary the attack rate of an agent machine in order to delay or avoid detection and response. Usually authors describe increasing and fluctuating attack rate dynamics (Douligeris and Mitrokots 2004; Mirkovic and Reiher 2002). We specify three possibilities:
 - *Uniformly increasing.* If the attack rate little by little increases, by certain rate every time, this attack is called uniformly increasing. In such a way, victim resources can be slowly exhausted and even without being notified concerning attack.
 - *Pulsar.* Attack is called pulsar if its rate time to time increases and then decreases. Changes are because of the victim behaviour or pre-programmed timing, occasionally relieving the effect to avoid detection.
 - *Random.* To avoid attack detection, attacker tries to generate as much random packages as possible. Such attack is called random.

All distinguished DoS attack properties and categories form certain DoS attack classification (see Fig. 2.1). Information about attack has to be indicated in all five categories in order to fully describe the DoS attack.

2.2. Proposed DoS Attack Countermeasures Taxonomy

To combine different ideas of DoS attack countermeasure taxonomy new DoS attack countermeasure taxonomy was proposed (see Fig. 2.2). It has three criteria to classify a DoS attack countermeasure: countermeasure phase, countermeasure deployment location and countermeasure cooperation with other systems level.

The main criteria for DoS countermeasure classification is the phase of actions in attack process. According to the criteria, we separate such categories:

- **Prevention.** Prevention tries to shut out DoS attack before it happened or adequately prepare for future attacks. Like J. Mirkovich (Mirkovic and Reiher 2002), we divide prevention phase into:
 - *DoS prevention.* DoS prevention includes all measures necessary to avoid DoS effect but not attack and can be divided into resource management, elimination of vulnerabilities and deflection. We suggest to describe it in more detailed way:
 - *Resource multiplication.* Resource multiplication means adding additional resources to be sufficient to scope incoming attack.

- *Resource accounting*. If resources are restricted for not privileged users, the prevention mechanism is called resource accounting.
- *Patches and upgrades*. All systems must be up to date, and the necessary update process is called prevention (regular patches and upgrades).
- *Build-in defences*. There are vulnerabilities which cannot be eliminated by means of patches and upgrades and requires the additional help. Build-in defences are another way to prevent system vulnerabilities exploitations.
- *Deflection*. Honeypots are systems with limited security and are used for traffic and analysis of its characteristics by deflecting the potential attack traffic.

Fig. 2.2. New proposed DoS attack countermeasure taxonomy

- *Attack prevention*. Other than DoS prevention, attack prevention tries to prevent the possibility of DoS attack, no matter what kind. This kind of prevention measures concentrate on the global networks maintenance, and not only the machine. Similarly to M. Specht (Specht and Lee 2004), we classify four possible ways of attack prevention:

- *Detect and neutralize handlers.* If we would detect all the existing handlers, there would be no possibility to execute DDoS attack managed by the handler-agent architecture.
 - *Detect/Prevent secondary victim.* Detecting of all the vulnerable machines and preventing them from being taken over would stop the DDoS attack possibility.
 - *Detect/Prevent potential attacks.* Monitoring the internet traffic, it is possible to observe attack signs. It can be done using egress filtering and MIB statistics.
 - *Dynamic pricing.* If network service providers would take taxes for the network usage, all machines would be more interested in prevention, i.e. that they would not be improperly used. This way, sometimes the price of desired flooding attack can become bigger in comparison to the benefit of the attacker.
- **Detection.** If prevention was not sufficient, it is very important to detect incoming attack as soon as possible. For attack detection, there can be mentioned couple of the information sources. Considering the observed attack detection information, detection measures can be divided into three categories:
- *Package header observation.* Packet header observation includes IP address and other package attributes supervision.
 - *Package content observation.* If package content is observed to identify DoS attack, this kind of detection mechanism is called package content observation.
 - *Traffic rate observation.* According to the incoming traffic rate, it is possible to indicate the incoming DoS attack. In such a case, the detection mechanism is called traffic rate observation.
- The other criteria used to classify detection phase are related to strategies used to detect the attack. We consider J. Mirkovic's (Mirkovic and Reiher 2002) reactive level DDoS defence mechanisms classification and separate three groups:
- *Pattern detection.* The strategy based on the comparison of all communications with the existing records in the attack signatures database is called pattern detection.
 - *Anomaly detection.* Anomaly detection strategy tries to compare usual traffic with the incoming, and alerts if the traffic is too much different then usually.

- *Third-party detection*. Mechanisms that deploy third-party detection and do not handle the detection process themselves but rely on external systems.
- **Mitigation/Stop**. Mitigation phase tries to lighten or even stop the attack. M. Specht (Specht and Lee 2004) classifies this phase into three categories, and we support this classification:
 - *Load balancing*. Load balance means bandwidth increase on critical connections or server replication. Usually it is done by the network provider.
 - *Flow control*. The max-min fair server-centric router throttle method sets up routers that access the server with logic to adjust (throttle) incoming traffic to the level that will be safe for the server to process.
 - *Drop request*. To mitigate attack some requests can be randomly or specifically dropped.
- **Post-action forensics**. All attacks must be suitably analyzed to prevent the similar situations in the future. C. Douligieris (Douligieris and Mitrokots 2004) calls it the intrusion response and describes such an internal classification, so do we:
 - *Traffic pattern analysis*. During DoS attack, traffic pattern data can be stored and then analyzed. It allows updating the information for later attack detection or even mitigation.
 - *Analysis of event logs*. Analysis of event logs, selected from the attack period, allows more specifically determine the type of attack and for the second time to prepare for such attacks.
 - *IP trace back*. In order to determine and possibly eliminate the attacker, it is important to trace back the attack traffic. This can be done in many ways.

The other criterion for DoS countermeasure classification is countermeasure location. C. Douligieris, A. Fadlallah and J. Mirkovic (Douligieris and Mitrokots 2004; Fadlallah and Serhrouchmi 2005; Specht and Lee 2004;) separate three possible places. We suggest four possible locations:

- **Victim machine**. Some DoS countermeasures are designed to use in the single machine, to defend it from attack or at least to mitigate it.
- **Victim network**. If countermeasure should save all computers in the network from DoS effect or even self attack, it has to be deployed in the victim network.
- **Source network**. The DoS countermeasure deployed in the same network as the attacker and meant to block all the outgoing attack traffic is called defence in the source network.

- **Intermediate network.** Intermediate network countermeasures should ensure defence to a large number of hosts. It should be deployed somewhere in the Internet and monitor all the suspicious traffics.

One more topic criterion for DoS countermeasure classification is the cooperation degree, mentioned in A. Fadlallag and J. Mirkovic's (Fadlallah and Serhrouchmi 2005; Mirkovic and Reiher 2002) works. This allows estimation of the attacker type, and we distinguish three possible cases:

- **Autonomous.** Autonomous mechanisms perform independently at the point where they are deployed (a host or a network).
- **Cooperative.** Cooperative mechanisms usually co-operate with the other systems, but also can work autonomously.
- **Interdependent.** Interdependent mechanisms cannot operate autonomously at the single deployment point. They require deployment at multiple points.

All distinguished DoS attack countermeasure properties and categories form certain DoS attack countermeasure classification (see Fig. 2.2). Information about attack has to be indicated in all three categories attempting to fully describe DoS attack countermeasure. Because of distributed and composed protection, there is possibility to distinguish one countermeasure even considering couple of categories.

2.3. Conclusions for Chapter 2

1. New DoS attack taxonomy was proposed, which includes new not used in existing taxonomies properties: CPU depletion as a new resource exhaustion attack; new method for attack agent formation. These categories allow for more precise DoS attack classification.
2. New DoS attack taxonomy suggests different taxonomy structure, to make it more intuitive. The victim node is divided into two categories, while new dimension is proposed to represent the method used to cause the DoS effect. All main properties of DoS attack countermeasure taxonomies were combined into new DoS attack countermeasure taxonomy. Two types of prevention were purposed: attack prevention and DoS prevention. It represents a wider DoS attack countermeasure area while keeping the taxonomy well structured.

Composite DDoS Attack Model

A reasonable number of DoS and DDoS attack models exists. Usually, however, they are concentrated on one specific attack type or even specific attack or situation representation. For practical purposes more general models can be required which would take into account as much as possible different attack threats as well as protection ways.

In this chapter, we present DDoS attack models for three main attack types (bandwidth depletion, memory exhaustion and CPU work exhaustion) as well as composite DDoS attack model, which would represent a probability to survive modelled DDoS attack no matter which resource was exhausted.

The analysis and research done in this chapter was published in (Ramanauskaitė 2010; Ramanauskaitė and Juknius 2010, Ramanauskaitė and Čenys 2011a, 2011c, 2012).

3.1. Bandwidth Depletion DDoS Attack Model

Analysing bandwidth depletion DDoS attack, incoming traffic which reaches the victim system is treated as composed of two components: normal traffic and attack traffic.

As mentioned in Chapter 1, many researches showed that on the package level and time scales smaller than 1s, wide area traffic behaves as multifractal (Riedi and Vehel 1997; Vedula and Riedi 1997). However, it applies exceptionally to normal traffic. Attack traffic usually is not so dependent on the traffic burstness and package autocorrelation because of its nature – it tries to exhaust victims resources with first sent packages and do not continue communicating with the victim as normal user. Such behaviour is very common to bandwidth depletion as well as memory exhaustion DDoS attacks. Taking into account fact that during DDoS attack the attack traffic can be bigger in size than the normal one, the overall victim incoming traffic is multifractal when attack traffic is close to zero, but when attack traffic is bigger than the normal one, the overall traffic can be represented as Poisson process even in package level.

In the conceptual model of bandwidth depletion DDoS attack we suggest do not analyse the whole network structure from the attack organiser, attack agents, global routers and victim because of potentially too big amount of simulated data. For bandwidth depletion DDoS attack modelling we take into account the behaviour of attack victim and the closest router to it. However, the router properties are not detailed and we assume, that it is capable to route all incoming packages without influencing the incoming traffic pattern. Meanwhile the link between the router and victim has limited throughput since it is capable to transmit packages only with the limited speed T .

Due to limitations of the link to victim, the router in the proposed model can act in three ways: to send the package to other node than victim if it was dedicated to it; to send the package to victim if it is dedicated to it and the link is capable to serve additional data; to drop the package because of too busy line to package receiver (see Fig. 3.1).

Fig. 3.1. Illustration of bandwidth depletion DDoS attack formal model

The drop of packets is result of router's inability to store waiting requests (the victim is responsible for it, and it will be discussed in Chapter 3.2 as a feature of the memory exhaustion DDoS attack) and insufficient link from victim to router capabilities to serve all incoming traffic. To increase the resistance to bandwidth depletion DDoS attack, the link throughput should be increased. Therefore additional channels can be opened to multiply the bandwidth resources.

The other way to decrease the bandwidth depletion DDoS attack success is usage of filters in the router (or other routers in the network). It would block a certain part of attack's traffic, however, as a result part of legitimate traffic can be blocked too. Therefore the two main router filtering system properties are the probability of false negative P_{fnr} (what is the percentage of not filtered out attack packages) and probability of false positive P_{fpr} (what is the percentage of incorrectly filtered out legitimate user packets).

For bandwidth depletion DDoS attack success probability estimation we suggest using queuing theory and M/G/K/K system with Poisson distribution for query arrival speed λ ; generally distributed service rate with parameter μ ; K channels when K channels can be used at time moment; there is no waiting line (if no free channels exists – the query is blocked and lost).

For model simplification we analyse situation on the packet level instead of byte or bit level. Therefore for normal traffic characteristics we take average packet arrival rate λ_{np} (packets/s) and average packet size v_n (bits). Attack traffic can have different characteristics, therefore average attack packet arrival rate λ_{bp} (packets/s) for one attack bot, its average packet size v_b (bit) and number of agents n in the attack should be obtained.

If all attack bots have very similar characteristics, the overall attack average packet arrival rate is equal to sum of all bot packet arrival rates (see formula 3.1), while the average packet size stays the same ($v_a=v_b$):

$$\lambda_{ap} = \lambda_{bp} \cdot n \quad (3.1)$$

However, if all attack bots act quite dissimilar the average arrival rate should be found a sum of all bot package arrival rates (see formula 3.2), while the average packet size should be taken as average of all attack traffic packets (see formula 3.3):

$$\lambda_{ap} = \sum_{i=1}^n \lambda_{bp_i} \quad (3.2)$$

$$v_a = \frac{\sum_{i=1}^n \lambda_{bp_i} \cdot v_{bi}}{\sum_{i=1}^n \lambda_{bp_i}} \quad (3.3)$$

Taking into account the router filtering parameters P_{fmr} and P_{fpr} , the attack and legitimate traffic decrease before facing the system waiting line:

$$\lambda_{ap}' = \lambda_{ap} \cdot P_{fmr} \quad (3.4)$$

$$\lambda_{np}' = \lambda_{np} \cdot (1 - P_{fpr}) \quad (3.5)$$

However, the victim system does not differentiate the legitimate and attack traffic, therefore we use general average packet arrival rate λ_{gp} , which is equal to sum of normal and attack traffic left after the filtering (see formula 3.6) and according general average packet size v_g :

$$\lambda_{gp} = \lambda_{np}' + \lambda_{ap}' \quad (3.6)$$

$$v_g = \frac{\lambda_{np}' \cdot v_n + \lambda_{ap}' \cdot v_a}{\lambda_{gp}} \quad (3.7)$$

The victim connection has T bits/s throughput, while the proposed model operates in packets rather than bits. Therefore the average channel service rate μ_{cp} is equal to number of v_g size packets, sent per second (see formula 3.8), while the overall average victim bandwidth service time μ_{gp} is equal to sum of all active channels service time μ_c :

$$\mu_{cp} = \frac{T}{v_g} \quad (3.8)$$

$$\mu_{gp} = \mu_{cp} \cdot K \quad (3.9)$$

To estimate the bandwidth depletion DDoS attack probability P_B , the legitimate user packet loss probability should be obtained. In this model, it is composed of two parts: legitimate user packets lost by router filtering system false positive situation probability P_{fmr} and legitimate user packet lost by unavailable bandwidth resource probability P_{ub} .

While the first one to us is known as a filter parameter (obtained from statistical or experimental data), the other one can be calculated using queuing system query lost formula (see formula 3.10) presented by Sanjay Kumar Bose (Bose 2010). In the proposed model, this formula shows the packet loss probability. We assume that both attack and normal flow packets are distributed uniformly in the router therefore it is equal to legitimate user packet loss probability P_{ub} .

$$P_{ub} = \frac{\frac{\rho^K}{K!}}{\sum_{j=0}^K \frac{\rho^j}{j!}}, \text{ where } \rho = \frac{\lambda_{gp}}{\mu_{gp}} \quad (3.10)$$

The router filtering system false positive filtering probability P_{fpr} applies to all of normal traffic packets, while the packets lost by unavailable bandwidth resource probability applies just to these packets which were not filtered by filtering system:

$$P_B = P_{fpr} + P_{ub} \cdot (1 - P_{fpr}) \quad (3.11)$$

We suggest analysing the router closer to the victim. The weakest link in the network should be chosen to represent more realistic situation. However, finding the weakest link can be quite challenging, because it requires analysing all possible ways to victim (while the source of attack is unknown) and if there are couple of them, the composite throughput should be obtained and used.

3.2. Memory Exhaustion DDoS Attack Model

In the memory exhaustion DDoS attack, victim has to serve requests from both legitimate users and attack bots as in bandwidth depletion DDoS attack. Therefore in this model the same Poison process can be used to represent the incoming traffic. However, packet level traffic analysis is suitable in this case.

Usually certain data are stored in the memory buffer just for appropriate stage of session, such as connection establishment in TCP. Therefore session level traffic analysis is more suitable to represent memory exhaustion DDoS attack. Taking this into account, the average size of packet or session is not important. Average bot session arrival rate λ_{bs} and average legitimate user session arrival rate λ_{ns} is enough to describe the general incoming traffic λ_{gs} .

As in bandwidth depletion DDoS attack model the general arrival rate λ_{gs} depends on system filtering false positive P_{fps} and false negative P_{fns} parameters as well as bot and legitimate user average session arrival rates λ_{as} , λ_{ns} and number of bots in the attack:

$$\lambda_{as} = \sum_{i=1}^n \lambda_{bsi} \quad (3.12)$$

$$\lambda_{as}' = \lambda_{as} \cdot P_{fns} \quad (3.13)$$

$$\lambda_{ns}' = \lambda_{ns} \cdot (1 - P_{fps}) \quad (3.14)$$

$$\lambda_{gs} = \lambda_{ns}' + \lambda_{as}' \quad (3.15)$$

In memory exhaustion DDoS attack, victim stores incoming requests until it will be finally served. However, the memory buffer is finite and is capable to store up to N sessions at once. All stored sessions are served not in FIFO (First In – First Out) queue, but are served during stay in the queue. Therefore we suggest using M/M/N/N queuing.

In this model, there are N services which represent how many session data can be stored in the memory buffer. There is no waiting line and new session data can be stored just if there is at least one free space in the memory buffer. Otherwise if all memory is occupied, new requests are dropped without serving them no matter if it is request from the legitimate user or bot (see Fig. 3.2).

Fig. 3.2. Example model of memory exhaustion DDoS attack

The memory exhaustion DDoS attacks can succeed because of possibility to occupy the memory buffer with mostly attack session data. It can be achieved by using sufficiently big number of incoming sessions by attack bots. Other way is to use not so many session requests, but affect the system to store attack sessions for a longer time in the buffer (as in TCP SYN attack). Therefore the average service time for normal session t_{ns} and attack session t_{as} are important parameters of this model.

For model simplification and difficulties to differ the attack and legitimate sessions in the real world, we use general average service time t_{gs} (see formula 3.16), which describes the average time of session data storage in the memory buffer (no matter if it is attack or legitimate user session data).

$$t_{gs} = \frac{\lambda_{ns}'}{\lambda_{gs}} \cdot t_{ns} + \frac{\lambda_{as}'}{\lambda_{gs}} \cdot t_{as} = \frac{\lambda_{ns}' \cdot t_{ns} + \lambda_{as}' \cdot t_{as}}{\lambda_{gs}} \quad (3.16)$$

The general average session service time t_{gs} , general average session arrival rate λ_{gs} and the size of memory buffer N can be used to calculate the request drop probability P_{im} caused by insufficient memory size:

$$P_{im} = \frac{\frac{\rho^N}{N!}}{\sum_{j=0}^N \frac{\rho^j}{j!}}, \text{ where } \rho = \lambda_{gs} \cdot t_{gs} \quad (3.17)$$

The memory exhaustion DDoS attack success probability P_M represents the proportion of legitimate user sessions which was not served during filtering or insufficient memory in the buffer to store it:

$$P_M = P_{fps} + P_{im} \cdot (1 - P_{fps}) \quad (3.18)$$

We do not analyse the changes of system filtering characteristics, depending on incoming traffic. These probabilities are used as constant, however, can be changed into desired functions.

3.3. CPU Work Exhaustion DDoS Attack Model

All incoming requests and system internal services have to be executed by its central processing unit (CPU). However, CPU resources are limited and it is capable executing commands in limited speed one at a time.

Multiplication of CPU frequency f and number of necessary cycles to execute the job c show how long it takes to finish one job. However, the job can wait until it will be executed, because one job is already under execution or even more jobs wait in the line before it.

Operating system scheduling algorithms manage the query of waiting jobs. In FCFS (First Come First Serve) scheduling algorithm new jobs will be placed at the end of the queue and will wait until all the previous jobs will be executed (see Fig. 3.3).

Fig. 3.3. Principle of FCFS scheduling algorithm

Other famous Round-Robin scheduling algorithm places a new query also at the end of the queue. However, all jobs are executed in quantum's and after

executing one quantum the job goes back to the end of the queue if it has not been finished yet or out of the system if its execution is finished (see Fig. 3.4). This scheduling algorithm has its own scope, however, the performance compared to FCFS is quite similar when time quantum is large and can be slower when time quantum is too small.

Fig. 3.4. Principle of Round-Robin scheduling algorithm

Both of these scheduling algorithms do not use priorities for jobs. Modern operating systems use more complex scheduling algorithms, taking into account job priority, job size, etc. However, CPU work exhaustion DDoS attacks can influence these parameters, therefore for CPU work exhaustion DDoS attack model we do not pay attention to the specific scheduling algorithm and assume that FCFS algorithm is used.

This assumption allows us to use M/M/1 queuing system for CPU work exhaustion DDoS attack model. In the number there is one channel and single infinite waiting line representing the ability to store as many job requests as it is needed (there is a limitation in the system memory, however, in practice it is hard to achieve).

All incoming traffic is processed by the CPU, however, different packets can require different amount of CPU work. Therefore in this model we use legitimate user and bot average packet arrival rates λ_{nc} , λ_{bc} as well as average number of cycles for legitimate user and bot sent packets processing c_n , c_b .

Knowing the number of bots, participating in the DDoS attack n , the average attack packet arrival rate λ_{ac} can be calculated:

$$\lambda_{ac} = \sum_{i=1}^n \lambda_{bci} \quad (3.19)$$

However, to know the general packet arrival rate λ_{gc} and average cycle number for general packet processing c_g requires additional system characteristics such as average service job arrival rate λ_s in the system and average cycle number needed for its processing c_s to make the model more realistic. These characteristics allow to take into account both packet processing and system running processes. However if it is not necessary, these parameters can be skipped in 3.20 and 3.21 formulas to calculate the general job arrival rate λ_{gc} and average cycle number for general job processing c_g .

$$\lambda_{gc} = \lambda_{nc} + \lambda_{ac} + \lambda_s \quad (3.20)$$

$$c_g = \frac{\lambda_{nc} \cdot c_n + \lambda_{ac} \cdot c_a + \lambda_s \cdot c_s}{\lambda_{gc}} \quad (3.21)$$

The general average job arrival rate represents the rate of incoming job into the CPU. While the average CPU service rate μ_{cc} can be found using average cycle number for general job processing c_g and CPU frequency f :

$$\mu_{cc} = c_g \cdot f \quad (3.22)$$

The victim can notice the CPU work exhaustion DDoS attack by increase of CPU utilisation. However, for service users only the service quality matters. Therefore we suggest judging the CPU work exhaustion DDoS attack probability by average time a job spends in the system L and its comparison to time limit t_w which the user is ready to wait for a job to be served.

The calculation of average time a job spends in the system can be made onlyt in the steady state (when the arrival rate is less than the service rate) (Anderson *et al.* 2002), otherwise the queue length would increase to infinity, as well as service time:

$$L = \frac{1}{\mu_{cc} \cdot S - \lambda_{gc}}, \text{ when } \frac{\lambda_{gc}}{\mu_{cc} \cdot S} < 1 \quad (3.23)$$

There S represents a number of CPU's in the system. All of them have its own waiting line, which is approximately of the same length, and therefore $1/S$ of the average general job arrival rate have to be served by one CPU.

For not steady M/M/1 system cases the CPU work exhaustion DDoS attack success probability P_C is equal to 1. Otherwise the attack success probability depends on the average time a job spends in the system: if it is less then t_w – success probability is equal to L/t_w ; if it is greater or equal to t_w – success probability is equal to 1:

$$P_C = \begin{cases} 1, & \frac{\lambda_{gc}}{\mu_{cc} \cdot S} \geq 1 \quad or \quad L \geq t_w \\ \frac{L}{t_w}, & \frac{\lambda_{gc}}{\mu_{cc} \cdot S} < 1 \quad and \quad L < t_w \end{cases} \quad (3.24)$$

This model does not take into account the time, needed to schedule tasks to multiple processors when more than one is used.

3.4. Composite DDoS Attack Model

All three designed models are proposed for a specific type of DDoS attack – bandwidth depletion, memory exhaustions or CPU work exhaustion. Usually during one type of DDoS attack all three resource types are used, but some of them in a smaller amount. Therefore for more real situation representation DDoS attack success probability should be estimated as a probability that at least one of different type of DDoS attacks will be successful:

$$P = 1 - \overline{P_B} \cdot \overline{P_M} \cdot \overline{P_C} \quad (3.25)$$

However, these models are not independent from each other (see Fig. 3.5) and require certain modifications to combine them together into the single composite model.

Fig. 3.5. Principle scheme of the composite DDoS attack

Incoming traffics for memory and CPU work exhaustion model should be correlated with bandwidth depletion model's output traffic. On the other hand the memory and CPU work exhaustions systems should work in parallel. In the real life, these two subsystems interact, however, for model simplification and because some of queries in CPU requires space in memory buffer and some do not, we separate these two subsystems as using similar incoming data, but not interacting with each other.

3.4.1. CPU Work Exhaustion Model Modifications to be Used with Bandwidth Depletion DDoS Attack Model

The CPU work exhaustion DDoS attack model uses the same incoming traffic which has been processed by bandwidth depletion DDoS attack model. However, because of used packet loss in bandwidth depletion model, the incoming flow should be accordingly decreased. While the bandwidth depletion DDoS attack probability P_B shows what part of incoming traffic was blocked out, we can use it to find the general job arrival rate λ_{gc} :

$$\lambda_{gc} = \lambda_{gp} \cdot (1 - P_B) + \lambda_s \quad (3.26)$$

Accordingly and average cycle number for general job processing c_g should change. Because we assume, that the attack and legitimate flow are distributed uniformly, therefore it is reduced by the same proportion:

$$c_g = \frac{(\lambda_{np} \cdot c_n + \lambda_{ap} \cdot c_a) \cdot (1 - P_B) + \lambda_s \cdot c_s}{\lambda_{gc}} \quad (3.27)$$

Usage of modified general job arrival rate λ_{gc} and average cycle number for general job processing c_g in formulas 3.22-3.24 should represent more realistic DDoS attack situation than using the same incoming traffic in CPU work exhaustion DDoS attack as in bandwidth depletion. Also combination of these two models requires less statistical data for model parameter values, than using it separately (λ_{nc} , λ_{bc} , n are not necessary anymore, because n is the same, while λ_{nc} and λ_{bc} can be computed from bandwidth depletion model).

3.4.2. Memory Exhaustion Modifications to be Used with Bandwidth Depletion DDoS Attack Model

Combining bandwidth depletion and memory exhaustion DDoS attack models can be complicated because of different type of incoming traffic level – bandwidth depletion model uses the packet level, while memory exhaustion model uses the session level.

To solve this problem, we use few new model parameters: average number of packets in one legitimate user session s_n ; number of packets in legitimate user session to set up the session r_n ; average number of packets in one bot session s_b ; number of packets in bot session to set up the session r_b . These parameters will show what part of legitimate user and bot packets should be used to store data about the session. This way we will be able to convert packet traffic into session level traffic as well as to represent average bot session arrival rate λ_{bs} and

average legitimate user session arrival rate λ_{ns} from data, used in bandwidth depletion model:

$$\lambda_{bs} = \frac{\lambda_{bp} \cdot (1 - P_B) \cdot r_b}{s_b} \quad (3.28)$$

$$\lambda_{ns} = \frac{\lambda_{np} \cdot (1 - P_B) \cdot r_n}{s_n} \quad (3.29)$$

3.28 and 3.29 formulas also represents the decrease of packet traffic comparing to its initial values because of losses in the bandwidth depletion model.

3.5. Conclusions for Chapter 3

1. Models for a certain type of DDoS attack are composed as a single queuing system and interact with the other systems just by system inputs or outputs, which makes the models more adaptable to different composite model situations.
2. The composite DDoS attack model success probability is equal to probability that at least one of the model subsystem success probabilities will be big enough, therefore this model allows different model subsystem combinations, leaving unnecessary subsystem success probability equal to zero. Accordingly the model can be extended by adding different subsystems if necessary.
3. The system resistance to DDoS attack is an inversion of DDoS attack success probability, therefore model results can be easily adapted for hosting ratings and other situations to define the system characteristics, not only DDoS attack success.

Experimental Results of Proposed Taxonomies and Models

In this chapter, comparison and evaluation of convenience of usage of the proposed DoS attack and their countermeasure taxonomies with the most common existing ones is presented. Different DoS attack situations are modelled using the proposed bandwidth depletion, memory exhaustion and CPU work exhaustion models to compare them with the alternative attack models. In addition, the composite DoS attack is modelled and analysed.

Research and its results described in this Chapter were published (Ramanauskaitė and Čenys 2011a, 2011c, 2012).

4.1. Comparison of Characteristics of DoS Attack and their Countermeasure Taxonomies

Criteria for DoS attack taxonomy analysis can vary depending on its usage area. For this work, some research was done to find out the usability characteristics for the proposed taxonomies and taxonomies analysed in the Chapter 1.

4.1.1. Analysis of DoS Attack Taxonomies

To describe DoS attack taxonomy quantitatively, we chose to analyse how many categories are represented in the taxonomy (overall and those which are not divided into smaller ones) and how many classification criteria (dimensions) are used in the taxonomy.

For this analysis, we used seven DoS taxonomies of the following authors:

- Asosheh (Asosheh and Ranezani 2008).
- Douligeris (Douligeris and Mitrokots 2004).
- Fadlallah (Fadlallah and Serhrouchmi 2005).
- Karig (Karig and Lee 2001).
- Mirkovic (Mirkovic and Reiher 2002).
- Specht (Specht and Lee 2004).
- The proposed taxonomy.

The analysis of DoS and DDoS taxonomies showed that all these taxonomies have different level of minuteness, which varies from 1 dimension, 5 undividable and 8 unique categories per taxonomy to 14 dimensions (different level), 52 undividable and 59 unique categories per taxonomy (Fig. 4.1). Meanwhile the proposed DoS attack taxonomy is close to the average considering these criteria among analysed taxonomies (7 dimensions, 25 undividable and 30 unique categories).

Fig. 4.1. Statistical data for the analysed DoS attack taxonomies

To estimate minuteness level, we consider the proportion of sum of choices in 1 a certain category and sum of categories, which can be divided into the smaller ones. This ratio should represent the average number of choices for the dividable category. The level of minuteness showed that DoS taxonomies proposed by J. Mirkovic (Mirkovic and Reiher 2002) have the biggest number of alternative values for the DoS attack – 8.83, the proposed model stands in the second place by this ratio – 5, while the other taxonomies has less than 4 choices per dividable category (Fig. 4.2).

Fig. 4.2. Category minuteness level for analysed DoS attack taxonomies

The statistical data show that taxonomy proposed by J. Mirkovic proposed taxonomy (Mirkovic and Reiher 2002) is the most detailed (is the widest by different attack characteristics and their possible value choices). Nevertheless, the proposed DoS attack taxonomy is the second with biggest number of choices and third by number of classification criteria.

4.1.2. Experiment Results of DoS Attack Taxonomy Usage

To represent taxonomy application characteristics, the practical experiments were done. Different DoS attacks were executed (SQL wildcard attack, Nuke attack and worm executed shutdown of specific service) in the local network and respondents had to classify them according to the analysed taxonomies. There were 34 respondents (Informatics engineering bachelor students) in the research, and all of them classified these three DoS attacks in all analysed taxonomies. They acted as victims and knew data from the victim machine observation (the attack machines were unknown to them).

One of these attacks could not be called a DDoS attack, as it happened because of one of the worm actions, while some of these taxonomies were meant for DDoS classification. However, this case allows us to judge if such an attack may be classified by DDoS taxonomies or no.

The research results showed that classifying SQL wildcard and worm attacks students marked the biggest number of categories in the proposed taxonomy. The biggest number of Nuke attack categories were marked in J. Mirkovic's (Mirkovic and Reiher 2002) taxonomy (Fig. 4.3 a). However, estimating the proportion of marked and existing categories in the analysed taxonomies, the biggest percentage of categories marked was in the proposed DoS attack taxonomy for all three attack cases (Fig. 4.3 b).

Fig. 4.3. Number (a) and percentage (b) of assigned categories in certain DoS attack taxonomy for different type of DoS attack

To distinguish specific properties of the taxonomy usage, the additional characteristics were calculated from this experiment: how many marked categories have smaller classification; how many dimensions were used to define an attack (Table 4.1).

Table 4.1. Classified attack statistics for different DoS attack type and taxonomy

Attack	Characteristic	DoS attack taxonomy author						
		Karig	Fadlallah	Spetch	Asosheh	Our	Douligieris	Mirkovic
SQL wildcard	Marked categories (%)	12.5	100.0	5.6	13.2	25.0	10.0	11.0
	Not expanded categories (%)	50.0	0.0	100.0	0.0	0.0	25.0	0.0
	Marked dimensions (%)	100.0	100.0	100.0	62.5	85.7	50.0	46.4
Nuke	Marked categories (%)	12.5	10.0	5.6	18.4	18.3	11.3	10.2
	Not expanded categories (%)	50.0	0.0	0.0	0.0	9.1	0.0	0.0
	Marked dimensions (%)	100.0	100.0	100.0	75.0	78.6	50.0	42.9
Worm	Marked categories (%)	12.5	10.0	0.0	5.3	15.0	3.8	5.9
	Not expanded categories (%)	50.0	0.0	0.0	0.0	0.0	0.0	0.0
	Marked dimensions (%)	100.0	100.0	0.0	25.0	64.3	18.8	25

Classifying DoS attacks, the percentage of marked dimensions decreases while the number of dimensions increases. This can be explained by disability to define all attack properties by victim if taxonomy include just attacker known properties. However, this could be not true if historical, well known attack data would be classified.

4.1.3. Analysis of DoS Attack Countermeasure Taxonomies

By analogy, as in DoS attack taxonomy analysis, DoS attack countermeasure taxonomies were analysed. The proposed DoS attack countermeasure and six DoS or DDoS attack taxonomies, analysed in Chapter 1, were chosen for the statistical analysis:

- Asosheh (Asosheh and Ranezani 2008).
- Douligeris (Douligeris and Mitrokotsa 2004).
- Fadlallah (Fadlallah and Serhrouhmi 2005).
- Karig (Karig and Lee 2001).
- Mirkovic (Mirkovic and Reiher 2002).
- Specht (Specht and Lee 2004).
- The proposed taxonomy.

The proposed DoS attack countermeasure taxonomy takes second place among other taxonomies in any of analysed criteria.

Fig. 4.4. Statistical data for analysed DoS attack countermeasure taxonomies

DoS attack countermeasure taxonomy presented by C. Douligeris (Douligeris and Mitrokotsa 2004) has the biggest number of unique and undividable categories (44 and 36) as well as the biggest value choice for one category (4.5). However, this taxonomy has only two dimensions.

Fig. 4.5. Category minuteness level for DoS attack countermeasure taxonomies

While DoS attack taxonomy category minuteness level varied from 1 to 9, the DoS attack countermeasure does not reach more than 4.5. More than half of the analysed taxonomies minuteness level also vary from 3.5 to 4.5.

4.1.4. Experiment Results of DoS Attack Countermeasure Taxonomy Usage

In the same experiment for DoS attack classification of three attacks, respondents were asked to suggest countermeasures that could be used against certain DoS attack. They had to mark all suitable countermeasures in all our analysed countermeasure taxonomies.

Fig. 4.6. Number (a) and percentage (b) of assigned categories in certain DoS attack countermeasure taxonomy for different type of DoS attack

Analysing the DoS attack countermeasure suggestions for a specific DoS attack, it appeared that the biggest number of proposed countermeasures were in the proposed taxonomy. Nevertheless, estimating the percentage between number of possible categories and suggested countermeasure categories, the biggest percentage of taxonomy categories was in the taxonomies proposed by A. Fadlallah (Fadlallah and Serhrouchi 2005) and J. Mirkovic (Mirkovic and Reiher 2002) (Fig. 4.6).

In the case of DoS countermeasure taxonomies for the worm type attack, the interrogated respondents had more suitable categories to suggest than for traditional DoS attacks.

Table 4.2. Suggested attack countermeasures statistics for different DoS attack type and taxonomy

Attack	Characteristic	DoS attack countermeasure taxonomy author						
		Fadlallah	Karig	Specht	Douligieris	Our	Asosheh	Mirkovic
SQL wildcard	Marked categories (%)	33.3	7.7	3.1	5.9	18.1	7.9	21.7
	Not expanded categories (%)	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Marked dimensions (%)	100.0	100.0	8.3	100.0	100.0	100.0	100.0
Nuke	Marked categories (%)	33.3	7.7	6.3	5.9	19.4	7.9	26.1
	Not expanded categories (%)	0.0	0.0	50.0	0.0	0.0	0.0	0.0
	Marked dimensions (%)	100.0	100.0	16.7	100.0	100.0	100.0	100.0
Worm	Marked categories (%)	38.9	11.5	3.1	5.9	20.8	7.9	26.1
	Not expanded categories (%)	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Marked dimensions (%)	100.0	100.0	8.3	100.0	100.0	100.0	100.0

Analysis of suggested DoS attack countermeasure structure in the taxonomy showed respondent desire to choose the most detailed solution (nodes, which cannot be expanded any more) and to mark all possible solutions as possible (in all dimensions). This tendency do not suite only for S. M. Specht (Specht and Lee 2004) taxonomy.

4.2. DoS Attack Model Validation Experiments

There are no models or methods to estimate the success probability of DoS attack. Therefore it is difficult to validate the proposed models: existing models produce different type of output while the real experiments relay on the subjective opinion of the expert.

For our model comparison to other models we choose characteristics, representing the utilisation of some kind of resource and threat it as attack success equality.

4.2.1. Experimental Situation and Validation Environment

For models validation OPNET IT Guru, the academic edition (Kapati) modelling tool was used. It is a useful engineering and research tool for streamlining the design and performance analysis of communication systems and protocols (Bermudez *et al.* 2003). This software supports different communication architectures, such as PSTN, ISDN, cable networks, xDSL technologies, fiber networks, and also wireless like Wi-Fi, GPRS, UMTS and has widely-supported construction parts' library with computer workstations and servers, routers, switches, bridges, stars, access points, links, firewalls, gateways, servers and the other comprising elements (Mohorko *et al.* 2008).

The situation map was created to represent the victim server situated in Spain, legitimate traffic sources in the North and South of Europe and attack traffic sources in Russia (Fig. 4.7). All situation nodes are connected with at least 100BaseT lines, while victim server is connected through 10BaseT connection.

Fig. 4.7. Modelled situation map in OPNET IT Guru tool

The victim server supports different kind of services: database, email, file transfer and web browsing, while system users can be spitted into five categories:

- Engineer – uses web browsing, email and file transfer.
- Researcher – uses web browsing and email.
- E-commerce customer – uses web browsing.
- Sales person – uses database access, email, web browsing.
- Bot agent – uses heavy database access, web browsing and file transfer.

There are three main model configurations to represent different possible situations. In all of them, the legitimate user sub networks (Legitimate1 and Legitimate2) contain three LANs. Each LAN contains 20 users with 5 engineers, 5 researchers, 5 e-commerce customers and 5 sales persons. One of three models has no attack sub network, while in other two there is attack sub network with five LANs in it. In one of cases, there are 10 nodes in each attack LAN (using heavy file transfer and web browsing services) while in the other one there are 25 nodes in each attack LAN (using heavy database, file transfer and web browsing services).

Those three model situations differ in used service parameters, which are represented in Table 4.3.

Table 4.3. Parameters of three experimental model situations

Parameter \ Situation name	No attack	Attack 1	Attack 2
Number of legitimate LANs	6	6	6
Number of legitimate database users in one LAN	5	5	5
Number of legitimate email users in one LAN	15	15	15
Number of legitimate file transfer users in one LAN	5	5	5
Number of legitimate web browsing users in one LAN	20	20	20
Number of attack LANs	0	5	5
Number of nodes in one attack LAN	0	10	50
Victim CPU frequency, GHz	10.33	10.33	10.33
Legitimate database service average packet size, bit	250.7	251.1	272.6
Legitimate database service average arrival rate, packet/s	0.010	0.006	0.003
Legitimate database service average cycle number for request	910.5	910.5	910.5

Parameter \ Situation name	No attack	Attack 1	Attack 2
Attack database service average packet size, bit	Not used	$1.634 \cdot 10^4$	$1.603 \cdot 10^4$
Attack database service average arrival rate, packet/s	Not used	0.002	0.006
Attack database service average cycle number for request	$9.655 \cdot 10^5$	$9.655 \cdot 10^5$	$9.655 \cdot 10^5$
Legitimate email service average packet size, bit	372.0	363.2	383.8
Legitimate email service average arrival rate, packet/s	0.002	0.002	0.001
Legitimate email service average cycle number for request	$2.012 \cdot 10^5$	$2.012 \cdot 10^5$	$2.012 \cdot 10^5$
Legitimate file transfer service average packet size, bit	827.8	715.3	902.4
Legitimate file transfer service average arrival rate, packet/s	0.0007	0.0005	0.0001
Legitimate file transfer service average cycle number for request	$2.203 \cdot 10^6$	$2.203 \cdot 10^6$	$2.203 \cdot 10^6$
Attack file transfer service average packet size, bit	Not used	$2.539 \cdot 10^4$	$3.270 \cdot 10^4$
Attack file transfer service average arrival rate, packet/s	Not used	0.010	0.003
Attack file transfer service average cycle number for request	$1.868 \cdot 10^7$	$1.868 \cdot 10^7$	$1.868 \cdot 10^7$
Legitimate web browsing service average packet size, bit	350.0	350.0	350.0
Legitimate web browsing service average arrival rate, packet/s	0.010	0.010	0.004
Legitimate web browsing service average cycle number for request	$1.701 \cdot 10^5$	$1.701 \cdot 10^5$	$1.701 \cdot 10^5$
Attack web browsing service average packet size, bit	Not used	350.0	350.0
Attack web browsing service average arrival rate, packet/s	Not used	0.229	0.085
Attack web browsing service average cycle number for request	$1.954 \cdot 10^4$	$1.954 \cdot 10^4$	$1.954 \cdot 10^4$

These parameters represent situations when the victim is not under DDoS attack (No attack) and when it is attacked by different size of DDoS attack (Attack 1 and Attack 2).

4.2.2. Bandwidth Depletion Model VS Bandwidth Utilization

The bandwidth utilisation can be the main criteria for bandwidth depletion attack success estimation. It determines the percentage of throughput that is used. Therefore we assume that under 0% bandwidth utilisation the attack probability is equal to 0% (there will be enough bandwidth for new user query), while 100% bandwidth utilisation show that new query will have no opportunity to be sent to the user (the attack success is 100%).

The analysed model use different packet arrival rates and sizes for different services, therefore the average normal packet arrival rate λ_{np} (packets/s) and average normal packet size v_n (bits) can be obtained by formulas 3.2 and 3.3 as well as average attack packet arrival rate λ_{ap} (packets/s) and average attack packet size v_a (bit).

In these experiments we do not use any filtering, therefore the attack and legitimate traffic will not decrease because of the lost packets and can be used in formulas 3.6 and 3.7 to find the general average packet arrival rate λ_{gp} and general average packet size v_g .

In “No attack” situation average normal packet arrival rate λ_{np} is equal to 1.724 packet/s, average normal packet size v_n is equal to 2727 bits, average attack packet arrival rate λ_{ap} is 0 packets/s, average attack packet size v_a is 0 bits, general average packet arrival rate λ_{gp} is equal to 1.7241 packet/s and general average packet size v_g is equal to 2727 bits. Using these values to get the packet loss probability (formula 3.10) we get the bandwidth depletion DDoS attack probability P_B equal to 0.047%.

Executing OPNET IT Guru model the expected link utilization value and sample mean is 0.070%, while maximum utilization value is 0.940, variance is 0.013 and standard deviation is 0.115.

Comparing the proposed model and experimental OPNET results (Fig. 4.8), our model produce smaller success probability that the expected link utilization percentage. The difference is -0.023 or -33% of OPNET expected link utilization.

Adding to examined situation attack agents with compared to legitimate traffic increased packet size (situation “Attack 1”), the average normal packet arrival rate λ_{np} is equal to 1.548 packet/s, average normal packet size v_n is equal to 2742 bits, average attack packet arrival rate λ_{ap} is 12.84 packets/s, average attack packet size v_a is 19526 bits, general average packet arrival rate λ_{gp} is equal to 14.39 packet/s and general average packet size v_g is equal to 17720 bits. The bandwidth depletion DDoS attack probability P_B equal to 2.487%, while executing OPNET IT Guru model the expected link utilization value and sample mean is 2.841%, while maximum utilization value is 6.3196, variance is 1.044 and standard deviation is 1,022.

Fig. 4.8. DDoS attack success probability and link utilization comparison in “No attack” situation

The proposed model success probability is 0.388 positions or 13% smaller than the observed link utilization in OPNET tool.

Fig. 4.9. DDoS attack success probability and link utilization comparison in “Attack 1” situation

Increasing the attack power to situation “Attack 2”, the average normal packet arrival rate λ_{np} is equal to 0.654 packet/s, average normal packet size v_n is equal to 2792 bits, average attack packet arrival rate λ_{ap} is 23.62 packets/s, average attack packet size v_a is 19832 bits, general average packet arrival rate λ_{gp} is equal to 24.27 packet/s and general average packet size v_g is equal to 19373 bits. The bandwidth depletion DDoS attack probability P_B equal to 4.491 % (Fig.

4.10), while executing OPNET IT Guru model the expected link utilization value and sample mean is 4.902 %, while maximum utilization value is 8.728, variance is 1.358 and standard deviation is 1.165.

The proposed model success probability is 0.411 positions or 8% smaller than observed link utilization in OPNET tool.

Fig. 4.10. DDoS attack success probability and link utilization comparison in “Attack 2” situation

There is close relation between link utilization and our proposed bandwidth depletion DDoS attack model success probability (Fig. 4.11).

Fig. 4.11. Bandwidth depletion success vs link utilization

However, our proposed bandwidth depletion attack success probability is less than link utilization, represented by OPNET. To make these values close to

each other, the average packet size in our model should add the additional bits to represent the intervals between data sent through the link.

4.2.3. CPU Work Exhaustion Model VS Processing Time

To estimate the CPU work DDoS attack success probability our proposed model uses critical time limit t_w , and therefore the attack success probability is not adequate to CPU utilization. To compare our proposed model result to that obtained with OPNET modelling tool, we use an average request service time. The average cycle number for request (Table 4.3) was obtained by analysing request load and task processing time for different services as well as the victim CPU parameters.

Average service job arrival rate λ_s in the system and average cycle number needed for its processing, c_s were assumed to be zero because of unavailability to change this property in OPNET tool. The general job arrival rate λ_{gc} was calculated by adding all arrival rates for all used services. The average cycle number for general job processing c_g was calculated accordingly to logic of formula 3.21, but taking into account all used services, not legitimate and attack flows.

The request processing time is greater than the request arrival rate, therefore the system is in steady state and average service time can be computed (formula 3.23). Calculated processing time in our model is larger than processing time, obtained by OPNET tool (Table 4.4), nevertheless, its value is not larger than the maximum value of processing time, obtained from OPNET tool (Fig. 4.12).

Table 4.4. Request processing time: comparison of the proposed model and OPNET tool

Property \ Situation	No attack	Attack 1	Attack 2
Our model suggested request service time, s	$1.635 \cdot 10^{-4}$	$6.968 \cdot 10^{-4}$	$6.926 \cdot 10^{-4}$
OPNET model average task processing time, s	$1.335 \cdot 10^{-4}$	$4.253 \cdot 10^{-4}$	$2.815 \cdot 10^{-4}$
OPNET model maximum task processing time, s	$1.011 \cdot 10^{-3}$	$1.333 \cdot 10^{-2}$	$4.088 \cdot 10^{-2}$
OPNET model standard deviation for task processing time	$1.404 \cdot 10^{-4}$	$2.163 \cdot 10^{-3}$	$4.466 \cdot 10^{-3}$
OPNET model variance for task processing time	$2.000 \cdot 10^{-8}$	$4.680 \cdot 10^{-6}$	$1.995 \cdot 10^{-5}$

Fig. 4.12. Request processing time comparison in our and OPNET model

The experimental OPNET model showed higher processing time in “Attack 1” than in “Attack 2” despite the bigger incoming request rate, while our model expected service time is very similar to that of both situations.

4.2.4. Memory Exhaustion Model VS Calculated Comparison Value

Our memory exhaustion DoS attack model take into account the buffer size for storage of the open connection. Usually the additional buffer is used, therefore we treat those two buffers as one. However, OPNET modelling tool does not represent the usage of the second buffer which makes the comparison of our model considering buffer usage percentage difficult.

For experiments with memory exhaustion DoS attack model, we analyse different situations by choosing the service to be able to store from 16 to 64 open connections, additional buffer size from 4 byte to 100 bytes and different as well as general average session arrival rate λ_{gs} . We do not use spoofed attack sources, therefore the general average service time t_{gs} is obtained from the experimental OPNET models (Table 4.5).

For validation of the memory exhaustion DoS attack model we use the calculated comparison value, which represents the average number of sessions served in one second ($\lambda_{gs} \cdot t_{gs} / (N+M)$).

Table 4.5. Examples of the memory exhaustion DoS attack experiment data

Property \ Situation	Situation 1	Situation 2	Situation 3	Situation 4
Active connection threshold N (sessions)	16	64	32	32
Receive buffer M (sessions)	64	16	4	100
General average session arrival rate λ_{gs} (sessions/s)	10.86	14.49	5.403	65.25
General average service time t_{gs} (s)	1.321	9.926	51.17	32.34
Calculated value	0.179	1.798	7.680	15.99
Request drop probability P_{im}	$6.5 \cdot 10^{-17}$	0.452	0.870	0.937

Fig. 4.13. Dependency of memory exhaustion probability and calculated value

The comparison of memory exhaustion probability and calculated value showed the logarithmical distribution of memory exhaustion probability. It represents the real world situation when the increase of session service time has not linear influence on the buffer utilisation and memory exhaustion probability.

4.3. Conclusions for Chapter 4

1. Proposed DoS attack classification is easier to use in practice from the victim perspective, while more detailed taxonomies require some information, known exceptionally for the attacker.

2. The proposed DoS attack countermeasure taxonomy stands in the second place by number of categories and criteria comparing to the other similar taxonomies. Using this taxonomy for different type of DoS attack countermeasure suggestion it takes the first place comparing to other taxonomies. This proves the sufficient taxonomy abstraction and minuteness levels.
3. Proposed bandwidth depletion DDoS attack model success probability is a little bit smaller ($\approx 0.2\%$) than bandwidth utilization, obtained by OPNET tool. While experiments that were made do not vary enough (because of OPNET tool limitations) to prove the model and bandwidth utilization similarity in any case, however, it is enough to see similarities in DDoS attacks of average size.
4. The CPU work exhaustion DDoS attack model represent the situation in steady state, while OPNET is discrete event modelling tool, and therefore query service time variations can be seen. As a result the model can not be used to represent changes of query service time.
5. The memory exhaustion DDoS attack model was designed as system with one memory buffer to store the open connection information. The second buffer can be used to queue request before entering the first buffer. The proposed model can be adapted to represent this case as well demonstrating extensibility of the model.

Experiments with Composite DDoS Attack Model

Distributed denial of service attack has more than one characteristic, which has an influence on the attack success probability. The influence can also be different depending on the situations. However, real experiments with DDoS attacks can be time and resource consuming.

In this chapter, we use our proposed model to represent different DDoS attack situations for illustrating of the main tendencies in this kind of cyber attacks.

Research and its results described in this Chapter were published in (Ramanauskaitė and Čenys 2011a, 2011c, 2012).

5.1. Experiments with Memory and CPU Work Depletion DDoS Attack Model

Memory and CPU work depletion models allow to analyse only some parts of the overall attack. The real situation involves different elements of the different models at the same time. As a result more precise results can be obtained using the combined models.

We eliminated bandwidth exhaustion part in 3.25 formulae to get the composite memory and CPU work depletion DDoS attack success probability. Treating these two models as the separate systems, the composite resource depletion DDoS attack probability P should be equal to probability that memory depletion P_M or CPU work depletion P_C DDoS attack will be successful:

$$P = 1 - (1 - P_M) \cdot (1 - P_{CPU}) \quad (5.1)$$

No requests are lost until reach the memory for storage. Because of insufficient place in the buffer some queries can be lost. Those who had enough space to be stored in the memory are processed by the CPU. The execution time of queries can vary depending on request properties and size of the queue in the memory therefore some request can be threaten as unneeded (lost), because user stopped waiting for it.

We threat memory and CPU work depletion DDoS attack as one system with two subsystems, which depend on each other very closely. Therefore we made an experiment where constant incoming traffic rate was used to see how it influences properties of memory subsystem. The overall (both legitimate and attack traffic) incoming traffic rate λ was chosen to be a little bit larger than the maximum memory buffer size M .

The model properties and experiment results showed that at the beginning the incoming traffic was not too big to serve most of the queries (although the incoming traffic rate λ is higher than buffer size M). This happens because of CPU properties and accordingly high query service time. However, the average query service time increases in time (Fig. 5.1), as well as the average queue size (Fig. 5.2).

Fig. 5.1. Dependency of number of queries in the buffer in time, when incoming traffic is bigger than the maximum buffer size

Fig. 5.2. Average request service time changes in time, when the incoming traffic is a little bit bigger than the buffer size

This happens due to the fact that at any time part of queries are served immediately (while one is served, another can come) and part of queries stay in the buffer to wait for their time to be served. Accordingly due to increasing queue size the service time also keeps increasing.

However, Fig. 5.1 and Fig. 5.2 show that when queue size becomes constant, the average queries service time for a while keeps increasing anyway. This could be influenced by the fact that system starts to drop some of the queries or legitimate user does not wait any more for serving of a certain query. So the CPU starts to serve unnecessary queries. However, after some time the steady state is reached and service time stays practically the same.

Because of increasing queue size and average query service time, the attack success probabilities increase as well (see Fig. 5.3).

Fig. 5.3. Changes of the composite resource depletion DDoS attack success probability and its components in time, when the attack incoming rate is constant

The composite attack success probability is higher than any of its component. Form Fig. 5.3 we can see that CPU work depletion DDoS attack

success probability correlates with the average service time. However, the memory depletion DDoS attack success probability correlation with the average queue size is not so obvious. This is influenced by fact, that it depends both on the queue size and service time.

The second experiment was done using different incoming attack traffic rate function. We examined how the properties of model change when the attack traffic rate increases linearly.

Analyzing the queue size, the tendencies are very similar as with the constant attack traffic rate, but it starts from the smaller value (because of the smaller attack traffic) and increases more quickly (Fig. 5.4). However, analyzing the average query service time (Fig. 5.5), the new tendency showed up: in the beginning when attack traffic changes in time are bigger by proportion, the average service time increases; however, it starts to decrease when the proportion changes become smaller.

Fig. 5.4. Comparison of the queue size and incoming traffic rate in time using the increasing incoming traffic rate

Fig. 5.5. Changes of the average service time in time, using the increasing incoming traffic rate

The volume of this peak at the beginning of attack depends on the difference of execution number in CPU for legitimate and attack queries. The peak inflicts the attack success probability as well (Fig. 5.6).

Fig. 5.6. Changes of the composite resource depletion DDoS attack success probability and its components in time, when the attack incoming rate is increasing

From these experiments, we noticed that using different attack properties and incoming attack traffic functions different attack characteristics can be obtained. We examined two most common attack traffic rate functions and noticed the main memory and CPU work depletion attack characteristics:

- Memory and CPU work depletion model are subsystems of the resource depletion DDoS model. However, these models are very closely connected, since both use the attack characteristics obtained from each other.
- Most of the composite resource depletion attack characteristics react to the changed incoming traffic with successive changes but not immediate. Such effect can cause appearance of different volume peaks analyzing queue size, service time or attack success probabilities change in time.

5.2. Experiments with Composite Bandwidth and Memory Exhaustion DDoS Attack Model

The most common DDoS attack situation is when attack traffic exhaust both bandwidth and memory resources. In this chapter while analyzing DDoS attack we take into account both bandwidth and memory depletion DDoS models as well as the filtering properties of the system. The concept of this composite model is explained in (Fig. 5.7):

Incoming traffic can be blocked due to insufficient bandwidth. The rest part of the traffic can be blocked by the filtering system. Finally, what is left after the filtering can be blocked by insufficient place in the buffer, devoted to store open connections.

Fig. 5.7. Conceptual model of the composite bandwidth and memory exhaustion DDoS attack

To represent the composite attack probability we modify the 3.25 formula to add legitimate traffic filtering probability P_{Fn} :

$$P = 1 - (1 - P_B) \cdot (1 - P_{Fn}) \cdot (1 - P_M) \quad (5.2)$$

Using the proposed composite DDoS attack model, different situations were examined. The purpose of these experiments was to distinguish the influence of the different attack properties on DDoS attack success.

For the analysis of these experiments, the standard situation parameters were chosen:

- Regular 20Mbps traffic (100 queries per second by 200 bits in each).
- 10Mbps attack traffic (50000 queries per second by 200 bits in each).
- 1 channel with 100Mbps bandwidth.
- Victim uses filters which filter 20% of attack and 2% of legitimate users queries.
- Legitimate query takes 200ms to execute.
- Attack query execution takes 2000ms.
- Buffer can hold information of 50 connections.

These attack and victim parameters lead to 8.7% bandwidth exhaustion, 2% of legitimate queries filtered out and memory depletion probability 39.3%. The composite DDoS attack success probability was 45.7%.

Changing the filtering properties, it becomes clear that the legitimate queries blocking probability is very important and linearly increases the composite DDoS attack success probability. Meanwhile, the attack traffic filtering causes the memory and the composite DDoS attack probability to change in nonlinear way (Fig. 5.8).

Fig. 5.8. Attack success dependency on the attack query filtering percentage

Similar tendencies apply to the service time influence on the attack success as well. Increasing the average service time (both legitimate and attack queries), the memory depletion probability increases but its influence on the composite attack success is not proportional (Fig. 5.9).

Fig. 5.9. Attack success probability depending on the attack query service time

The previous experiments had influence only on the memory depletion and composite attack probabilities. Meanwhile, changing incoming traffic properties have influence on the bandwidth exhaustion probability as well.

Increase of legitimate and attack traffics leads to the attack success increase but in our experimental environment the memory depletion increase is higher than the bandwidth exhaustion probability. Even if we would decrease the service time 10 times, the bandwidth exhaustion impact on the composite attack success probability would be bigger only until the attack rate is quite small (see Fig. 5.10). While using the bigger attack traffic, the memory depletion probability has the bigger influence and is more sensitive to the attack size (see Fig. 5.11).

Fig. 5.10. Attack success probability dependency on the query arrival rate with the average service time

Fig. 5.11. Attack success probability dependency on the query arrival rate with very short service time

These experiments show how different part and composite DDoS attack success can depend on the various attack and victim parameters. Using this model some facts concerning DDoS attack success probability were revealed:

- Bad filtering system configurations can make more damage than DDoS attack itself.
- Memory depletion attacks are more sensitive to the incoming traffic changes and can be the main reason for the high composite attack success probability.
- The changes in DDoS attack and victim property values has non linear influence on attack success probability not mentioning the composite DDoS attack success probability.

5.3. Experiments with Bandwidth Exhaustion DDoS Attack Survive Model

Three main categories of DDoS attack countermeasures can be defined: attack prevention (usually increase of resources), attack mitigation (usually filtering attack traffic) and attack postvention (usually attack source traceback) (Mirkovich *et al.* 2005). Struggling with DDoS attack, the efforts would be successful if at least one of these three components is strong enough. If we knew the attack prevention probability P_{pr} , attack mitigation probability P_{mt} and attack postvention probability P_{pv} , we would be able to find the composite DDoS attack survivor probability P_S :

$$P_S = 1 - (\overline{P_{pr}} \cdot \overline{P_{mt}} \cdot \overline{P_{pv}}) \quad (5.3)$$

5.3.1. Attack Postvention Probability

DDoS attack postvention combines all technologies and methods which help to trace back the attack source. The postvention success probability P_{pv} mostly depends on the probability P_{TA} to trace back attack organizer and can vary for different type of DDoS attacks or even Botnet management type. Also this probability varies in time, because it is easier to trace back the attacker when attack agents are active, and when Botnet agent vanishes (agent shuts down or is eliminated from the Botnet), the trace back probability decreases.

Therefore the probability of attack organizer's traceability in time for one Botnet agent p_{Ta} should be equal to the probability to trace back the attack organizer analysing active agent p_{ta} when time t is less than the average agent lifetime T_w , and the probability to trace back attack organizer analysing not otherwise existing agent p_m :

$$p_{Ta}(t) = \begin{cases} p_{ta} & , t < T_w \\ p_{tn} & , t \geq T_w \end{cases} \quad (5.4)$$

$p_{Ta}(t)$ represents the probability to trace back attack organizer analysing just one Botnet agent. Meanwhile, in DDoS attack there are number of agents. Some of them can be full time active, while others can become inactive in time. So the probability to trace back attack organizer P_{TA} should be equal to probability that at least one analysed agent will lead to the real attack organizer. Human researchers can analyse only a small part of attack agents in one time frame, but computer systems are capable to analyse huge data sources at once. So the number of active $n_{t < tw}(t)$ and already not existing $n_{t \geq tw}(t)$ analysed Botnet agents since the beginning from the start of DDoS attack can help finding the total probability to trace back attack organizer:

$$\begin{aligned} p_{TA}(t) &= 1 - \prod_{k=1}^n p_{Tak}(t_k) = \\ &= 1 - (1 - p_{ta})^{n_{t < tw}(t)} \cdot (1 - p_{tn})^{n_{t \geq tw}(t)} \end{aligned} \quad (5.5)$$

We assume that in the postvention phase, the attempt to trace back the attack organiser is done by analysing all attack agents at the same time.

The proportion of attack query volume in all incoming queries $\rho_n(t)$ is also important (see formulae 5.6). This coefficient helps to reflect the difficulties of finding out which queries are legal (leads to no or fake trace back results) and which are attack queries (can lead to some trace back results) and should be used to estimate the real probability to trace back attack organizer.

$$\rho_n(t) = \frac{\sum n_a(t)}{\sum n_a(t) + n_n \cdot t} \quad (5.6)$$

The time t_{tr} required to execute a successful trace back operation should be also taken into account:

$$P_{pv}(t) = \begin{cases} p_{TA}(t) \cdot \frac{t}{t_{tr}} \cdot \rho_n(t) & , t < t_{tr} \\ p_{TA}(t) \cdot \rho_n(t) & , t \geq t_{tr} \end{cases} \quad (5.7)$$

5.3.2. Attack Mitigation Probability

DDoS attack mitigation phase tries to reduce the DoS effect during the attack. Usually it uses different filters to filter the attack traffic. So the DDoS attack

mitigation probability depends on the proportion of legitimate traffic in all traffic devoted to the victim (5.8), where legitimate queries filtering P_{fns} and attack queries filtering P_{fps} percentages are important as well as legitimate T_n and attack T_a traffic at that moment.

$$P_{mt}(t) = \frac{T_n \cdot (1 - P_{fps})}{T_a \cdot P_{fns} + T_n} \quad (5.8)$$

The legitimate (see formulae 5.9) and attack traffic (see formulae 5.10) can be found as multiplication of average query size and number of incoming queries at the moment for query types.

$$T_n(t) = \lambda_{np} \cdot v_n \cdot n_n(t) \quad (5.9)$$

$$T_a(t) = \lambda_{ap} \cdot v_a \cdot n(t) \quad (5.10)$$

For the legitimate user queries incoming rate, we assume that it is constant and do not go into details how legitimate user activity can inflict the DDoS attack success. However, for more detail legitimate user more detail statistics of the traffic representation should be obtained (Noreika and Drąsutis 2007). Meanwhile, the changes of attack agent amount in attack at the specific time moment can be quite important. We tried to find out how different agent recruitment strategies and agent properties affect the effectiveness of DDoS attack surviving probability.

For describing the number of agent in DDoS attack, we take into account such properties as:

- the initial number of agent in attack n_{in} ;
- average life time for one agent t_{alt} ;
- attack agent allocation time t_{aar} ;
- additional agent allocation function (describes how many agents are added to the attack every time moment) $n_r(t)$;

All these properties describe what the number of attack agents $n(t)$ is in the specific time moment:

$$n(t) = \begin{cases} n_{in} & , t = 0 \\ n(t-1) + n_r(t) & , t < t_{arr} \text{ and } t \leq t_{alt} \\ n(t-1) + n_r(t) - n_r(t-t_{alt}) & , t < t_{arr} \text{ and } t > t_{alt} \\ n(t-1) - n_r(t-t_{alt}) & , t \geq t_{arr} \text{ and } t > t_{alt} \\ n(t-1) & , t \geq t_{arr} \text{ and } t \leq t_{alt} \end{cases} \quad (5.11)$$

5.3.3. DDoS Prevention Probability

Attack prevention can combine many activities but in this case we are only interested in how well the victim is prepared to restrain the resource exhaustion. Therefore we use queries lost probability calculation to realise what is the probability that all queries will be lost and use its inversion:

$$P_{pr}(t) = 1 - P_{mt}(t) = 1 - \frac{\frac{\rho(t)^i}{i!}}{\sum_{j=0}^i \frac{\rho(t)^j}{j!}} \quad (5.12)$$

In this formula, i means a number of open channels, where all channels have a T_g throughput (used for bandwidth multiplication) and $\rho(t)$ represents the bandwidth utilisation rate:

$$\rho(t) = \frac{T_a(t) \cdot P_{fns} + T_n(t) \cdot (1 - P_{fps})}{T_g \cdot i} \quad (5.13)$$

5.3.4. Research of Botnet Agent Allocation Strategies in DDoS Attack

The proposed model allows to analyse DDoS attack survive probability and its component probabilities using different victim and Botnet configurations. To analyze efficiency of Botnet agent allocation strategies, we modelled one example corresponding to the standard situation: a victim uses filtering system where averagely 2% from 1 MB/s legitimate traffic (40 queries of 25 KB size every second) and 30% attack traffic is filtered out; Botnet and DDoS attack is managed the way that knowing one attack agent allows to trace back the attack organizer with 0.3% probability and requires 1 hour to do it under the allocation that all attack agents are analysed at the same time. Victim uses one 5 MB/s throughput channel.

To attack the victim, four agent recruitment functions $n_r(t)$ are used:

- No agent allocation during attack – $n_r(t)=0$;
- Constant agent allocation speed – $n_r(t)=12$;
- Linearly increasing agent allocation speed – $n_r(t)=0.04 \cdot t$;
- Linearly decreasing agent allocation speed – $n_r(t)=20-0.0267 \cdot t$.

When no additional agents are involved in the attack during the time, the initial agent number is 7288, and in all the other cases it is 100 agents. One agent can send averagely 25 KB/s traffic to the victim and its lifetime is 5 minutes.

The agent allocation is executed for 10 minutes, so all used agent allocation strategies will use the same number of individual agents (7288).

The proposed model allows us to analyse how the number of agents changes in attack, using different agent allocation strategies. It also allows estimating how average agent lifetime can influence the DDoS attack power.

Analysing the four agent allocation strategies we have chosen (see Fig. 5.12), it is possible to see that even the same number of agents differently distributed in the time period, predetermines achieving of completely different attack rate variants:

- Constant agent allocation speed allows to organize long lasting attacks, no matter what is the agent lifetime, but it's feature is especially small momentum attack power.
- Linearly increasing or decreasing agent allocation speed, the attack power always changes (increases and after reaching its peak decreases) and cannot be stable for a longer time.

Fig. 5.12. Changes of number of the bots in DDoS attack when different agent recruitment strategies are used

The analysed changes of agent number in DDoS attack can be used to achieve the biggest DDoS attack power in different moments or intervals of attack. So we examined the efficiency of DDoS attack in different time intervals after the attack was launched (Table 5.1).

The DDoS attack efficiency, in different time intervals after attack was launched, shows that the biggest effect at the beginning can be achieved when

the instant agent number is large. However, using no agent allocation the DDoS effect would decrease very rapidly, comparing with other analysed strategies.

Table 5.1. Comparison of DDoS effectiveness in different time intervals and using different agent allocation strategies.

Agent allocation strategy	Attack success probability, %			
	In the middle of attack, 300 s	At the end of agent allocation, 600 s	At the end of attack flow generation, 900 s	After the end of attack, 1500 s
No agent allocation	91.2	45.7	30.6	18.5
Constant agent allocation	75.2	77.5	72.0	43.3
Linearly increasing agent allocation	54.7	67.0	67.0	40.3
Linearly decreasing agent allocation	80.2	79.9	70.2	42.2

Using constant agent recruitment number, it can lead not only to the most steady attack efficiency in different time intervals but – in the current situation – to the biggest attack effect in the long time period (even after the end of attack and if no further attacker activities are implemented). However, the linearly changing agent allocation speed predetermined gives quite similar results with the exception that the bigger difference of attack effectiveness, which is steady over the time, is noticed.

Analysis of the attack survive probability changes in time using different agent allocation strategies in the current situation reveals that increasing the attack traffic linearly, the attack survive probability decreases exponentially, as well as decreasing the attack traffic, it increases exponentially (Fig. 5.13, Fig. 5.14, Fig. 5.15, Fig. 5.16).

In addition, this research showed that survive probability mostly depends on the prevention probability. However, analysing the other situations and not only this one, it appears that the mitigation probability can be dominant as well.

Postvention probability in practice is very small and does not act as the main component of the survival DDoS attack. However, in the later stages of attack composite survival probability can increase.

Comparing the prevention, mitigation and postvention probability changes during the DDoS attack, similarities between prevention and mitigation are perfectly noticeable while postvention probability changes are completely

different. It can be explained paying attention to the fact that both prevention and mitigation probabilities mostly depend on the incoming traffic. Nevertheless, incoming flow has influence on the postvention probability not being as important as filtering properties and time which passed from the beginning of the attack.

Fig. 5.13. DDoS attack survival probability and its component probabilities changes in time when no agent recruitment is used

Fig. 5.14. DDoS attack survival probability and its component probabilities changes in time when attack adds constant number of agents every second

Fig. 5.15. DDoS attack survival probability and its component probabilities changes in time when every time of moment the increasing number of agents is added to the attack

Fig. 5.16. DDoS attack survival probability and its component probabilities changes in time when every time of moment the decreasing number of agents is added to the attack

The research showed that prevention and mitigation probabilities are much more dependent on the attack traffic in comparison to the postention probability. So attempts to employ the countermeasure actions, seeking effective tracing back of the attack source, should be made at the beginning of attack only if the attack is going to last for long time.

Botnet agent allocation strategies influence the attack traffic dynamics in time respectively defining the attack power and effectiveness of the changes in time. Therefore expected attack progress can be estimated analyzing how attack traffic changes at the beginning of attack.

5.4. Conclusions for Chapter 5

1. Experiments with different system configurations were executed and showed that the composite DDoS attack model can be easily modified by eliminating not necessary components or adding new ones, such as filtering, traceability subsystems etc. Therefore the model can be applied to new types of DDoS attack as well.
2. All experiments with model characteristic combinations revealed that the biggest part of model properties depend on non linear changes of the model attack success probability or other characteristics. This proves that it can be difficult to find the optimal system configurations without mathematical or programming models.
3. Analysing Botnet agent allocation strategies in bandwidth exhaustion DDoS attack model different strategies were examined. It is demonstrated that for non zero agent allocation strategy there are at least three attack phases. Therefore the attack source should consider the optimal configuration to get the desired result (fast and powerful attack, long lasting attack, etc.).

General Conclusions

1. DoS attack taxonomy proposed by J. Mirkovich encompasses the biggest number of categories and the number of its taxonomy has the highest category minuteness level. However, our DoS attack taxonomy usage experiments demonstrated that proposed taxonomy is more suitable for ongoing DoS attacks. On average 8.6% more categories were marked in our taxonomy than the in taxonomy proposed by J. Mirkovich. Our results confirm that the most important DoS attack taxonomy characteristics are its structure and clearness, and not the number of properties.
2. Analysis of existing DoS and DDoS attacks taxonomies demonstrated tendency to use stochastic elements for representation of the internet traffic randomness. From 21 analysed models only seven did not use probabilistic expressions, all introduced before 2007. This allows to suggest that agent models are not able accurately represent huge amount of DDoS attack data and stochastic models should be used instead.
3. Composite DDoS attack success probability represents a probability that legitimate user request will be lost at least in one of model components (will be blocked in bandwidth, memory or CPU work exhaustion models). It inflicts model flexibility and extensibility.

- Therefore it can be adopted to represent new type of DDoS attacks as well.
4. There are no reliable and detailed historical data of DDoS attacks while experiments in LAN cannot represent all characteristics of DDoS attack. Therefore we use OPNET modelling tool to validate our proposed DDoS attack models. Validation results proved that there exist model data similarities to OPNET modelling results. Therefore DDoS attack models that we propose can be used for estimation of theoretical attack success probability. However proposed models do not guarantee the precision of real DDoS attacks success probability.
 5. Numerous experiments with composite DDoS attack model showed that the main component of composite DDoS attack success probability can vary depending on more than 5 parameters. Therefore using the composite DDoS attack model for hosting ranking purposes at least in a few standard situations should be analysed.

References

- Abry, P.; Veitch, D. 1998. Wavelet analysis of long-range-dependent traffic, *IEEE Transactions on Information Theory* 44.
- Agha, G.; Gunter, C. A.; Greenwald, M.; Khanna, S.; Meseguer, J.; Sen, K.; Thati, P. 2005. Formal modeling and analysis of DoS using probabilistic rewrite theories, in *Proc of – International Workshop on Foundations of Computer Security*, Chicago.
- Aiello, W.; Bellovin, S. M.; Blaze, M.; Canetti, R.; Ioannidis, J.; Keromytis, A. D.; Reingold, O. 2004. Just fast keying: Key agreement in a hostile internet, *ACM Transactions on Information and System Security*, 7 (2): 242–273.
- Aissani, A. 2008. Queueing Analysis for Networks Under DoS Attack, in *Proc. of – IAENG International Conference on Communication Systems and Applications*, Hong Kong: 500–513.
- Altman, E. 1994. Flow control using the theory of zero-sum markov games, *IEEE Transaction on Automatic Control* 39: 814–818.
- AlTurki, M.; Meseguer, J.; Gunter, C. A. 2009. Probabilistic modeling and analysis of DoS protection for the ASV protocol. *Electronic Notes in Theoretical Computer Science*, 234: 3–18.

- Anderson, D. R.; Sweeney, D. J.; Williams, T. A. 2002. *An Introduction to Management Science: Quantitative Approaches to Decision Making*. 10th edition. South-Western Educational Publishing.
- Asosheh, A.; Ranezani, N. 2008. A comprehensive taxonomy of DDoS attacks and defense mechanism applying in a smart classification, *WSEAS Transactions on Communications* 7 (4): 281–290.
- Aycock, J. 2006. *Computer Viruses and Malware*. New York: Springer.
- Bencsath, B.; Vajda, I.; Buttyan, L. 2003. A game based analysis of the client puzzle approach to defend against dos attacks, in *Proc. of – 2003 International Conference on Software, Telecommunications and Computer Networks*, Split: 763–767.
- Beran, J.; Sherman, R.; Taqqu, M.; Willinger, W. 1997. Variable-bit-rate video traffic and long range dependence, *IEEE Transactions on Communications*, 43:1566–1579.
- Bermudez, A.; Casado, R.; Quiles, F. J.; Pinkston, T. M.; Duato, J. 2003. Modeling InfiniBand with OPNET, in *Proc. of – Second Annual Workshop Novel Uses of System Area Networks*.
- Bernstein, D. J. 1999 [online]. SYN cookies. Cr.yp.to: [cited 14 March 2011]. Available from internet: <<http://cr.yp.to/syncookies.html>>.
- Bose, S. K. 2010 [online]. M/G/m/m Loss System. Nanyang Technological University: [cited 14 July 2011]. Available from internet: <http://www3.ntu.edu.sg/home/eskbose/qbook/MGmm_Queue.PDF>.
- Bright Hub [online], 2010. Common Security Terms Dictionary: C to D: [cited 16 March 2011]. Available from internet: <<http://www.brighthouse.com/computing/smb-security/articles/8391.aspx>>.
- Burdach, M. 2010 [online]. Hardening the TCP/IP stack to SYN attacks. Symantec: [cited 14 March 2011]. Available from internet: <<http://www.securityfocus.com/infocus/1729>>.
- Cao, J.; Cleveland, W.S.; Gao, Y.; Jeffay, K.; Smith, F. D.; Weigle, M.C. 2004. Stochastic Models for Generating Synthetic HTTP Source Traffic, in *Proc. of – IEEE INFOCOM*, Hong Kong: 1547–1558.
- Cao, Z.; Guan, Z.; Chen, Z.; Hu, J.; Tang, L. 2007. An economical model for the risk evaluation of DoS vulnerabilities in cryptography protocols, in *Proc. of – The Third International Conference on Information Security Practice and Experience*, Hong Kong: 129–144.
- CCITT X.800, 1991. Data Communication Networks: Open Systems Interconnection (OSI); Security, Structure and Applications. Security Architecture for Open Systems Interconnection for CCITT Applications.

CERT [online]. 1998. CERT® Advisory CA-1998-01 Smurf IP Denial-of-Service Attacks. CERT: [cited 14 September 2011]. Available from internet: <<http://www.cert.org/advisories/CA-1998-01.html>>

CERT [online]. 1999. Denial of Service Attacks: [cited 14 March 2011]. Available from internet: <http://www.cert.org/tech_tips/denial_of_service.html>.

Chang R. K. C. 2002. Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial, *IEEE Communications Magazine*. 40: 42-51.

Chaturvedi, N.; Mohant, H. 2011. A Mathematical Model for Randomly-Occurring Low-Rate Denial of Service Attack, in *Proc. of – First Student Research Symposium in conjunction with Seventh ICDCIT-2011*, Bhubaneswar.

Chen, T. M. 2007. Network Traffic Modeling, *Chapter in – The Handbook of Computer Networks*, Hossein Bidgoli (ed.), Wiley.

Choudary, K.; Shilpa, M. 2011. Smurf Attacks: Attacks using ICMP, *International Journal of Computer Science and Technology*, 2 (1):75–77.

Committee on National Security Systems 2010. CNSS Instruction No. 4009: National Information Assurance (IA) Glossary.

Diffie, W.; Oorschot, P. C.; Wiener, M. J. 1992. Authentication and authenticated key exchanges, *Designs, Codes, and Cryptography*, 2 (2): 107–125.

Douligeris, C.; Mitrokots, A. 2004. DDoS attacks and defense mechanisms: classification and state-of-the-art, *Computer Networks*, 4(2004): 643–666.

Eichin, M.; Rochlis, J. 1989. With Microscope and Tweezers: An Analysis of the Internet Virus of November 1988, in *Proc. of – IEEE Computer Society Symposium on Security and Privacy*, Oakland.

Fadlallah, A.; Serhrouchi, A. 2005. Denial of service attack and defense schemes, Analysis and Taxonomy, in *Proc. of – 3-rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications*, Tunisia.

Fallah. M. 2010. A Puzzle-Based Defense Strategy Against Flooding Attacks Using Game Theory, *IEEE Transactions on Dependable and Secure Computing*, 7 (1).

Feily, M.; Shahrestani, A; Ramadass, S. 2009. A Survey of Botnet and Botnet Detection, in *Proc. of – Third International Conference on Emerging Security Information, Systems and Technologies*, Athens.

Field, A. J.; Bradley, J. T. 2004. Analysing Clocked Process Algebras via Stochastic Petri Nets, *Department of Computing, Imperial College London*.

Fokink, W. 2007. Introduction to Process Algebra. *Computer Science – Monograph*. Springer-Verla.

Frost, V. S.; Melamed, B. 1994. Traffic Modeling For Telecommunications Netwrks. *IEEE Communications Magazine*, 33: 70–80.

Gligor, V. D. 1983. On Denial-of-Service in Computer Networks, in *Proc. ICDE*: 608–617.

Gligor, V. D.; Blaze, M.; Ioannidis, J. 2001. Denial of Service – Panel Discussion. SECURITY PROTOCOLS, *Lecture Notes in Computer Science* 2133/2001: 194–203.

Gong, L.; Syverson, P. 1998. Fail-stop protocols: An approach to designing secure protocols, *Chapter in Dependable Computing for Critical Applications* (ed. Iyer, R. K.; Morganti, M.; Fuchs W. K.; Gligor, V.), IEEE Computer Society.

Goranin, N.; Cenys, A. 2008. Genetic Algorithm Based Internet Worm Propagation Strategy Modeling, *Information Technology and Control* 37(2): 133–140. ISSN 1392-124X.

Goranin, N.; Cenys, A. 2008. Malware Propagation Modeling by the Means of Genetic Algorithms, *Elektronika ir elektrotechnika* 6(86): 23–26. ISSN 1392-1215.

Goranin, N.; Cenys, A. 2009. Genetic Algorithm Based Internet Worm Propagation Strategy Modeling Under Pressure of Countermeasures, *Journal of Engineering Science and Technology Review* 2(1): 43–47. ISSN 1791–2377.

Grizzard, J. B.; Sharma, V.; Nunnery, C.; Dagon, D. 2007. Peer-to-Peer Botnets: Overview and Case Study in *Proc. of – The first conference on First Workshop on Hot Topics in Understanding Botnets*, Cambridge.

Gu, Q.; Liu, P. 2008. Denial of Service Attacks, *The Handbook of Computer Networks 3, Distributed Networks, Network Planning, Control, Management, and New Trends and Applications*.

Heidari, M. 2006 [online]. The Role of Modeling and Simulation in Information Security The Lost Ring. Mega Security: [cited 14 June 2011]. Available from internet: <http://www.megasecurity.org/papers/The_Role_of_Modeling_and_Simulation_in_Information_Security.pdf>.

Hespanha, J. P.; Bohacek, S. 2001. Preliminary results in routing games, in *Proc. of – The American Control Conference*, 3:1904–1909.

Houghton Mifflin Company 2005 [online]. Model – definition of model. Free Online Dictionary, Thesaurus and Encyclopedia [cited 14 December 2011]. Available from internet: <<http://www.thefreedictionary.com/model>>.

Huang Q.; Kobayashi H.; Liu B. 2003. Analysis of a New Form of Distributed Denial of Service Attack, in *Proc. of – Conference of Information Science and Systems*, Baltimore.

Huang Q.; Kobayashi H.; Liu B. 2003. Modeling of Distributed Denial of Service Attacks in Wireless Networks, in *Proc. of – IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, 1: 41–44.

Jacquet, G.; Harba, R. 2004. Wavelet based estimator for fractal Brownian Motion: an Experimental Pint of View, in *Proc. of – EUSIPCO 2004*, Vienne: 175–178.

- Jain, R.; Routhier, S. A. 1986. Packet Trains – measurements and a new model for computer network traffic, *IEEE Journal on Selected Areas in Communications*, 4(6): 986–995.
- Jenkins, J. L.; Wang, J. L. 1997. A measurement analysis of ATM cell-level aggregate traffic, in *Proc. of – Global Telecommunications Conference GLOBECOM '97*.
- Juknius, J.; Čenys, A. 2008. BotNet prevencija interneto paslaugų tiekėjų lygmenyje, in *Proc. of – Science – Future of Lithuania*, Vilnius.
- Juknius, J.; Goranin, N. 2010. BotNet Spreading Detection and Prevention Via Websites, *Jaunųjų mokslininkų darbai*, 1 (26): 293–298.
- Kajackas, A.; Rainys, R. 2011. Estimation of critical components of internet infrastructure, *Elektronika ir elektrotechnika* 4(110): 35–38.
- Kajackas, A.; Rainys, R.; Aputis, A. 2011. Assessment of cyber attacks influence over internet network, *Elektronika ir elektrotechnika* 7(113): 89–92.
- Kajackas, A.; Rainys, R. 2010. Internet infrastructure topology assessment, *Elektronika ir elektrotechnika* 7(103): 91–94.
- Kaparti, R. [online]. OPNET IT Guru: A tool for networking education. REGIS University: [cited 14 Decenber 2011]. Available from internet: <http://www.opnet.com/university_program/teaching_with_opnet/textbooks_and_materials/materials/ITGAE_Tool_Ntwrk_Ed.pdf>.
- Karig, D.; Lee, R. 2001. Remote Denial of Service At-tacks and Countermeasures, *Princeton University Department of Electrical Engineering Technical Report CE-L2001-002*.
- Khan, S.; Traore, I. 2005. Queue-based Analysis of DoS Attacks, in *Proc. of – IEEE Workshop on Information Assurance and Security*, West Point: 266–273.
- Kim, M.; Stehr, M. O.; Talcott, C. L.; Dutt, N. D.; Venkatasubramanian, N. 2007. A probabilistic formal analysis approach to cross layer optimization in distributed embedded systems, *Formal Methods for Open Object-Based Distributed Systems*, 4468.
- Kotenko, I. 2005. Agent-Based Modeling and Simulation of Cyber-Warfare between Malefactors and Security Agents in Internet, in *Proc. of – 19th European Simulation Multiconference “Simulation in wider Europe”*, Lausanne.
- Kuma, A.; Paxso, V.; Weave, N. 2005. Exploiting Underlying Structure for Detailed Reconstruction of an Internetscale Event, in *Proc. of – ACM SIGCOMM Internet Measurement Conference*, Berkeley.
- Lafrance, S.; Mullins, J. 2003. An information flow method to detect denial of service vulnerabilities, *Journal of Universal Computer Science*.

- Laikas.lt 2011 [online]. Ažiotazas dėl bilietų į „EuroBasket 2011“ sutrikdė pardavimų sistemos darbą. Laikas.lt: [cited 24 January 2012]. Available from internet: <<http://www.laikas.lt/lt/info/3917/aziotazas-del-bilietu-i-eurobasket-2011-sutrikde-pardavimu-sistemos-darba/>>.
- Lauritis, R. 2003. Application of Neural Networks for Data Protection Research, *Electronics and Electrical Engineering* 4(46): 61–64.
- Leland, W. E.; Taqqu, M. S.; Wilinger, W.; Wilson D. V. 1993. On the self-similar nature of Ethernet traffic, in *Proc. of – Communications architectures, protocols and applications*, San Francisco: 183–193.
- Leland, W. E.; Taqqu, M. S.; Wilinger, W.; Wilson D. V. 1994. On the Self-Similar nature of Ethernet Traffic (extended version), *IEEE/ACM Transactions on Networking*, 2(1):1–15.
- Lemon, J. 2002. Resisting SYN flood DoS attacks with a SYN cache, in *Proc. of – USENIX BSDCon 2002*, San Francisco.
- Lin, S.; Chiueh, T. 2006. A Survey on Solutions to Distributed Denial of Service Attacks, *Stony Brook University*.
- Lye, K.; Wing. J. M. 2002. Game strategies in network security, in *Proc. of – The Workshop on Foundations of Computer Security*, Copenhagen.
- Macia-Fernandez, G.; Diaz-Verdejo, J. E.; Garcia-Teodoro, P. 2006. Mathematical foundations for the design of a low-rate DoS attack to iterative servers (short paper), *Lecture Notes in Computer Science*, 4307: 282–291.
- Macia-Fernandez, G.; Diaz-Verdejo, J. E.; Garcia-Teodoro, P. 2009. Mathematical Model for Low-Rate DoS Attacks against Application Servers, *IEEE Transactions On Information Forensics and Security*, 4 (3): 519–529.
- Mahathi, K.K. 2008. Botnets: Overview and Case Study, *IAS Program Master Project Defense*.
- Maheswaran, R.; Basar, T. 1998. Multi-user flow control as a nash game: Performance of various algorithms, in *Proc. of – 37th IEEE Conference on Decision and Control*, Tampa: 1090–1095.
- Mahimkar, A.; Shmatikov, V. 2005. Game-based analysis of denial-of-service prevention protocols, in *Proc. of – 18th IEEE Computer Security Foundations Workshop*, Aix-en-Provence.
- Marathe, M.; Hawe, W. 1982. Predicted capacity of Ethernet in a university environment, in *Proc. of – Southcon*, Orlando.
- Mavituna, F. 2008 [online]. *DoS Attacks Using SQL Wildcards*. Portcullis Computer Security: [cited 14 March 2011]. Available from internet: <http://www.portcullis-security.com/uploads/wildcard_attacks.pdf>.

- Meadows, M. 1999. A Formal Framework and Evaluation Method for Network Denial of Service, in *Proc of – 12th IEEE workshop on Computer Security Foundations*, Washington.
- Meadows, C. 2001. A cost-based framework for analysis of denial of service in networks, *Journal of Computer Security*, 9 (1-2): 143–164.
- Mirkovich, J.; Dietrich, S.; Dittrich, D.; Reiher, P. 2005. *Internet Denial of Service: Attack and Defense Mechanisms*. New Jersey: Prentice Hall.
- Mirkovic, J.; Reiher, P. 2002. A taxonomy of DDoS Attack and DDoS defense Mechanisms, *ACM SIGCOMM Computer Communication Review*, 34 (2).
- Mohorko, J.; Matjaz, F.; Sasa, K. 2008. Advanced Modelling and Simulation Methods for Communication Networks, *Microwave Review*: 41–46.
- Morparia, J. 2008. Peer-to-Peer Botnets: Analysis and Detection, *San Jose State University*.
- Moore, D.; Paxson, V.; Savage, S.; Shannon, C.; Staniford, S.; Weaver, N. 2003. Inside the Slammer Worm, *IEEE Security and Privacy*, 1 (4): 33–39.
- Moore, D.; Shannon, C.; Brown, J. 2002. Code-Red: a case study on the spread and victims of an Internet worm, in *Proc. of – Internet Measurement Workshop 2002*, Marseille: 273–284.
- Naseem, F.; Shafqat, M.; Sabir, U.; Shahzad, A. 2010. [15] A Survey of Botnet Technology and Detection, *International Journal of Video & Image Processing and Network Security*, 10 (1):13–17.
- Noreika, A.; Drąsutis, S. 2007. Website Activity Analysis Model. *Information Technology And Control*, 36 (3): 41–46.
- Norros, I. 1994. A storage model with self-similar input, *Queueing Systems*, 16:387-396.
- Erramilli, A.; Roughan, M.; Veitch, D.; Willinger, W. 2002. Self-Similar Traffic and Network Dynamics, in *Proc. of – the IEEE 90* (5).
- Pastore, M.; Dulaney, E. 2006. *CompTIA Security+ Study Guide: Exam SY0-101*, Wiley.
- Paulauskas, N. 2009 Analysis of Computer System Incidents and Security Level Evaluation: *Doctoral Dissertaion*. Vilnius Gediminas Technical University. Vilnius: Technica.
- Paulauskas, N.; Garšva, E. 2006. Computer System Security Incident Analysis, in *Proc.of – Seventh International Baltic Conference on Databases and Information Systems: Communications, Materials of Doctoral Consortium*. Vilnius, 336–339.
- Paulauskas, N.; Garsva, E. 2008. Attacker Skill Level Distribution Estimation in the System Mean Time-to-Compromise, in *Proc. of – the 1st International conference on information technology*. Gdansk, 463–466.

- Paulauskas, N.; Garsva, E.; Skudutis, J. 2009. Network Scan Detection Simulation, *Electronics and Electrical Engineering* 2(90): 43–46.
- Paulauskas, N.; Skudutis, J. 2008. Investigation of the Intrusion Detection System "Snort" Performance, *Electronics and Electrical Engineering* 7(87): 15–18.
- Paxson, V.; Floyd, S. 1995. Wide-Area Traffic: The Failure of Poisson Modeling, *IEEE/ACM Transactions on Networking*, 3(3): 226–244.
- Pearson, M.; Cleary, J.; Unger, B.; Williamson, C. 1996. Current Techniques for Measuring and Modeling ATM Traffic, *Technical Report, Computer Science Department, Waikato University*.
- Puigjaner, R. 2003. Performance Modelling of Computer Networks, in *Proc. of – The 2003 IFIP/ACM Latin America conference on Towards a Latin American agenda for network research*, La Paz: 106–123.
- Puniškis, D.; Laurutis, R. 2005. The Use of Neuron Networks for the Performance of Epidemics caused by computer viruses, *Electronics and Electrical Engineering* 4(60): 28–32.
- Puniškis, D.; Laurutis, R. 2007. Behavior Statistic based Neural Net Anti-spam Filters, *Electronics and Electrical Engineering* 6(78): 35–38.
- Puniškis, D.; Laurutis, R.; Dirmeikis, R. 2006. An Artificial Neural Nets for Spam email Recognition, *Electronics and Electrical Engineering* 5(69): 73–76.
- Ramakrishnan, K. K.; Tripathi, S. K. 1982. A common framework for studying the performance of channel access protocols, in *Proc. of – Computer Performance Evaluation Users Group*, Washington: 365–373.
- Riedi, R. H.; Vehel, J. L. 1997. Multifractal properties of TCP traffic: a numerical study, *Technical Report 3129, INRIA Rocquencourt*.
- Salah, K. 2010. Queuing Analysis of Network Firewalls, in *Proc. of – IEEE Global Communications Conference*, Miami.
- Schuba, C. L.; Krsul, I. V.; Kuhn, M. G.; Spafford, E. H.; Sundaram, A.; Zamboni, D. 1997. Analysis of a Denial of Service Attack on TCP, in *Proc. of – IEEE Symposium on Security and Privacy*, Oakland.
- SecuringMyNetwork [online]. 2011. Are you Denied Access to Any Website or Internet? SecuringMyNetwork: [cited 25 March 2011]. Available from internet: <<http://www.securingmynetwork.com/network-security-articles/denial-of-service-attack.php>>.
- Shenker, S. 1994. Making greed work in networks: A game-theoretic analysis of switch service disciplines, in *Proc. of – SIGCOMM Symposium on Communications Architectures and Protocols*, London: 47–57.

- Smith, J.; Gonzalez-Nieto, J.M.; Boyd, C. 2006. Modelling denial of service attacks on jfk with meadows's cost-based framework, in *Proc. of – The Fourth Australasian Information Security Workshop*.
- Sonic WALL, Inc. 2001 [online]. Denial of Service Attacks: An Emerging Vulnerability for the “Connected” Network. SonicWALL: [cited 14 March 2011]. Available from internet: < ftp://ftp.sonicwall.com/pub/info/Denial_Of_Service_Attacks.pdf>.
- Specht, S. M.; Lee, R. B. 2004. Distributed Denial of Service: Taxonomy of Attacks, Tools and Countermeasures, in *Proc. of – 17th International Conference on Parallel and Distributed Computing Systems*, San Francisco.
- Staniford, S.; Paxson, V.; Weaver, N. 2002. How to Own the Internet in Your Spare Time, in *Proc. of – 11th USENIX Security Symposium*, San Francisco.
- Stone-Gross, B.; Cova, M.; Cavallaro, L.; Gilbert, B.; Szydlowski, M.; Kemmerer, R.; Kruegel, C.; Vigna, G. 2009. Your Botnet is My Botnet: Analysis of a Botnet Takeover, in *Proceedings of the ACM CCS*, Chicago.
- Thing, V. L. L.; Sloman, M.; Dulay, N. 2007. Enhanced TCP SYN Attack Detection, in *Proc. of – IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation (MonAM)*, Toulouse.
- Vedel, J. L.; Riedi, R. H. 1997. Fractional Brownian motion and data traffic modeling: The other end of the spectrum, *Chapter in – Fractals in Engineering'97* (ed. Vedel, J. L.; Lutton, E.; Tricot, C.).
- Vishwanath, K. V.; Vahdat, A. 2006. Realistic and responsive network traffic generation, in *Proc. of – 2006 Conference on Applications, Technologies, Architectures, and Protocols For Computer Communications*, Pisa.
- Zou, C. C.; Towsley, D.; Gong, W. 2006. On the performance of Internet worm scanning strategies, *Elsevier Journal of Performance Evaluation*, 63(7): 700–723.
- Wang, P.; Sparks, S.; Zou, C. C. 2007. An Advanced Hybrid Peer-to-Peer Botnet, in *Proc. of – The first conference on First Workshop on Hot Topics in Understanding Botnets*, Cambridge.
- Wang Y.; Lin C.; Li Q.; Fang Y. 2007. A queueing analysis for the denial of service (DoS) attacks in computer networks, *Computer Networks*, 54: 3564–3573.
- Weave, N. C. 2002 [online]. *Warhol Worms: The Potential for Very Fast Internet Plagues*. The TAZ: [cited 25 April 2011]. Available from internet: <http://taz.newffr.com/TAZ/_VX_/papers/avers/warhol_worms.pdf>.

The List of Scientific Author's Publications on the Subject of the Dissertation

Papers in the Reviewed Scientific Journals

Ramanauskaitė, S.; Čenys, A. 2009. DoS atakų modeliavimas stochastiniais metodais. *Jaunųjų mokslininkų darbai* 3 (24):. 97-101. ISSN 1648-8776.

Ramanauskaitė, S. 2010. Modeling of SYN Flooding Attacks. *Jaunųjų mokslininkų darbai* 1 (26): 331-335. ISSN 1648-8776.

Ramanauskaitė, S.; Juknius, J. 2010. BOTNET agentų naudojimo DDoS atakose strategijų modeliavimas. *Jaunųjų mokslininkų darbai* 3 (28): 114-119. ISSN 1648-8776.

Ramanauskaitė, S.; Čenys, A. 2011. Stochastinis TCP SYN atakų modelis. *Science – Future of Lithuania* 3 (1): 20-24. ISSN 2029-2341/ISSN 2029-2252.

Ramanauskaitė, S.; Čenys, A. 2011. Taxonomy of DoS Attacks and Their Countermeasures. *Central European Journal of Computer Science* 1 (3): 355-366. ISSN: 1896-1532/ ISSN: 2081-9935.

Ramanauskaitė, S.; Čenys, A. 2012. Composite DoS Attack Model. *Science – Future of Lithuania* 4 (1): 20-26. ISSN 2029-2341/ISSN 2029-2252.

Other Papers

Ramanauskaitė, S.; Čenys, A. 2011. Modelling of Central Processing Unit Work Denial of Service Attacks. in *Proc. of – The 17th International Conference on Information and Software Technologies*, Kaunas, 99–104. ISSN 2029-0063/ISSN 2029-0020.

Simona RAMANAUSKAITĖ

MODELLING AND RESEARCH OF DISTRIBUTED
DENIAL OF SERVICE ATTACKS

Doctoral Dissertation

Technological Sciences,
Informatics Engineering (07T)

Simona RAMANAUSKAITĖ

SRAUTINIŲ ATAKŲ ĮTAKOS INTERNETINĖS PASLAUGOS SUTRIKDYMUI
MODELIAVIMAS IR TYRIMAS

Daktaro disertacija

Technologijos mokslai,
informatikos inžinerija (07T)

2012 04 23. 11,0 sp. l. Tiražas 20 egz.
Vilniaus Gedimino technikos universiteto
leidykla „Technika“,
Saulėtekio al. 11, 10223 Vilnius,
<http://leidykla.vgtu.lt>
Spausdino UAB „Ciklonas“
J. Jasinskio g. 15, 01111 Vilnius.