A NOVEL SYSTEM MODEL IMPLEMENTATION OF NETWORK ATTACKS AND COUNTERMEASURES

by

Beytül İnal

B.S., Electrical-Electronics Engineering, Yıldız Teknik University, 2008

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical-Electronics Engineering
Boğaziçi University

2012

ACKNOWLEDGEMENTS

There are numerous people I would like to thank due to their support and encouragement in completion of this thesis.

First of all, I am heartily thankful to my supervisor, Prof. Emin Anarım, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject. He has truly inspired me during my research, provided helpful hints and always supported me through the good and the bad times. So, It is an honour for me to work with him. I also would like to thank Tolga Kurt for his genuine support, valuable advice and sincere comments which helped me a lot to finish this study.

My thanks also go to my thesis juries Prof. Kemal Cılız and Assoc. Prof. Fatih Alagöz for spending their rare available time on examining my thesis. I also would like to thank ADAX to support my thesis project. In addition, I would like to thank TÜBİTAK for financial support.

Finally, I want to thank my family due to their patience. I am particularly grateful to my friend, Bilgehan Öz, for helping and assisting me in all the stages of this work. Without his help this study would never have been possible.

ABSTRACT

A NOVEL SYSTEM MODEL IMPLEMENTATION OF NETWORK ATTACKS AND COUNTERMEASURES

The Internet is an open and efficient TCP/IP protocol which is most widely used in the world. As usage of the Internet and TCP/IP protocols increases, their lack of built-in security has become more and more problematic. Nowadays, network attacks are the most important and serious problem on the Internet. DoS attack is the one of these network attacks which takes advantage of the lack of authenticity in the IP protocol and stateless nature of the Internet. This thesis examines the TCP/IP architecture, general network structures and network security mechanisms. Second, DDoS attacks and server system models are covered in this work. New attack models are proposed and developed. These new models and also existing attack models are simulated. While generating different attacks, new proposed server model is also designed and coded. This server is called "virtual server with a new perspective". Third, after generating the custom attacks against the custom server system, protection and warning mechanisms are required on the server side. In that case, custom intrusion detection and intrusion prevention system models are designed. In addition, numerous of simulation are completed. Finally, at the end of the research, many point of views about conutermeasures systems are proposed, and their results are investigated.

ÖZET

AĞ SALDIRILARI VE ÖNLEMLERİ KONUSUNDA YENİ BİR SİSTEM MODELİ UYGULAMASI

Internet, açık ve etkin TCP/IP protokolüdür ve dünyada yaygın biçimde kullanılmaktadır. İnternet kullanımı ve TCP/IP protokollerinin kullanımı arttıkça, yapısal güvenlik eksiklikleri önemli bir problem haline gelmiştir. Günümüzde, ağ saldırıları, Internet üzerindeki en önemli ve ciddi sorun olarak gözükmektedir. Uygulamaların servis veremeyecek hale getirilmesine sebep olan saldırılar, IP protokolünün güvenlik eksikliğinden yararlanarak geliştirilmektedir. Bu tez dahilinde TCP/IP mimarisi, genel ağ yapıları ve ağ güvenlik mekanizmaları konularında incelemelerde bulunulmuştur. İkinci olarak, uygulamaların servis veremeyecek hale getirilmesine sebep olan saldırılar ve sunucu sistem modelleri bu çalışmada ele alınmıştır. Yeni saldırı modelleri önerilmiş ve geliştirilmiştir. Bu yeni modeller ve mevcut saldırı modelleri simüle edilmiştir. Farklı saldırılar oluştururken, aynı zamanda, yeni sunucu (saldırıya maruz kalan, hedef) modeli de tasarlanmış ve gerçeklenmiştir. Üçüncü olarak, özel sunucu sistemine karşı saldırılar oluşturduktan sonra, sunucu tarafında koruma ve uyarı mekanizmalarına ihtiyaç duyulmuştur. Bu durumda, özel saldırı tespit ve saldırı önleme sistemleri tasarlanmıştır. Ek olarak, bu sistemlerin birçok simülasyonu yapılmıştır. Sonuç olarak, tez çalışmasında, çok sayıda karşı önlem sistemleri hakkında görüş sunulmuş ve bunların sonuçları incelenmiştir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZET v
LIST OF FIGURES viii
LIST OF ACRONYMS/ABBREVIATIONS xi
1. INTRODUCTION 1
2. TCP/IP PROTOCOL ARCHITECTURE AND STANDARDS 3
2.1. TCP/IP Architectural Model 3
2.1.1. Internetworking (Internet) 3
2.1.2. The TCP/IP Protocol Layers 4
2.2. TCP/IP Standards 6
2.2.1. Request for Comments (RFC) 7
2.2.2. Internet Standards 9
2.2.3. The OSI Reference Model 10
2.3. Network Architecture and Security 12
2.3.1. Networking Concepts and Components 12
2.3.1.1. Models of Network Computing 13
2.3.1.2. Network Models 14
2.3.1.3. Network Topologies 14
2.3.2. Network Security 15
2.3.2.1. OSI Security Architecture 15
3. DDoS ATTACKS AND SECURITY MECHANISM 17
3.1. DDoS Attacks 17
3.1.1. Preparation of DDoS Attack 19
3.1.2. DDoS Taxonomy 20
3.1.3. DDoS Attack Classification 21
3.1.3.1. Architecture Base DDoS Classification 21
3.1.3.2. Degree of Automation Base 26
3.1.3.4. Vulnerability Base 28

3.1.3.5. Attack Rate Dynamic Base 29
3.1.3.6. Scanning Strategy Base 30
3.1.3.7. Propagation Strategy Base 30
3.1.3.8. Packet Content Base 31
3.2. DDoS Timeline and DDoS Incident 31
3.3. DDoS Countermeasures 34
3.3.1. DDoS Mitigation 34
3.3.2. DDoS Prevention 39
3.3.3. DDoS Deterrence 41
4. PROPOSED SYSTEMS 43
5. SYSTEM IMPLEMENTATIONS AND SIMULATIONS 51
5.1. DDoS Attack Tool and Features 51
5.1.1. Port Scanning 52
5.1.2. Other Features 53
5.2. DDoS Counter Measures 57
5.2.1. Intrusion Detection Systems (IDS) 58
5.2.2. Intrusion Prevention Systems (IPS) 62
5.2.2.1. SYN Proxy 64
5.2.2.2. SYN Killer (Transparent Gateway) 66
5.2.2.3. SYN Killer (Evil IP) 69
5.3. DDoS Attacks Tool Properties in Terms of Literature 71
5.4. DDoS Defense Tool Properties in Terms of Literature 75
5.5. Virtual Server with New Perspective 77
6. CONCLUSION AND FUTURE WORKS 80
REFERENCES 82

LIST OF FIGURES

Figure 2.1. Two interconnected sets of networks, each seen as one logical network [1]. 4

Figure 2.2. The TCP/IP protocol stack [1]. 5

Figure 2.3. The OSI model is composed of seven layers [1]. 11

Figure 3.1. DDoS attack taxonomy [2]. 22

Figure 3.2. Trinoo. 24

Figure 3.3. IRC based attack model. 25

Figure 3.4. Detecting reflector attacks. 26

Figure 3.5. Taxonomy of DDoS countermeasures [3]. 35

Figure 4.1. TCP connection's establishment and release. 43

Figure 4.2. New proposed client tool. 45

Figure 4.3. New proposed virtual server. 47

Figure 4.4. Realtime graph. 50

Figure 4.5. Realtime logs. 50

Figure 5.1. DDoS attack client. 52

Figure 5.2. Dos attack client (windows command prompt display). 53

Figure 5.3. DDoS attack client. 55

Figure 5.4. DDoS attack client (wireshark logs). 55

Figure 5.5. DDoS attack client. 56

Figure 5.6. DDoS attack client (wireshark logs). 56

Figure 5.7. DDoS attack client. 57

Figure 5.8. DDoS attack client (wireshark logs). 57

Figure 5.9. IDS work flow. 60

Figure 5.10. IDS online monitoring work flow. 61

Figure 5.11. IDS tool (real time connection monitoring feature). 62

Figure 5.12. IDS tool (user warning feature). 63

Figure 5.13. IPS tool (general view). 63

Figure 5.14. SYN proxy work flow. 65

Figure 5.15. SYN proxy tool. 66

Figure 5.16. SYN Proxy work flow. 67

Figure 5.17. SYN killer transparent gateway tool. 68

Figure 5.18. SYN Killer Transparent Gateway work flow. 70

Figure 5.19. SYN killer evil IP tool. 71

Figure 5.20. SYN Killer Evil IP Address work flow. 72

Figure 5.21. SYN client tool and DDoS taxonomy. 73

Figure 5.22. IPS/IDS mechanisms tool and taxonomy. 76

Figure 5.23. New proposed virtual server. 78

Figure 5.24. Decision algorithm flow chart. 79

LIST OF ACRONYMS/ABBREVIATIONS

ACK	Acknowledgement
API	Application Programming Interface
BGP	Border Gateway Protocol
CGI	Common Gateway Interface
CPU	Central Processing Unit
DoS	Denial of Service
DDoS	Distributed Denial of Service
DNS	Domain Name System
EBCDIC	Extended Binary Communication Data Interchange Code
FDDI	Fiber Distributed Data Interface
FIN	Final
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
IAB	Internet Activities Board
IANA	Internet Assigned Number Authority
IBM	International Business Machines
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IESG	Internet Engineering Steering Group
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IP	Internet Protocol
IPv4	Internet Protocol Version 4
ISO	International Organization for Standardization
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector (was CCITT)
LAN	Local Area Network
MAC	Message Authentication Code

MAC	Medium Access Control
NANOG	North American Network Operators' Group
NCP	Network Control Protocol
NUI	National University of Ireland
OS	Operating System
OSI	Open Systems Interconnect
P2P	Peer to Peer
QoS	Quality of Service
RARP	Reverse Address Resolution Protocol Algorithm
RST	Reset
RFC	Request for Comments
SNA	System Network Architecture
SQL	Structured Query Language
STD	Standard Number
SYN	Synchronization
SYN_ACK	Synchronize Acknowledge
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TFN	Tribe Flood Network
TTL	Time to Live
UDP	User Datagram Protocol
URL	Uniform Resource Locator
WWW	World Wide Web
X.25	CCITT Packet Switching Standard

1. INTRODUCTION

The Internet is a worldwide collection of computer networks, cooperating with each other to exchange data using TCP/IP protocol which is common protocol in the world. People use the Internet for many purposes such as, do and share the research, communicating with others and transmitting files via E-mail, requesting and providing assistance with problems and questions, marketing and publicizing products and services etc. When people do these various of operations, the privacy, security and server service availability concepts become more and more important for the providers and the users also. Nowadays, network attacks are the most serious problem on the Internet. Furthermore, DoS attack is the most common network attacks which takes advantage of the lack of authenticity in the IP protocol and stateless nature of the Internet. On the other hand, while the attack mechanisms are being developed so fast, the countermeasure mechanisms are also trying to be developed by the software engineers day by day. The thesis covers the whole situations together and proposes new algorithms on them.

Second, DoS attacks and server system models are covered in this work. New attack models are proposed and developed. These new models and also existing attack models are simulated. While generating different attacks, a new proposed server model is also designed and implemented.

Third, after generating the custom attacks against the custom server systems, protection and warning mechanisms are required on the server side. In that case, custom intrusion detection and intrusion prevention system models are designed. In addition, numerous of simulation are completed.

Finally, at the end of the research, many point of views about countermeasures systems are proposed, and the results are investigated.

The thesis is organized as follows:

- In Chapter 2, we give an overview of TCP/IP protocol architecture and standards. Inddition, we review the general network architectures illustrating the main components.
- In Chapter 3, we introduce DDoS attacks and security mechanisms. The types of DDoS attack mechanisms, DDoS timeline and DDoS countermeasures are handled in this chapter in detail.
- In Chapter 4, new proposed system are introduced. A new intrusion detection system model, a new virtual server client model, and a new attack tool clients are proposed in this thesis. In the next chapter, the system implementations of proposed system and also some current systems are shared. In addition the simulation results are added.

Finally, Chapter 6 summarizes the contributions of this thesis and draws the main conclusions. It also points out the future research lines.

2. TCP/IP PROTOCOL ARCHITECTURE AND STANDARDS

The Transmission Control Protocol/Internet Protocol (TCP/IP) concept has become the industry-standard method of interconnecting hosts, the networks, and the Internet. As such, it is seen as the infrastructure behind the Internet and networks worldwide.

2.1. TCP/IP Architectural Model

The TCP/IP protocol concept is composed of two most important protocols: Transmission Control Protocol (TCP) and Internet Protocol (IP). A less used name for it is the Internet Protocol Suite, that is the phrase used in official Internet standard documents. TCP/IP was created in 1983 and then NCP is replaced by TCP/IP. The advantage of TCP/IP is it's variability. It can successfully switch packets of all shapes and sizes, and work across a varieties of networks.

2.1.1. Internetworking (Internet)

The main design goal of TCP/IP was to build an interconnection of networks, referred to as an internetwork, or Internet, that provides universal communication services over physical networks.

The words internetwork and Internet are simply an abbreviation of interconnected network. If we write the Internet with a capital "I", the Internet refers to the worldwide set of interconnected networks. Therefore, the Internet is an Internet, but the reverse does not apply. The connected Internet is also called to the Internet [1].

The Internet consists of these four main groups of networks:

- Backbones: Backbones networks known as network access points (NAPs) or Internet Exchange Points (IXPs) are large networks that used for interconnecting other networks.
- Regional: Regional networks connecting, for example, universities and colleges.
- Commercial: Commercial networks belongs to the commercial organizations for internal use that also have connections to the Internet. Those networks are used to provide access to the backbones so to subscribers.
- Local: Local networks are small networks, such as campus-wide university networks or company owned network for internal use.

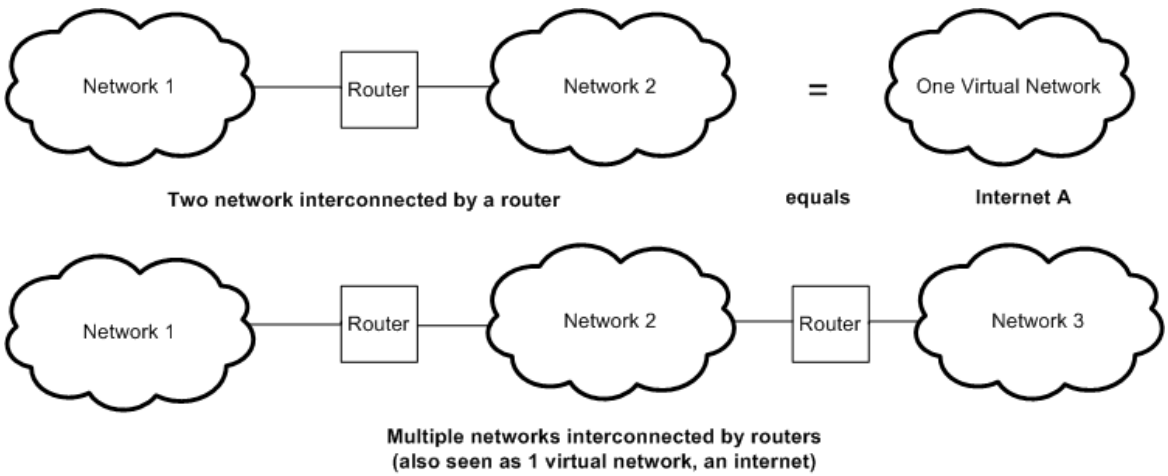

Figure 2.1. Two interconnected sets of networks, each seen as one logical network [1].

TCP/IP protocol is a standard communication mechanisms and there is no dependency to the physical network in TCP/IP. TCP/IP provides a common communication interface for the applications, therefore there is no need to know about the physical interface by the developers or users. The application needs only code to the standardized communication abstraction to be able to function under any type of physical network and operating platform [1].

2.1.2. The TCP/IP Protocol Layers

TCP/IP is modelled into layers. By dividing the protocols into layers provides a service for the layer directly above it and makes use of services provided by the

layer directly below it. Each layer has different responsibility for communication of two or more hosts/networks. For example, transferring data from one host to another without any guarantee to reliable delivery or duplicate suppression is responsible for the IP layer, on the other hand transport protocols such as TCP make use of this service to provide applications with reliable, in-order, data stream delivery [1].

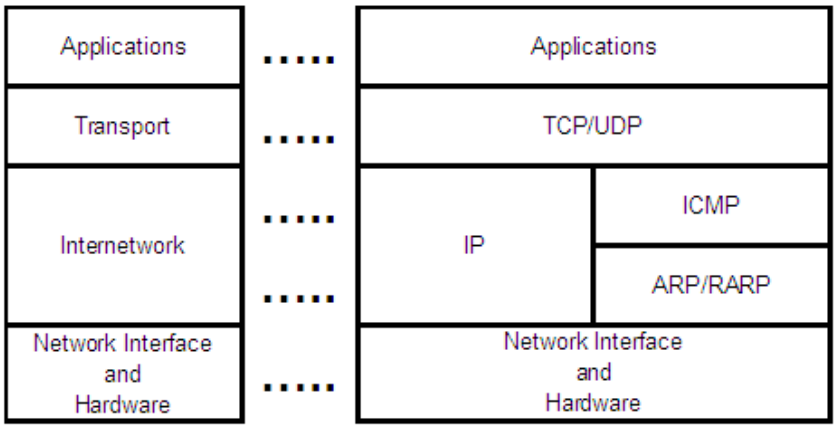

Figure 2.2. The TCP/IP protocol stack [1].

TCP/IP Protocol Stack Layers are divided into four main layers [1]. These are:

Application layer: The application layer is provided by the program that uses TCP/IP for communication. Examples of applications include Telnet and the File Transfer Protocol (FTP).

Transport layer: The transport layer provides the end-to-end data transfer by delivering data from an application to its remote peer. Examples of transport layer protocol are Transmission Control Protocol (TCP), and User Datagram Protocol (UDP)

Internetwork layer/Network layer: This layer is responsible for routing messages/data through Internet. Examples of internetwork layer protocols are IP, ICMP, IGMP, ARP, and RARP. *Network interface layer / Data Link layer:* The network interface layer is the interface to the actual network hardware. Examples of network

interface layer protocols are IEEE 802.2, X.25, ATM, FDDI, and SNA.

2.2. TCP/IP Standards

Due to the fact TCP/IP has a free and continuous renewal, it has been very popular between developers. Although there is no overall governing body to issue directives for the Internet, mutual cooperation Internet Society (ISOC) serves as the standardizing body for the Internet community. The societies that contribute to compose of Internet Standards [1]:

- IAB: ISOC is organized and managed by the Internet architecture Board (IAB).
- IEFT: The IAB itself relies on the Internet Engineering Task Force (IETF) for issuing new standards.
- IANA: The IAB itself relies on the Internet Assigned Numbers Authority (IANA) for coordinating values shared among multiple protocols.
- RFC: The RFC Editor is responsible for reviewing and publishing new standards documents.
- IESG: The IETF is governed by the Internet Engineering Steering Group (IESG)

The aims of the Internet Standards Process are:

- Technical excellence
- Prior implementation and testing
- Clear, concise, and easily understood documentation
- Openness and fairness
- Timeliness

The process of standardization is divided into 5 steps [1]:

- Step 1: Submitting the specifications to the IESG
  (i) Duration: No shorter than two weeks and no longer than six months

(ii) Details: In order to have a new specification approved as a standard, applicants have to submit that specification to the IESG where it will be discussed and reviewed technically and also published as an Internet draft document.

- Step 2: Last-call notification
  (i) Duration: Short term
  (ii) Details: After the IESG reaches a positive conclusion, it issues a last-call notification to allow the specification to be reviewed by the whole Internet community.
- Step 3: Inclusion in to Standards Track
  (i) Duration: Short term
  (ii) Details: After the final approval by the IESG, an Internet draft is recommended to the Internet Engineering Taskforce (IETF), another subsidiary of the IAB, for inclusion into the Standards Track and for publication as a Request for Comments.
- Step 4: Revising (Not always)
  (i) Duration: Short term
  (ii) Details: Once published as an RFC, It may also be revised over time or phased out when better solutions are found.
- Step 5: Removing from Internet drafts
  (i) Duration: Short Term
  (ii) Details: If the IESG does not approve of a new specification after, or if a document has remained unchanged within, six months of submission, it will be removed from the Internet drafts directory.

2.2.1. Request for Comments (RFC)

Request for Comments (RFC) helps Internet Protocol to make progress and develop day by day. Researchers design and implement new protocols and bring to the attention of the Internet community in the form of an Internet draft (ID). However, anyone can submit a memo proposed as an ID to the RFC Editor. RFC/ID authors must follow a set of rules those described in RFC (RFC 2223) in order for an RFC

to be accepted. After an RFC has been published, all revisions and replacements are published as new RFC. Therefore, the existing RFC is said to be "updated by" or "obsoleted by" the new one [1].

There are two types of RFCs:

- Information documents
- Internet protocols

The Internet Architecture Board (IAB) maintains a list of the protocol suite RFCs. These RFCs are assigned a state and a status [1].

The state types of an Internet protocol are:

- Standard

  The IAB has established this standard Internet protocol type as an official protocol for the Internet. These are separated into two groups [1]:

  (i) IP protocol and above, protocols that apply to the whole Internet

  (ii) Network-specific protocols, generally specifications of how to do IP on particular types of networks
- Draft Standard

  Draft standard protocol is considered as a possible standard protocol by the. Widespread testing and comments are required. There is a possibility that changes will be made in a draft protocol before it becomes a standard [1].
- Proposed Standard

  Proposed standard is a proposal that might be considered by the IAB for standardization in the future. Implementations and testing by several groups are required [1].
- Experimental

  Unless it is participating in the experiment and has coordinated its use of the protocol with the developer of the protocol, the system should not implement an experimental protocol.

- Informational

  Protocols developed by other standard organizations may be published as RFCs for the convenience of the Internet community as informational protocols.
- Historic

  These protocols are not become standards in the Internet because they have been superseded by later developments. Due to lack of interest can be the other reason.

  The status types of a Protocol are [1]:

- Required: The required protocols must be implemented by the system.
- Recommended: The recommended protocol should be implemented by the system.
- Elective: An elective protocol may or may not be implemented by the system.
- Limited use: These protocols are for use in limited circumstances.
- Not recommended: These protocols are not recommended for general use.

2.2.2. Internet Standards

STD means the standard number. When a protocol reaches the standard state, it is assigned a standard (STD) number. The goal of STD numbers is to point out which RFCs describe Internet standards. STD numbers can reference multiple RFCs (If the specification of a standard is spread across multiple documents). STD numbers do not change when a standard is updated, unlike RFCs. For RFC, the number shows to a specific document. In addition, the RFC numbers are unique [1].

The important Internet standards are:

- STD 1 – Internet Official Protocol Standards [1]

  The state and status of each Internet protocol or standard is given with this standard. This standard also defines the meanings attributed to each state or status. It is issued by the IAB approximately three months.

- STD 2 – Assigned Internet Numbers [1]

  The numbers and other protocol parameters in the Internet protocol suite is assigned. It is issued by the Internet Assigned Numbers Authority (The current edition is RFC 3232).
- STD 3 – Host Requirements [1]

  The requirements for Internet host software is defined by this standard.

  The standard is diveded into three main parts:

  (i) RFC 1122 – Communications layer (Requirements for Internet hosts)

  (ii) RFC 1123 – Application and support (Requirements for Internet hosts)

  (iii) RFC 2181 – Clarifications to the DNS Specification
- STD 4 – Router Requirements [1]

  The requirements for IPv4 Internet gateway software is defined by this standard.

2.2.3. The OSI Reference Model

OSI model, first released in 1984 by the International Standards Organization (ISO), provides a useful structure for defining and describing the various processes underlying open systems networking [1].

The OSI model is a blueprint for vendors to follow when developing protocol implementations. The OSI model divides communication protocols into seven layers. Each layer addresses an important portion of the communication process [1].

The Seven OSI Layers are in detail:

The Physical Layer: It's a set of rules regarding the hardware used to transmit data. Items covered at this layer:

- The voltages used
- The timings of transmission
- The rules used for the initial handshaking connection

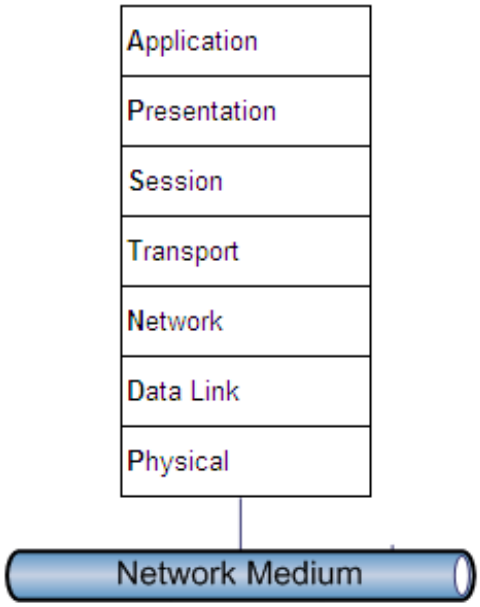

Figure 2.3. The OSI model is composed of seven layers [1].

The Data Link Layer: The physical layer provides the data link layer with bits. This layer provides the bits with some meaning. The data link layer adds flags to show the start and end of messages. This layer's standards perform two important tasks. Items covered at this layer [1]:

- Ensuring that data is not mistaken for flags
- Checking for errors within the frame

The Network Layer: The network layer, is concerned with packet switching. It establishes virtual circuits (Paths between terminals) for data communications. Items covered at this layer [1]:

- Repackaging messages from the transport layer above it into data packets, so the lower layers can transmit them.

The Transport Layer: The transport layer of the OSI model has many functions. Items covered at this layer [1]:

- Error recognitions and recoveries. As the highest order, the Transport layer can detect errors, identify packets that have been sent in the incorrect order, and then rearrange them.
- Regulating the information flow by controlling the messages movements.

The Session Layer: The session layer is concerned with the management of the network. The user communicates directly with this layer. Items covered at this layer:

- Verify passwords entered by the user
- Determining who uses the network, for how long, and for what purpose
- Controlling data transfers and even handles recovery from system crashes

The Presentation Layer: This layer is concerned with the network security, file transfers and formatting functions. Items covered at this layer [1]:

- Encoding data in a variety of different forms including ASCII and EBCDIC.

The Application layer: The application layer handles messages, remote logons and the responsibility of network management statistics. Items covered at this layer [1]:

- The database management programs, electronic mail, file server and printer server programs. The operating systems command and response language.

2.3. Network Architecture and Security

2.3.1. Networking Concepts and Components

A network is a group of interconnected systems sharing services and interacting by means of a shared communications link [4].

In Network concept, two or more individual systems are sharing something which

is called the data. The individual systems must be connected through a physical pathway or in technical words the transmission medium. All systems on the medium must follow a set of common communication rules for data transferring, data receiving and to understand each other. The communication rules are called protocols [4].

In summary, all networks must include the following items:

- Something to share which is called data
- A physical pathway which is called transmission medium
- Rules of communication which are called protocol

2.3.1.1. Models of Network Computing. There are three types of computing model:

- Centralized computing

  The earliest computers were large mainframes. Terminals, which came later, enabled users to reach and establish communication with the centralized computer systems. The terminals were not independent devices, just had input/output properties and that had no independent processing power.

  The features of this type of model are [4]:

  (i) All processing takes place in the central computer which is called, mainframe computer.

  (ii) Terminals are connected to the central computer mainframe and their function is only as input/output devices.

  (iii) Networks may be used to connect two or more mainframe computers.

- Distributed computing

  In distributed computing model, personal computers (PCs) came on the stage. PCs made it possible to give each worker an independent working ability. PCs can process and store data locally, without assistance from another machine. This is the biggest difference between personal computer and terminal device.

  The features of this type of model are [4]:

  (i) Single of multiple computers have ability to work and operating indepen-

dently.

(ii) Tasks which are given to the computers are completed locally.

(iii) It is possible to exchange data and services between computers by the networks.

- Collaborative or cooperative computing

  In collaborative type of network computing model, computers in a distributed computing environment have an ability to share processing power in addition to data, resources, and services [4].

  The features of this type of model are:

  (i) Multiple computers operating together to perform a task

  (ii) The computers exchange data and services by the help of network

  (iii) Software designed to take advantage of the collaborative environment.

2.3.1.2. Network Models. There are two types of network model:

- Server Based Networking Model

  It is also possible to call a server-client model to this type of network model. Because, there is a user-oriented PC (called client) that requests and receives network services from specialized computers called servers. Servers are generally higher-performance systems. Mainframes are also the biggest server system [4].

- Peer to Peer Networking Model

  A group of user-oriented PCs and each PC is called a peer that basically operate as equals. The peers can share resources, such as files and printers. The difference between this model and server based networking model is, there is no specialized server exists [4].

2.3.1.3. Network Topologies. A network topology is a kind of a plan for how the network cabling (OSI Layer 1) will interconnect the connection points (nodes) and how they will function in relation to one another.

There are two basic categories form of topologies:

- Physical topology: The actual layout of the network transportation path which is transmission medium
- Logical topology: The logical pathway is a signal follows as it passes among the network connection points (network nodes).

  Another way to think about this distinction is that a physical topology defines the way the network looks, and a logical topology defines the way the data passes among the nodes. Bus topologies, Ring topologies, and Star topologies can be count as an example of logical topology [4].

2.3.2. Network Security

Advent of the computer, the need for automated tools for protecting files and other information stored on the computer became an important concept. Indeed, the network security measures are needed to protect data during their transmission through media.

2.3.2.1. OSI Security Architecture. The International Telecommunication Union (ITU) Telecommunication Standardization Sector (ITU-T) is a United Nations-sponsored agency that develops standards, called Recommendations, relating to telecommunications and to open systems interconnection (OSI) [5].

This authority defines a systematic approach and it is called X.800, Security Architecture for OSI.The OSI security architecture is useful for organizing the task of providing security. In addition, since this architecture is an international standard, computer and communications vendors have developed security features of the products and services compatible with this structured definition of services and mechanisms.

The OSI security architecture is mainly interested in security attacks, mechanisms, and services. These items can be defined as follows:

- *Security attack:* If the attack action is made against the security of information

owned by an organization, it is called security attack

- *Security mechanism:* To detect, prevent, or recover from a security attack is called security mechanism
- *Security service:* A service that helps to improve the security of the data processing systems and the information transfers of an organization. To providing countermeasures against the security attacks is expecting from security services.

3. DDoS ATTACKS AND SECURITY MECHANISM

DoS attacks appoint an important problem in the Internet. The main aim in the DoS is the interrupting of services by attempting to limit access to a machine or service instead of corrupt the service itself. This kind of attacks purpose to make a network incapable of providing normal service by attacking the network's bandwidth or service connectivity. These attacks accomplish their goal by sending at a victim a stream of packets, like packet flood and make unavailable the network or its processing capabilities.

The logic of DDoS is a simple, but still it is very powerful technique to attack server (possibly servers run on Internet) resources. DDoS attacks add the many-to-one dimension to the DoS problem making the prevention more difficult and the impact proportionally severe. There are no distinctive characteristics of DDoS attacks that the victim could directly used for their detection [1].

DDoS attack is not a newly created attack technology. It was appeared firstly in late 1990s. In 2000, a massive DDoS attack occurred against very well-known web sites such as Yahoo, Amazon, etc. Although, it has been many years passed since DDoS attack first occured, it is still one of the most powerful attack, and biggest threats for Internet infrastructure and IT world.

3.1. DDoS Attacks

According to the WWW Security FAQ on DDoS attacks: "A DDoS attack uses many computers to launch a coordinated DoS attack against one or more targets. Using client/server technology, the perpetrator is able to multiply the effectiveness of the DoS significantly by harnessing the resources of multiple unwitting accomplice computers, which serve as attack platforms" [6].

In fact, the DDoS attack is the enhanced or improved version of DoS attack. It is

different from other attacks by its ability to put and work its agents in a “distributed” way over the Internet and to gather all these forces to create huge attack flood.

The aim of the DDoS attack is not trying to impair the target system. The main goal of a DDoS attack is to cause damage on a victim mostly for popularity reasons. Therefore, making any traditional security defense mechanism is inadequate.

DDoS attack action opens several security issues that can be exploited by any attackers.

Some of them are listed:

- Due to the fact that, Internet security is mutually dependent to all systems, it is not important how secure a victim's system is. DDoS victim depends on the rest of the global Internet [7].
- Of course, Internet resources are also limited, like other resources on the earth. Therefore, all Internet hosts has limited resources and these resources sooner or later can be consumed by sufficient number of users.
- Powerful against much more powerful. If the resources of the attackers are greater than the resources of the victims. It is really simple for the attacker, to consume victims' resources simply and the success is inevitable.
- Most of the resources needed for service guarantees is located in end user hosts. At the same time, in order to have large throughput and high bandwidth pathways are designed in the intermediate network. This way, attackers can exploit the abundant resources of an ignorant network in order to flood towards to the victim.

A DDoS attack is composed of four main items:

- The attacker: The real attacker, who plans and initiates the attack action.
- The handlers or masters: The handlers are compromised hosts with a special program running on them, capable of controlling multiple agents.
- The attack daemon agents or zombie hosts: The agents are compromised hosts

that are running a special program and are responsible for generating a stream of packets towards the intended victim. Those machines are commonly external to the victim's own network, to avoid efficient response from the victim, and external to the network of the attacker, to avoid liability if the attack is traced back.

- A victim or target host: The victim machine is probably a server service such as a website, under the attack action.

3.1.1. Preparation of DDoS Attack

The following steps take place while preparing and conducting a DDoS attack:

- Selection of agents: The attacker who is the brain and also the initiator chooses the agents that will perform the attack. These agent hosts need to have some vulnerability that the attacker can use to gain access to their systems. They should also have ample networks and system resources that will enable them to generate powerful attack floods. At the beginning, this process was performed manually, but it was soon automated by scanning tools.
- Compromise: The attacker exploits the security holes and vulnerabilities of the agent machines and plants the attack code. This time, the agent host starts to work for the attacker like a slave and he is unconscious of this status. Furthermore, the attacker always tries to protect the planted code from discovery of anyone (especially administrator of the agent host) and deactivation. Self-propagating tools such as the Ramen Worm [8] and Code Red [9] soon automated this phase. The administrator or the users of the agent systems have no idea that their system has been compromised part in a dangerous DDoS attack. When participating in a DDoS attack, each agent program uses only a small amount of resources in terms of memory and bandwidth, so that the users of computers experience minimal change in performance. Generally, virus protection programs cannot understand, this planted code which is running on the user machine, and not even trying to delete it.

- Communication: The handlers (one or more) and the attacker communicate to identify which agents are up and running. The attacker needs this information when he wants to schedule attacks, or upgrade the agents. Depending on how the attacker configures the DDoS attack network, agents can be instructed to communicate with a single handler or multiple handlers. The communication protocol that is used between attacker and handlers and between the handlers and the agents can be TCP, UDP, or ICMP protocols.
- Attack: At this step, the attacker commands the inception of the attack. The following features of the attack can be adjustable:
  (i) The victim
  (ii) The duration
  (iii) The type
  (iv) The length
  (v) TTL
  (vi) Port numbers

  Having ability to create various kind of attack packets can be beneficial for the attacker, in order to avoid detection and preventation.

3.1.2. DDoS Taxonomy

There are many DDoS attack taxonomies exist in the literature [3, 10–14].

A taxonomy is described in [3] is the to merge such a large work into a single, detailed classification.

In [11], they put forth for consideration that DDoS in two main branches based on vulnerability which are bandwidth depletion and resource depletion attacks.

Various classification criteria are explained in [12]. These criteria are degree of automation, exploited vulnerability, attack rate dynamics and impact.

In [13], they put emphasised on the attack networks, oppressed vulnerability,

impact of DDoS attack, attack intensity dynamics and the level of computerization.

In [14], they put emphasised on the classifications. They classified DDoS attacks by Congestion based, anomaly based and source based techniques and defense is classified by destination network and source network filtering.

A realistic model of DDoS simulation and experimentation have been put forth for consideration a formalized and scalable taxonomy in [13], in [14] authors introduced a framework for classifying DoS attacks based on header content, transient ramp-up behavior and novel techniques such as spectral analysis.

Eight features such as architecture, degree of automation, impact, vulnerability, attack rate dynamics, scanning strategy, propagation strategy and packet content are explained in [2].

3.1.3. DDoS Attack Classification

Eight features of DDoS attack taxonomy will be examined in this document. The architecture, degree of automation, impact, vulnerability, attack rate dynamics, scanning strategy, propagation strategy and packet content features will be described here in details.

3.1.3.1. Architecture Base DDoS Classification.

(i) Agent Handler Model Attack: Clients, handlers, and agents are playing role in this type of attack model.

- Client: Client helps the attacker to establish a communication with the other parts in the attack system.
- Handler (master): Which are compromised hosts with a special program running on them, capable of controlling multiple agents. The handlers are software codes and located in a network probably in the Internet. There-

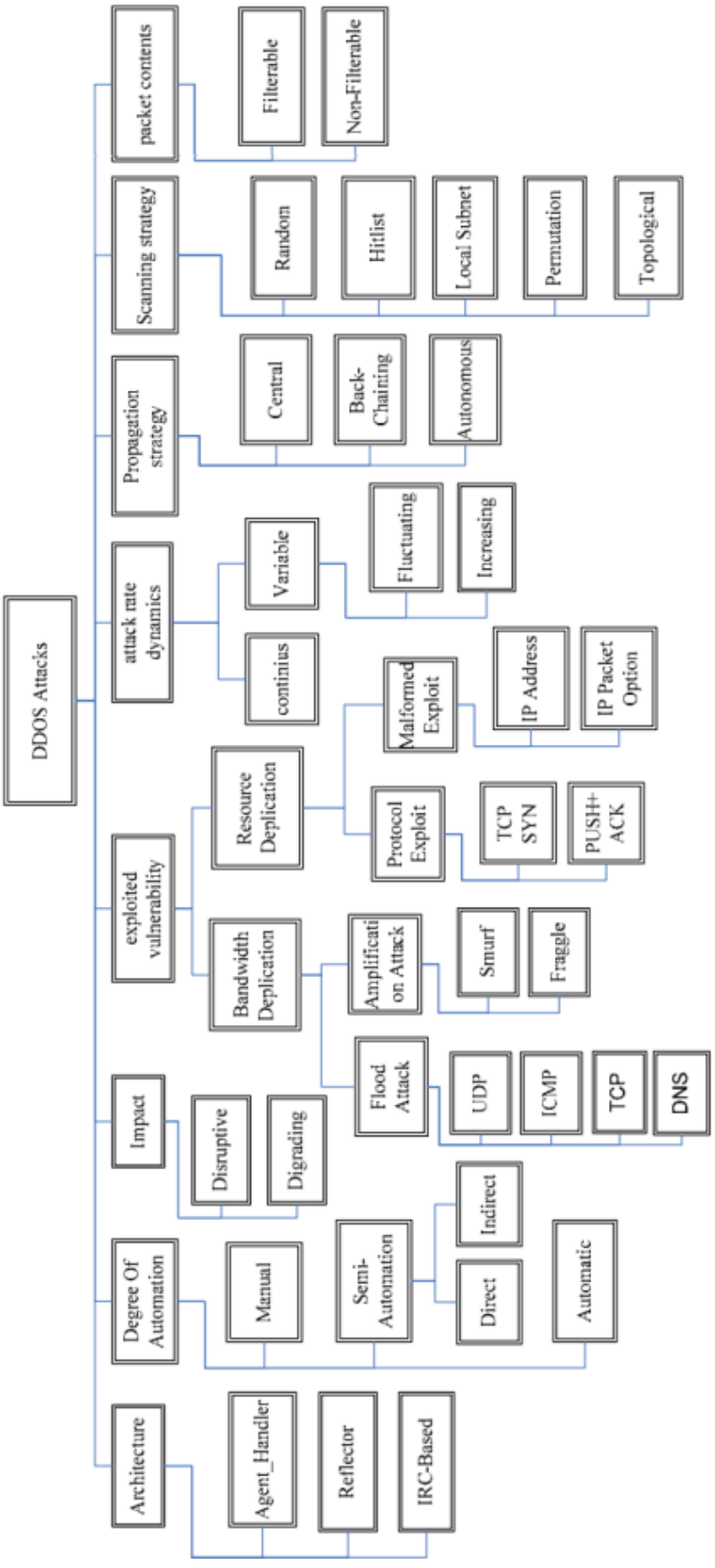

Figure 3.1. DDoS attack taxonomy [2].

fore, the attacker's client uses those software programmes to gain access and communicate with the agents.

- Agent (daemon): The agents are compromised hosts which are running a special program and responsible for generating a stream of packets towards the intended victim. Those machines are commonly external to the victim's own network, to avoid efficient response from the victim, and external to the network of the attacker also, to avoid liability if the attack is traced back. The agent software exists in compromised systems that will eventually carry out the attack.

The attacker may communicate with his handlers (one or more handlers can be possibly exist) to identify which agents are up and running, to schedule attacks, or when to upgrade agents. The administrator or the owner of the agent systems has no idea that they are manipulated for a bad purpose by another system.

Trinoo is an agent-based attack tool that can be used to generate UDP flood attacks against an IP address (one or more is possible). In the trinoo, constant-size UDP packets are used to send to target random ports on the victim machine, furthermore, some versions of trinoo supports IP source address spoofing. Trinoo works is so simple. The trinoo agent gets installed on a system, and abusing some system bugs the attacker can remotely compile and run the agent installation within the secondary victim's system buffer. The handler uses UDP or TCP to communicate with the agents and also this communication channel can be encrypted and password protected as well.

(ii) IRC Based Attack: In the IRC-based DDoS attack model architecture, an IRC (Internet Relay Chat) communication channel is used to connect the client to the agents. Main participants are same with the agent handler model attack, such as client, handler and agent. An attacker can use "legitimate" IRC ports for sending commands to the agents in this type of attack [7]. Due to the fact that, the attacker uses the legitimate ports, tracking the DDoS packets more difficult. In addition, because of it's normal execution, IRC servers can handle and generally have large volumes of traffic. This is also a big benefit for the attacker to hide its existence. Another important advantage of this model is

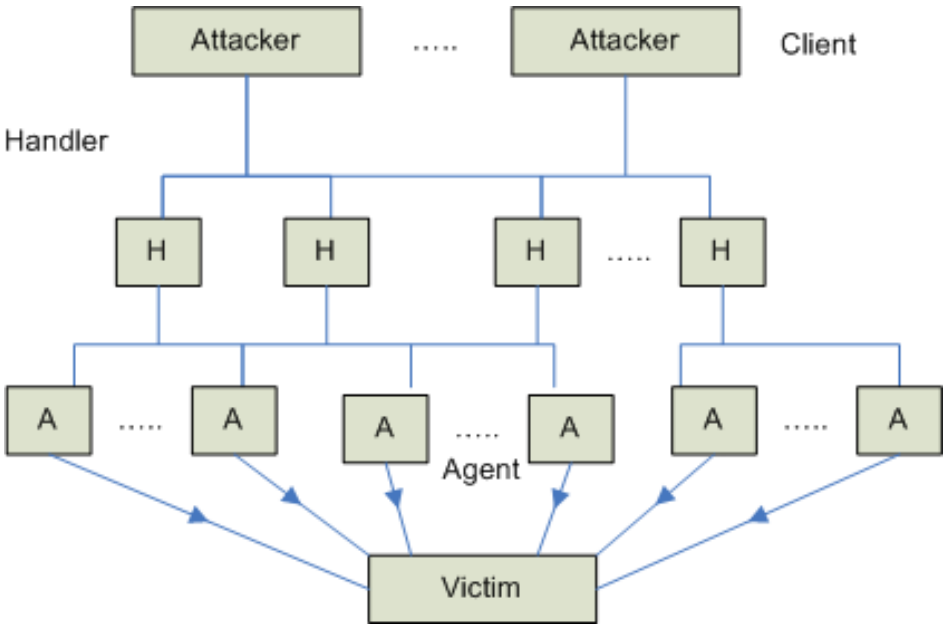

Figure 3.2. Trinoo.

that the attacker does not need to maintain the list of the agents, since he can log on to the IRC server and see a list of all available agents [14]. The agent software installed in the IRC network usually communicates to the IRC channel and therefore, warns the attacker if a new agent is up and running.

IRC-based DDoS attack were developed after the agent–handler attack. This has as a result many IRC-based tools to be more sophisticated, powerful and intelligent as they include some important features that can be found in many agent–handler attack tools.

Knight is an IRC-based DDoS attack tool very lightweight and powerful that was first reported in July 2001 [15]. SYN attacks, UDP Flood attacks, and an urgent pointer flooder can be done via knight IRC attack tool [16]. This tool is designed to run on windows operating systems. Some features such as an automatic updater via HTTP or ftp, a checksum generator and more are available on this tool. The Knight tool is generally installed simply. For example, using Trojan horse program called Back Orifice cause knight tool installed on the local host. Another IRC based DDoS tool is Kaiten [17]. The important features of Kaiten is UDP, TCP flood attacks, SYN and PUSH+ACK attacks and randomizes the 32 bits of its source address.

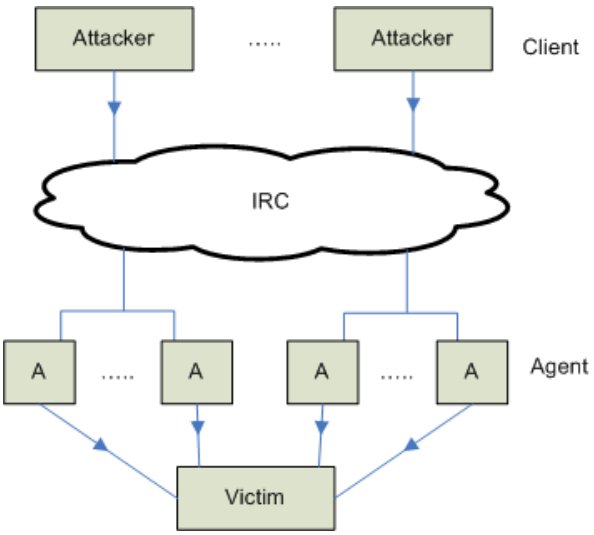

Figure 3.3. IRC based attack model.

(iii) Reflector Model: The main components of the reflector model are:

- Attacker
- Handler
- Agent
- Reflector

The scenario of this type of attack can be explained like in the following. The attackers have control over the handlers, and the handlers have control over the agents. The controlled agents send flood packets which includes the victim's IP address as the source IP address to other uninfected machines. The infected machines are known as reflectors, exhorting this type of machines to establish a connection with the real victim. A reflector can be a web server that responds to TCP SYN requests with a SYN-ACK reply, or any host that responds to ICMP echo requests with ICMP echo replies. Reflectors can also be used as amplifiers by sending packets to the broadcast address on the reflector network, soliciting a response from every host on the LAN [18, 19]. A set of predetermined reflectors are needed if an attacker wants to perform this type of attack. The reflectors could also be distributed on the Internet, because the attacker does not need to install any agent software. Due to the fact that, the reflected packets are normal packets with legitimate source addresses, they cannot be filtered based

on route-based mechanisms.

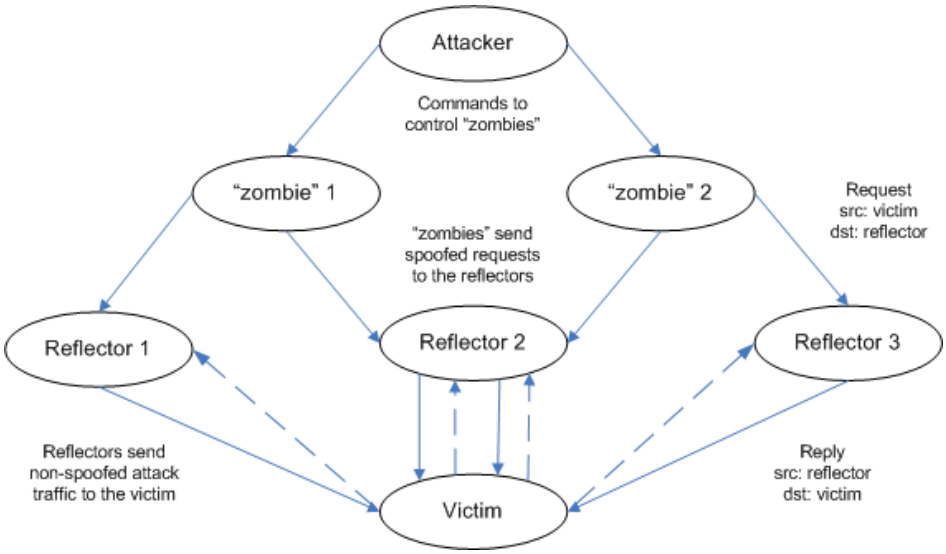

Figure 3.4. Detecting reflector attacks.

3.1.3.2. Degree of Automation Base. Based on the degree of automation of the attack DDoS attacks can be classified into 3 main groups:

- Manual DDoS attacks
- Semiautomatic DDoS attacks
- Automatic DDoS attacks

In fact, all types of DDoS attacks were manual in the beginning, later then many of attacks get being automated.

(i) Manual: The scanning of remote handler and agent machines for vulnerabilities, controlling and intruding them, putting the attack code into them are done manually.

(ii) Semi-Automatic: The trespasser puts the automated scripts to find and expose the target machines for facility of the attack code. The handler machines will be employed to specify the attack type, the victim's address and then order the inception of the attack to agent who is going to send packets to the victim.

If the agent and handler machines need to know each other's identity in order to communicate, this is called direct communication. This is achieved by hard-coding the IP address of the handler machines in the attack code that is later installed on the agent. On the other hand, if an attacker controls the agents using IRC communications channels and therefore, determination of a single agent may lead no further than the identification of IRC server channel is an indirect communication.

There is a drawback exists in this type of approach. The discovery of one compromised machine can expose the whole DDoS network and attacks with indirect communication use indirection in order to achieve a greater survivability of DoS attacks.

(iii) Automatic: There is no communication is done between attacker and agent machines. The attack features are pre-programmed in the planted code in advance. The attack type, the duration and the victim's address can be counted as a attack feature. This way, probably the attacker's identity cannot to be conceived by others.

There is a drawback exists in this type of approach. The propagation mechanisms usually leave the backdoor to the compromised machine open, making possible future access and modification of the attack code.

(iv) Disruptive: In this type of attack, the entire of the bandwidth will be interrupted, because of this features, this type of attack also known as disorderly attack.

(v) Degrading: In this type of attack, the entire of the bandwidth will not be interrupted, it causes partial bandwidth consumption. Therefore, this type of attack is called "degrading attack". It is hard to detect this attack, because of slowly interrupting the bandwidth. Actually, the main purpose of degrading attacks is instead of making the whole service unavailable totally, consuming some portion of the victim's resources. The consequence of this attack is not only the delay of the detection but also an immense damage on the victim.

3.1.3.4. Vulnerability Base.

(i) Bandwidth Depletion Attacks: Bandwidth depletion attacks can be characterized as flood attacks and amplification attacks. There are two types of depletion attack exist:

- Flood Attacks

  In this type of attack, the agents (zombies) send large volumes of IP traffic to a victim system in order to become obstructed the bandwidth of the victim's system. The influence of these flood packets which are sent by the agents alters from slowing it down or crashing the system to saturation of the network bandwidth. UDP flood attacks and ICMP flood attacks are the most common flood attack types.

- Amplification Attacks

  In this type of attack, the broadcast IP address feature, that most routers and the other network devices use for many reasons, is abused by the attacker or the agents(zombies). In this type of DDoS attack, the attacker can send the broadcast message directly, or by the use of agents to send the broadcast message in order to increase the volume of attacking traffic. If the broadcast message is sent directly, the attacker can use the systems within the broadcast network as agents without needing to penetrate them or install any agent software. Smurf and Fraggle attacks can be counted as the most well known amplification attack types.

(ii) Resource Depletion Attacks: DDoS resource depletion attacks involve the attacker sending packets that misuse network protocol communications or are malformed. Network resources are tied up so that none are left for legitimate users.

- Protocol Exploit Attacks

  Protocol Exploit attacks take advantage of some implementation bugs of the protocols that installed at the victim device. The most well known example of protocol exploit attacks is TCP SYN attacks. TCP SYN attacks take advantage of the inherent weakness of the three-way handshake involved in the TCP connection setup. TCP/IP stack of the victim server, receiving an

initial SYN request from a client, sends back a SYN_ACK packet and waits for the client to send the final ACK (acknowledge) to complete the three-way handshake and establishing the connection. On the other hand, the attacker sends a large number of SYN packets and never acknowledges any of them. In fact, the attacker leaves the server waiting for the nonexistent ACKs [20]. Because of the server has the limited buffer queue, it cannot accept new connection requests. SYN flood results in the server being unable to process other incoming connections as the queue gets overloaded [21]. Other examples of protocol exploit attacks are PUSH_ACK attacks, CGI request attacks and the authentication server attacks.

- Malformed Packet Attacks

  A malformed packet attack is an attack which the attacker teaches the zombies to send malformed IP packets to the victim in order to crash the victim system.

  There are at least two types of malformed packet attacks. In an IP address attack type, the packet contains the same source and destination IP addresses. This cause of confusing the operating system of the victim system and can cause to crash.

  In an IP packet options attack, a malformed packet may randomize the optional fields within an IP packet and set all quality of service bits to one so that the victim system must use additional processing time to analyse the traffic. If this attack is multiplied, it can exhaust the processing ability of the victim system.

3.1.3.5. Attack Rate Dynamic Base. There are two types of attacks based on the attack rate dynamics, these are continuous and variable rate attacks.

(i) Continuous Rate: This type of attack is generating the attack with a full force and without a break. Therefore, the influence is so temporary and quick.

(ii) Variable Rate: This type of attacks as their name indicates, “vary the attack rate”. It has a rate change mechanism feature. There are two subcategories in

this type of attacks, these are increasing rate and fluctuating rate. Increasing rate attacks lead to the exhaustion of victim's resources therefore, delaying detection of the attack. On the other hand, fluctuating rate attacks have a wavy rate that is defined by the victim's behavior and victim's response to the attack, at times decreasing the rate in order to keep away from the detection.

3.1.3.6. Scanning Strategy Base.

(i) Random: During random scanning type, each agent (zombie) probes random addresses in the IP address space. This potentially creates a high traffic volume since many machines probe the same addresses. Code Red (CRv2) used this method [22].

(ii) Hitlist: In hitlist sacnning type, an agent that performs hitlist scanning, probes all addresses from an externally supplied list. When it detects the vulnerable machine, it sends one half of the initial hitlist to the recipient and keeps the other half. Therefore, it allows a fast multiplication and no collisions during the scanning phase.

(iii) Topological: Topological type of scanning uses the information on the agents to select new victims. All email worms use this method.

(iv) Permutation: In permutation scanning type , all agents share a common IP address space; each IP address is mapped to an index. An agent begins scanning by using the index computed from its IP address as a starting point. Whenever the agent sees a machine that is already infected, it chooses a new random start point.

(v) Local Subnet: In local subnet type of scanning can be added to any of the scanning to scan for targets that occupy on the same subnet as the agent.

3.1.3.7. Propagation Strategy Base.

(i) Central: In central type, the attack code is put on a central server or set of servers. After the agent machine is determined, the code is downloaded from the

central source through a file transfer mechanism (FTP).

(ii) Back-chaining: In back-chaining type, the attack code is downloaded from the machine that is used to exploit the system. The infected machine then becomes the source for the next propagations. Example of this type of attack is Ramen Worm [7].

(iii) Autonomous: In autonomous type, the file getting step by injecting attack instructions directly into the target host during the exploitation phase is abstained. Example of this type of attack is Warhol Worm [23].

3.1.3.8. Packet Content Base.

(i) Filterable: Filterable attacks use fake packets or packets for non-critical services. This type of attack can be filtered by the victim's firewall. UDP flood attack or an ICMP request flood attack on a web server can be counted as a filterable type of attack.

(ii) Non-filterable: In this type of attacks, the packets that request legitimate services from the victim are captured and used. Therefore, filtering all packets that match the attack packet pattern would cause to prevent the access of the specified service to both attackers and the real clients to the server. HTTP request flood targeting a web server or a DNS request flood targeting a name server can be counted as an example.

3.2. DDoS Timeline and DDoS Incident

DDoS Timeline [24]

- Before 1999: Point2Point (SYN flood, Ping of death, ...), first distributed attack tools ('fapi')
- 1999: more robust tools (trinoo, TFN, Stacheldraht), auto-update, added encryption
- 2000: bundled with rootkits, controlled with talk or IRC

- 2001: worms include DDoS-features (i.e. Code Red), include time synchro
- 2002: DrDos (reflected) attack tools, (179/TCP; BGP)
- 2003: Mydoom infects thousands of victims to attack SCO and Microsoft

DDoS Incidents

- February 2000 - DDoS attack caused shutdown of Yahoo, eBay and Amazon for a few hours.
- January 2001 - First major attack involving DNS servers as reflectors. The target was "Register.com".
- February 2001 - The Irish Government's Department of Finance server was hit by a DoS attack carried out as part of a student campaign from NUI Maynooth.
- May 2001 — Worm Code Red was supposed to attack White House website.
- October 2002 - Attackers performed DNS Backbone DDoS Attacks on the DNS root servers and disrupted service at 9 of the 13 root servers.
- August 2003 — Worm Blaster attacks Microsoft web pages.
- January 2004 — MyDoom attacked 1 million computers.
- February 2007 - Attackers performed a second set of DNS Backbone DDoS Attacks on the DNS root servers and caused disruptions at two of the root servers.
- February 2007 - More than 10,000 online game servers in games such as Return to Castle Wolfenstein, Halo, Counter-Strike and many others were attacked by "RUS" hacker group. The DDoS attack was made from more than a thousand computer units located in the republics of the former Soviet Union.
- April-May 2007 - A spree of DoS attacks against Estonia's prime minister, banks, and less-trafficked sites run by small schools.
- July 2008 — A DDoS attack directed at Georgian government sites containing the message: "win+love+in+Rusia" effectively overloaded and shut down multiple Georgian servers. Websites targeted included the web site of the Georgian president, Mikhail Saakashvili, rendered inoperable for 24 hours, and the National Bank of Georgia.
- March 30 - April 1, 2009 - Cloud computing provider GoGrid is hit by a "large,

distributed DDoS attack, which disrupts service to about half of its 1,000 customers."

- March 31, 2009 - A DDoS attack knocks ultra DNS offline for several hours.
- April 2-5, 2009 - Domain registrar "Register.com" is hit with a DDoS that causes several days of disruptions for its customers.
- April 6-7, 2009 - Customers of The Planet are hit by web site outages as a result of a DDoS aimed at the huge hosting company.
- June 2009 - The famous P2P site known as "The Pirate Bay" was rendered inaccessible due to a DDoS attack.
- June 2009 - Iranian election protests, foreign activists seeking to help the opposition engaged in DDoS attacks against Iran's government. The official website of the Iranian government was rendered inaccessible on several occasions. Critics claimed that the DDoS attacks also cut off Internet access for protesters inside Iran; activists countered that, while this may have been true, the attacks still hindered President Mahmoud Ahmadinejad's government enough to aid the opposition.
- July 2009 - Multiple waves of cyber attacks targeted a number of major websites in South Korea and the United States: the White House, the Department of Transportation, Federal Trade Commission, and the Department of the Treasury. Hit at the same time were the Washington Post and the New York Stock Exchange. The attacker used botnet and file update through the Internet is known to assist its spread. Investigation is still underway.
- August 6, 2009 - Several social networking sites, including Twitter, Facebook, Livejournal, and Google blogging pages were hit by DDoS attacks, apparently aimed at Georgian blogger "Cyxymu". Although Google came through with only minor set-backs, these attacks left Twitter crippled for hours and Facebook did eventually restore service although some users still experienced trouble.

3.3. DDoS Countermeasures

There are many types of DDoS attacks have been developed by attackers for years. On the other hand, new countermeasure methods are also being developed by software engineers. But still there are not enough and adequate defense mechanisms have been developed to stop all kind of attack types.

There are several reasons that why attackers can avoid prevention, detections and defenses:

- DDoS attacks generate a huge traffic volume to overwhelm the victim's network.
- It is almost impossible to separate attack packets from real client packets.
- Most DDoS attacks use spoofed IP addresses and malformed packets [25].
- It is almost impossible to do the trace back the attacker's IP address, because there are large number of attacking machines and they use spoofed source IP address
- Although the router performs an ingress filtering, a lot of spoofing packets can pass it. Because some DDoS tools provide the several spoofing levels in order to pass the ingress filtering router.
- The distributed nature of the attacks calls for a distributed response, but cooperation between administrative domains is hard to achieve [26].

DDoS countermeasures are divided into 3 subcategories [3]. These are mitigation, deterrence and prevention.

3.3.1. DDoS Mitigation

Mitigating the effects of a DDoS attack does not mean to detect of the DDoS attack. Applying some distinct policies that separate trusted portions of traffic from others can limit the impact of malicious behavior without the need for an attack detection mechanism.

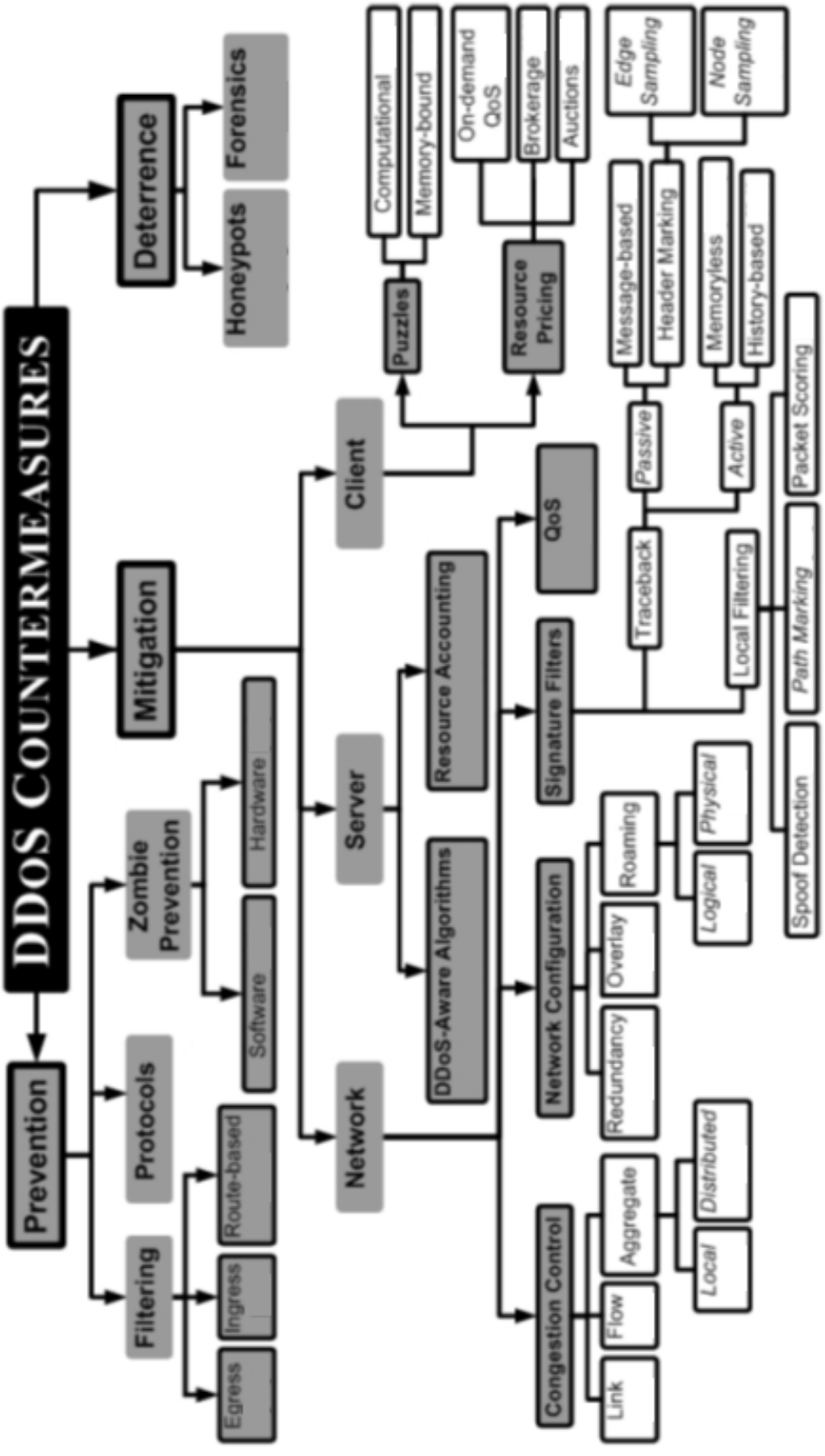

Figure 3.5. Taxonomy of DDoS countermeasures [3].

Load balancing methods may help to mitigate a DDoS attack. Network providers can increase bandwidth for critical connections to protect them from being unavailable during any possible attack. Furthermore, target system owners may add new servers in advance and provide additional failsafe mechanisms against any possible DDoS attack. Throttling is an another technique which proposes to prevent servers from going down. The max-min fair server-centric router throttle method [27] causes routers that access a server with logic to control incoming traffic to levels. Therefore, the server process will be safer and flood damage to the server can be prevented. Additionally, there is a difficulty with implementing throttling that, it is hard to decipher legitimate traffic from the malicious traffic.

Network mitigation method is divided into three subcategories:

- Network
- Server
- Client

Network

Intermediate network as well as some mitigation methods are included in network-based mitigation.

Congestion Control: Link, flow or aggregate are the types of congestion control defense mechanism. In a link-based congestion control scheme, there is a queue for each incoming packets that the router device handles. Router forwards packets by sampling the head packet from each queue and the balancing method between queues is round-robin. It is also possible to arrange packets according to their origin or destination networks. The packet flow concept actually means the classes of packets. Throttling certain flows rather than a specific ingress link can be used for routers which have overloaded traffic. This means, the routing policies do not affect a well-behaved flow sharing an ingress link with a misbehaved one. The attackers have the possibility to produce attacking flows that individually appear well-behaved, when controlling a large

number of agents.

It also depletes resources on the target's network. Defense systems can prevent those attacks by more sensible traffic classification than what is possible with flows. The resulting classes of packets are called aggregates. An aggregate is defined as a group of packets sharing a common property [14]. Network origin and destination, an application, and protocol type can be counted as this common properties. For example, refined aggregates could consider only TCP SYN packets. Distinguishing specific aggregated flow from the rest of the traffic allows more certain filtering that reduces the influence of DDoS attacks on irrelevant traffic.

Network Configuration: Redundancy, overlay and roaming are the types of network configuration DDoS mitigation techniques, those schemes protect from DDoS by modifying physical or logical configuration of any component in the network infrastructure.

Redundancy: The main purpose of redundancy is to process all incoming traffic at any time, so the increases in a server's load do not affect any of its clients. Those techniques help to protect server from the sudden increase in demand of a service. These sudden increases are sudden bursts in traffic due to a large number of content requests from real users.

Overlay: This technique requires either adding an extra layer of networking components or extending the functionality of nodes already in place. In [28] a Secure Overlay Service (SOS) is provided by an overlay network of routers which use a hash-based algorithm to route packets to a server. An outside host wishing to communicate with that server must first contact a Secure Overlay Access Point (SOAP), a designated router that lets a packet enter the overlay only after authenticating its source.

Roaming: Changing server's IP address during the ongoing attack. Especially, if the clients are coming with a URL or DNS name, it is a possible solution for protecting the server from the attack. Doing so affects legitimate clients until they perform a

new DNS lookup, but it protects the server from the malicious traffic still directed at the old IP address. If this DNS change can be connected to a detection unit such as IDS, the victim server can be protected automatically just after the detection of DDoS attack.

Signature Filters: In signature-based strategies, defense system reacts only when an attack detection mechanism flags certain packets as malicious. There are two categories for this type of countermeasures, these are local filtering and traceback mechanisms.

Local Filtering: Local filtering methods are noticable by the type of signature they use to classify traffic. When a router determines that packets with a certain signature are malicious, subsequent incoming packets with the same signature are discarded or rate-limited.

IP Traceback: Traceback mechanisms focuses on localizing the origin of a stream of attacking packets, because knowing the attacker's location allows activation of filters closer to the source of the attack. This reduces the impact of consequence damage and false positive alarms. Active traceback mechanisms recursively query upstream routers to obtain information about a certain dangereous flood packets. With passive mechanisms, the intermediate network automatically sends path information to the victim. In both cases, the victim collects the partial or complete path information to construct the sequence of routers used by a poisonous stream.

QOS: The attacks targeting depletion of network bandwidth can be mitigated by applying service differentiation mechanism that reserves a share of the bandwidth to certain categories of traffic. Such a mechanism creates different classes of packets that are each treated differently by the network, for example higher priority classes are forwarded first or the lowest priority packet can be ignored in case of emergency.

Server

Several strategies techniques generally obtain by developing softwares. For example, operating system such windows can periodically trace the backlogqueue of the listener ports and drop halfopen connections. By doing so, the OS prevents a TCP SYN attack from grabbing memory resources.

Lazy Receiver Processing (LRP) [29] can also help an operating system in the case of a flooding attack by avoiding certain computations on packets that end up being dropped due to overload. Some OS-level resource accounting schemes like Escort are more elaborate than simple DDoS-aware algorithms. They enforce policies which control allocation of time-multiplexed resources such as CPU time or network bandwidth.

Client

Client-centric classes of countermeasures use of the limited computational resources of client hosts in order to force them to regulate their traffic. Every client requesting access to services must commit a certain amount of resources determined by the network or server in puzzles strategies. On the other hand, different market-like schemes in which resources are available for purchase by the clients is in resource Pricing strategies.

3.3.2. DDoS Prevention

In DDoS prevention stage, it is tried to stop DDoS attacks from being launched in the first place. Of course, reactive measures protecting critical services during an ongoing attack are important, but proactive measures are needed to be implemented to prevent the occurrence of an attack in the first place. There are many types of DDoS preventation mechanisms are developed, such as filtering, protocol and agent(zombie) prevention.

Filtering

When internetworking protocols such as TCP and IP were being designed, the functionality and ability to transpor of the data without error were the big deal. The data security was not the main concern in those days. As a result, malicious parties can generate invalid information or send harmful commands without being detected by those protocols, therefore packet filtering gains an importancy. Packet contents filtering is a first step towards reducing an eventual enemy's balefulness.

In other words, filtering refers to the looking into the IP packet headers leaving a network and checking to see if they meet certain criteria. If the packets could pass the criteria, they are routed outside of the sub-network from which they originated. Otherwise, the packets will not be sent. Since DDoS attacks often use spoofed IP addresses, there is a good probability that the source addresses of DdoS attack packets will not represent the source address of a valid user on a specific sub-network. If the network administrator places a firewall in the sub-network to filter out any traffic without an originating IP address from the subnet, many DDoS packets with spoofed IP addresses will be discarded [11].

Protocols

How to design protocols that close all the possible entrance and do not propose any opportunities for DoS attackers in TCP/IP protocol enviroment. But there are still an unsolved research problems exist. However, there are some desirable properties known to prevent specific types of attacks. For example, the goal of some attacks is to deplete a server's resources by establishing a large number of fake TCP connections. That way, the TCP buffers are saturated and incoming connection requests must be ignored. A remedy to this problem involves designing stateless protocols which shift the burden of state holding from the server to the clients. SYN cookies [30] partially reach this goal by remaining stateless in the first steps of a TCP connection establishment.

Zombie Prevention

An important step towards solving the DDoS problem consists in preventing the

attacker from finding of zombie or agent computers in the first place. To achieve this goal, it is necessary to repair the weaknesses that attackers manipulate and take into control of hosts connected to the public Internet. Buffer overflow is the most important vulnerability of the agents. It can be mitigated using either software or hardware mechanisms.

3.3.3. DDoS Deterrence

Although some techniques allow the tracing back towards to the attacking hosts, seldomly a victim can be able to find out and identify the attacker also controlling those zombies. Any method changing this mode of freedom might discourage some malicious parties from engaging DDoS attacks.

Honeypots

Honeypots are computer systems placed on a network for the sole purpose of being abused by unsuspecting attackers [31]. In other words, honeypots are systems intentionally set up with limited security to be an enticement for an attacker's attack. Honeypots serve to deflect attacks from hitting the systems they are protecting as well as serving as a means of gaining information about attackers by storing a record of their activity and learning what types of attacks and software tools the attacker is using. Since a honeypot does not offer any useful service, nearly all activities detected on it are malicious. It is thus simple to use such a computer as an intrusion detection system.

In normal operating systems, malwares can be used to cover the attacker's traces. In advanced honeypot systems, the OS is encapsulated in a logging framework so all attacker activity is recorded, ignoring the attempts of the attacker to alter the audit trail. In addititon, it is also possible to track the attacker's every action can be a deterrent since the attacker may not want to show his plans.

Forensics

New developed custom-made “sniffers” and scripts, skilled programmers can manually trace the activity of malicious programmes back to the IRC channel used by the attackers for controlling the zombies [32]. Such forensic activity may eventuate to the discovery of the attacker’s legtimate identity.

4. PROPOSED SYSTEMS

DDoS attack is still a serious problem on the Internet, as it takes advantage of the lack of authenticity in the IP protocol and stateless nature of the Internet.

TCP SYN flooding attack which is one of the DDoS attacks is the most commonly used one and still dominates DDoS attacks according to the recent NANOG report [33] in 2008.

The main cause of SYN flood attack is the structure of TCP three-way handshake protocol, which enables the attacker to consume existing resources at the server, while sparing its own resources.

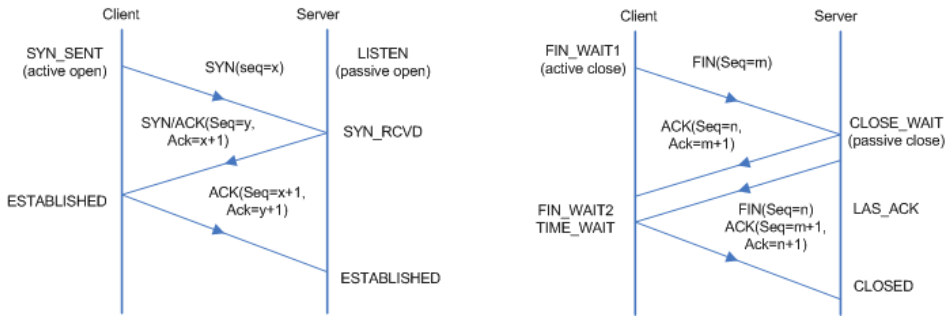

Figure 4.1. TCP connection's establishment and release.

Figure 4.1 shows the three-way handshake protocol. There are the steps of the three-way handshake protocol:

(i) A client sends a SYN packet to a server to open a connection request

(ii) The server reserves connection and put it into the backlog queue to track the TCP state on receiving a SYN packet and replies to the client with a SYN-ACK packet as a response

(iii) Finally, the client sends an ACK back to the server as an acknowledgement, and the connection comes to the established state.

New Proposed SYN Flood Tool

To generate SYN flood attacks, the ordinary attackers need agents and handlers to make the huge SYN packet volumes. In addition, they need these agents to create the attacks with different source IP addresses. On the other hand, we proposed a new attack tool that generates approximately 100 connection/second with just one regular client computer. The source IP addresses of those connections can be chosen randomly. If you run the tool in a powerful windows server, generated traffic might be 1000 connections/second.

Custom SYN Flood Attack Tool Model Architecture and Algorithm

These are the steps to create a SYN flood:

(i) Creating a SYN packet
(ii) Sending request to the server
(iii) Multithread processing, generating flood

To creating a SYN flood, we must generate the whole packet from the beginning. First, we should create a new TCP packet. There are many parameters and calculations are needed to create a new packet. Respectively, ethernet, TCP and IP header lengths, source and destination mac addresses, ethernet protocol type, source and destination IP addresses, IP protocol type, source and destination ports, sequence numbers, fragmentation offsets, and windows size are the parameters that we defined. After these parameters are set in the TCP raw packet, TCP and IP checksum functions are done and also checksum values are added to the packet. The most important thing is setting the SYN flag bit in the packet. When the raw packet is ready, it is time to send the SYN to the server application. To create a flood, the client should be multithread. After many code developments, it is possible to make a multithread client with appropriate loops. The conclusion is like just one computer (or server) and just one client programme running on the machine, but like thousands of clients are running on that machine at the same time. They are all trying to send SYN packet to

the particular victim server application. They can easily create a flood.

This client tool (the attacker tool) does not need any handler or agent for helping him to create a flood or accessing the server. However, it is also possible to distribute the programme to the agents and create much bigger traffic volume at the same time.

The most important features of the client tool are summarized below:

- It is possible to create a huge volume just using one client
- It is possible to create a packet with different source IP addresses
- İt is possible to create a packet with the source IP address of vulnerable computer that runs on the same LAN with the victim

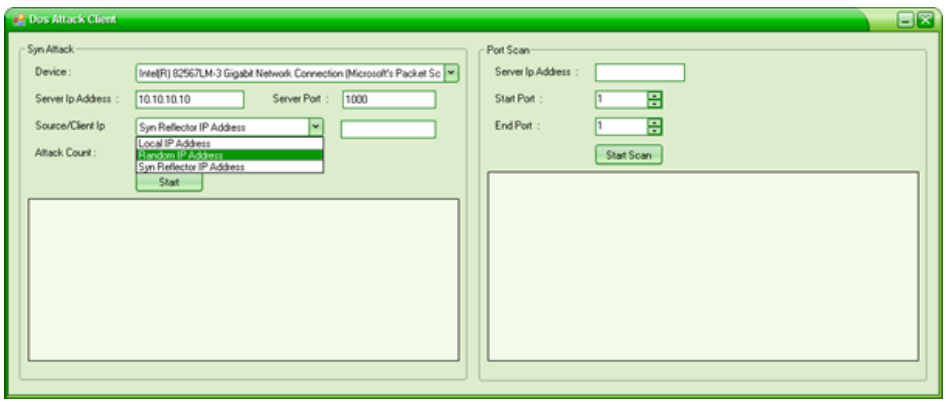

Figure 4.2. New proposed client tool.

The main differences and also the main advantages of the client tool are:

- It can generate huge volume of traffic
- It is possible to generate different kind of attack mechanisms (FIN, PUSH_ACK) with the same structure
- It is possible to send the packets using local IP, fake IP,IP address of a trusted client
- It is an integrated application, you can scan the server side ports first, then start different kind of DDoS attacks from the application window.

New Proposed Virtual Server

Nowadays, virtual server structures are used to protect real servers from the half open TCP connections generally. Sometimes, they are used to sharing the connection load between many real servers.

On the other hand, we create a new virtual server model, which is really different from the regular servers. The feature of the server tool can be divided into three sub categories:

- It accepts new connection requests. After three way handshake completed and be sure about the confidence of the client, it forwards the connections to the real server.
- It has a decision mechanism. According to this mechanism, it can decide about the client, and deduce the client is reliable or not.

Custom Virtual Server Model Architecture and Algorithm

Generaly, custom virtual server works like a server in terms of functionality. It has a port number and IP address. It issues regular socket api functions like listen, bind, accept etc. The virtual server program is a socket application program and the socket type of the server which is also called the listener is a stream socket.

To run the virtual server, we need the real server information (IP address and port number), the virtaul server service port number and the threshold value.

When a virtual server starts, it opens a new port on the system. When a client wants to connect to that port, desicion mechanism comes to the stage. Virtual server never accepts the connection, before it gets positive feedback from the decision side. Assume that, desicion mechanism sends positive reponse to the virtual server, then it accepts the connection and forward it to the real server. Thus, the mission is completed without a problem. On the other hand, if it gets negative response from the desicion

algorithm then it rejects the client and never forwards that connection to the real server.

The main purpose of the decision algorithm is to understand the reliability of the client.

It is composed of the following components:

- Historical database: Trusted client information is recorded here
- Instant connection threshold value: To decide at the first sight for group of connection
- Data mining: To calculate historical trusted threshold values for new connections

The system components provide the virtual server to decide both an instant connection rate and historical values. It makes the virtual server more reliable, and helps to protect the real server itself.

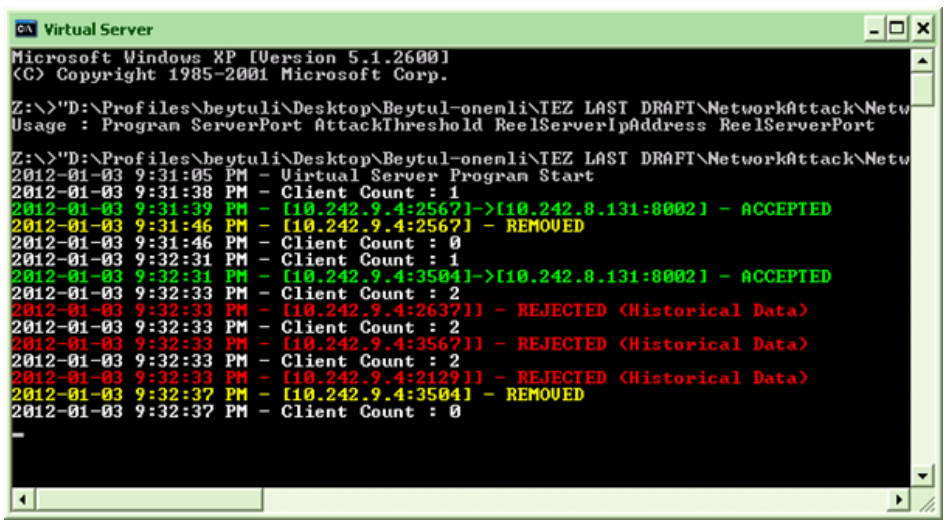

Figure 4.3. New proposed virtual server.

New Proposed IDS Tool

During SYN flood attacks, an attacker or in other words the client generates

a large number of SYN requests but never sends the ACK packets to complete the connections. The victim server's backlog queue can be easily exhausted and all the new incoming SYN requests are not accepted and dropped. Furthermore, other system resources like network bandwidth are occupied.

Among various SYN flood defense mechanisms, victim modifications, like SYN cookies, is used for mitigating SYN floods. However, SYN cookies are not able to encode all TCP options and TCP flags. Furthermore, they are only trying to solve the asymmetry feature in TCP protocol.

Recently, some SYN flood detections [34,35], are putting more attention in current literatures and may solve the bandwidth consumption problem.

These recent methods benefit from the relationship between the TCP flags in the control packets during the connection establishment and release, like SYN, SYN/ACK, ACK, FIN,and RST, and their ideas are better than the SYN cookies method.

On the other hand, they all have an important deficiency that the attacker can avoide the detection by spoofing the control packets. Using SYN-FIN(RST) pair attack is widely used [34–36] but the attacker can easily spoof the FIN(RST) packet. In fact, the sequence number in the FIN or RST packet has almost no relationship with the SYN packet. Using SYN-ACK pair attack is also used [37] but has the same problem. The attacker can easily spoof the third packet which is the ACK packet.

We propose a more accurate SYN flood detection scheme which has an improvement the SYN-FIN(RST) pair's behaviour. As shown in Figure 4.2, it has only one SYN packet and one FIN packet for each normal TCP connection. During the SYN flood attack, since the SYN packets have no corresponding FIN packets.

We record to the database many parameters about TCP connection such as, client and server's IP addresses and ports, initial sequence numbers, SYN and FIN flag arrives and TCP connection start time. In this case, it is impossible for an attacker to

generate the spoofed FIN packets to discard the detection.

Custom IDS System Model Architecture and Algorithm

We developed custom sniffer tool that listens all the network traffic of the server's ethernet card. IDS tool is filtering this network traffic according to the inputs that given by the user. It also records the specific connection's parameters to the SQL database.

Depends on what the application does, the server response time can be changed.

In our custom IDS system model, NRTT and server response time value is counted as 1 second. It is expected that, within one second, the FIN packet should come from the client (the client already sends the SYN packet) with the expected parameters. If there is just a SYN packet arrives and no FIN packet comes with in a second, it means this connection is an open connection. The IDS programme, controls the connections in the database every second and warns the user about the opened connections.

The IDS tool has another feature which makes a huge differences between other IDS tools, which is the ability of tracing network traffic. It composes a base line according to the previous network traffic of the server.

To do this, the sniffer tool is used again, the client and also the server port and IP address are noted to the database. In addition, the number of connection requests and the start time of them are also noted. IDS tool is calculating a base line not only for specific port number and specific applications but also general traffic and all server ports. If the current traffic of the server exdeeds the base line by 20%, the user is warned about this situation also. The last feature of the IDS tool is monitoring connection count in real time. Not only server bases but also per port bases graphics are included in the IDS tool.

Figure 4.4. Realtime graph.

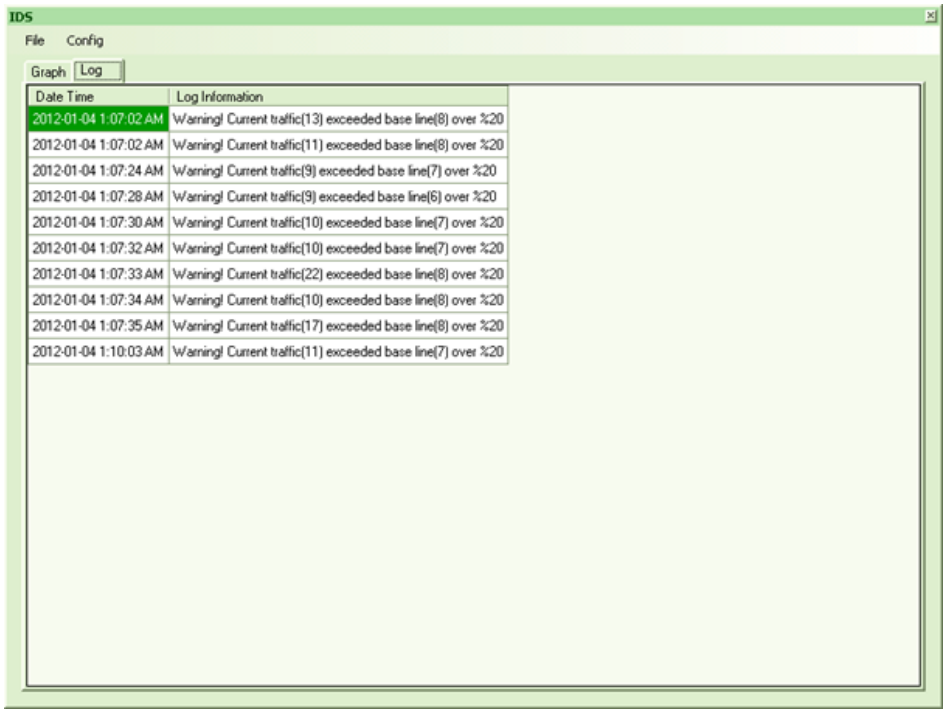

Figure 4.5. Realtime logs.

5. SYSTEM IMPLEMENTATIONS AND SIMULATIONS

5.1. DDoS Attack Tool and Features

The DDoS attack client tool is developed to be able to create a DDoS attack. DDoS client tool has four different features. These features are port scanning, SYN packet flood (using Local Source IP), SYN packet flood (with manipulated fake source IP), SYN reflector. The DDoS attack client is an socket application, therefore runs on TCP protocol application layer. Socket API and raw socket services are used for generating a single SYN packet. On the other hand, Socket API and also stream type of socket service are used for port scanning feature. When we are generating a raw SYN packet, we must form the whole headers including TCP, IP and ethernet.

Almost every parameter within the headers such as TCP checksum, device IP address, device mac-address, flags, sequence number etc. are playing vital role on managing to send the packet and successfully received by the destination. Because, when we create a packet with wrong or inadequate information, this packet is not received by the destination device's TCP stack, it is ignored without a reply. Furthermore, if the victim (in order words the destination device) is not on the same LAN, we should to send the packet to the default gateway. Therefore, particular NETSTAT command outputs also are needed to be processed by the SYN attack tool while generating the SYN packet, because we will need the mac-address of the gateway device. For stream socket service, there is not this kind of problem, because in stream type of socket, we only need the socket functions, packet generating process is done automatically by done TCP/IP stack of the local device. How many SYN client will join the attack should be predetermined by the attacker. This value is parametrical and given by the user. The clients are running at the same time simultaneously, on the attacker device, since the tool has the multithread working ability.

The features of the tool is explained in detailed below.

5.1.1. Port Scanning

Port scanning feature is used for checking the port status of the destination (victim) server machine. The input values are victim server IP address and port number. The client tries to connect this IP address and port. In conclusion, it lets us know the port or the services is available or not.

The main purpose of this feature is trying to understand and making forensic which service is available on the server before performing the real attack.

There are some inputs should be given to the port, these are:

- The IP address of the destination server
- Maximum and minimum port numbers of the destination server. In other words, the port range that will be traced by the client.

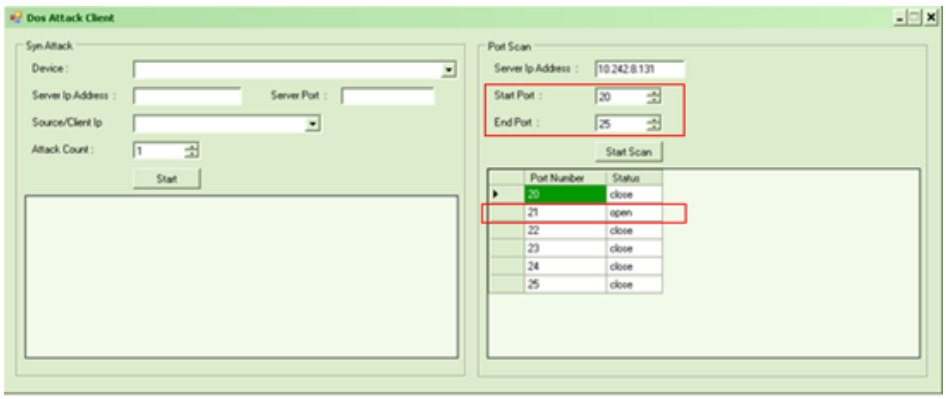

Figure 5.1. DDoS attack client.

As you see from the output log of the scanner tool, the only listener port is which is available is “21” with in the 21-25 port range. Now, we can check this via netstat command in the server side.

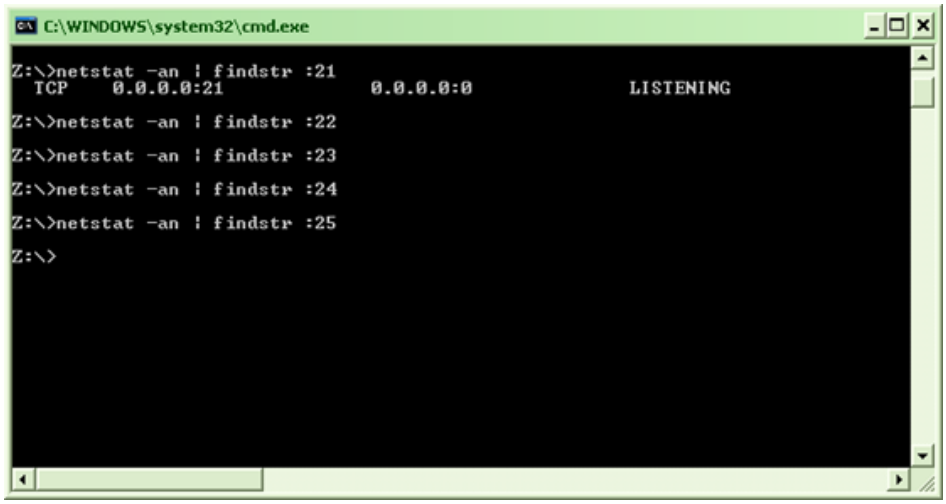

Figure 5.2. Dos attack client (windows command prompt display).

5.1.2. Other Features

SYN attack tool has three different versions, but the common work of this tool is sending one or more SYN packets to the destination (victim) server. The required input values of the tool are:

- Ethernet card of the local machine (there may be more than one ethernet card available)
- The IP address of the victim server
- The port number of the victim server
- Attack count

Generally this tool is worked for sending amount of attack count value of SYN packets from local IP address of the device to the destination victim device. Victim device sends SYN_ACK reply packets to the coming SYN packets but never gets ACK response back from the source attacker device.

Due to the fact that, victim server device does not get ACK reponse from the attacker part, it needs to retransmit to the SYN_ACK packet for many times (it depends

on the TCP/IP preferences of the device but at least 3 times).

This situation causes victim machine to consume resource and also create network traffic. In addition, if the attacker is strong enough and has enough resource to create a huge network traffic, this SYN attack action may cause victim server machine to be unreachable and unresponsive.

Under SYN attack, some victim (server) devices might be accesible but their session tables will be full shortly because of retransmission packets and endless SYN packets.

There are three different SYN attack client tool versions. The difference between them is the source IP addresses of the SYN packets. These are the varieties of the client tool:

- Local IP Address
- Random IP Address
- SYN Reflector

In Local IP address type, the source IP address of the SYN packet is local IP address of the attacker device. This method can be used while the attacker is pretty sure that the victim has not got any opportunity to traceback the attacker IP address.

In the following example, the local machine IP address is used. First, the attacker sends SYN packet to the victim, then the target sends SYN_ACK back to the attacker but never gets ACK response back and never establishes a connection with the attacker

In many network data centers there are devices for tracing malicious clients and preventing them to access the network again. In the first feature of SYN attack tool, after the first attack performs, prevention mechanisms never let to access this attacker machine again because of its known source IP address. This machine remains useless anymore, to specific victim networks.

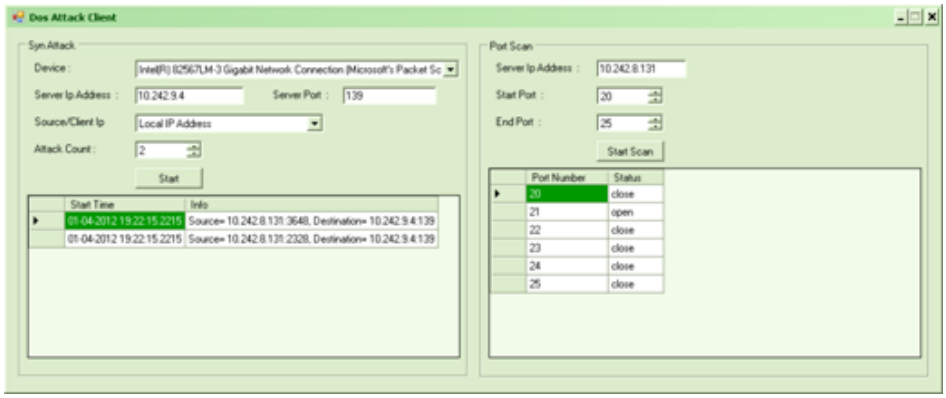

Figure 5.3. DDoS attack client.

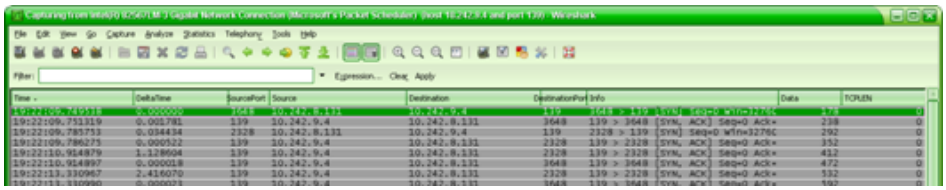

Figure 5.4. DDoS attack client (wireshark logs).

In order to overcome this problem, we improved client to generate SYN packets with fake source IP addresses.

In this type of client tool, we choose a range that the tool picks a random IP address in this pool, and it is almost impossible to reuse the same IP address again. Therefore, it will be impossible for the victim to prevent the access of the specific source IP address. For example, access lists based on IP address is useless under this kind of attack mechanism.

SYN reflector is a little bit different from other features of the SYN client tool. Firstly, we need to select thirth device that locates in the same LAN with the victim machine. After the selection, we generate SYN flood with the source IP of this selected device. Because of we pretend to send this packet from the thirth device, all SYN_ACK packets will be sent to this machine, not to the attacker. Because of the victim machines shares the same network with the server (victim) device, high probability there is no

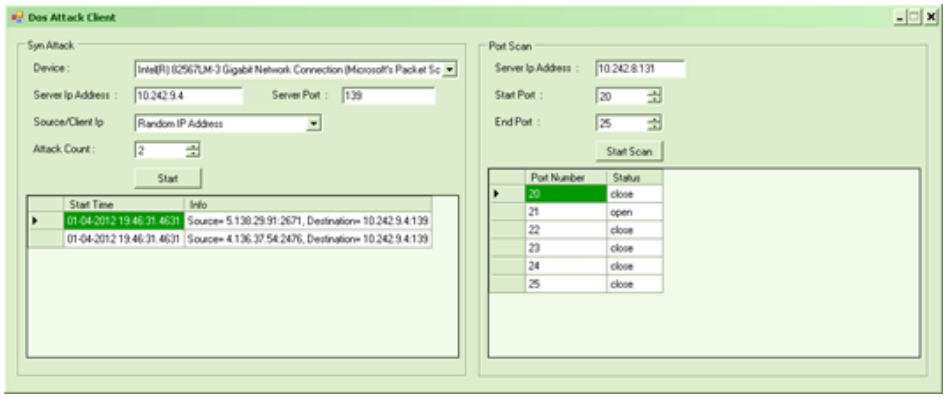

Figure 5.5. DDoS attack client.

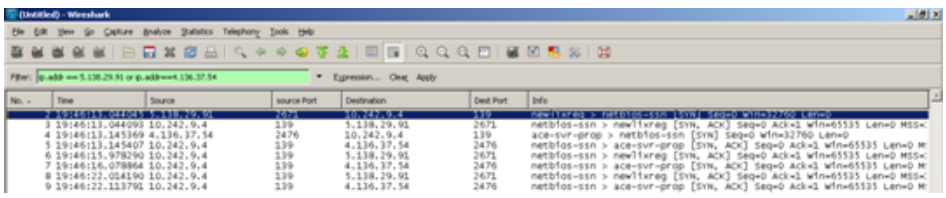

Figure 5.6. DDoS attack client (wireshark logs).

Firewall or IDS and IPS mechanisms between these devices. Therefore, this kind of attack might be more dangerous than the others.

In this example, the reflector device is chosen on the same LAN with the victim machine.

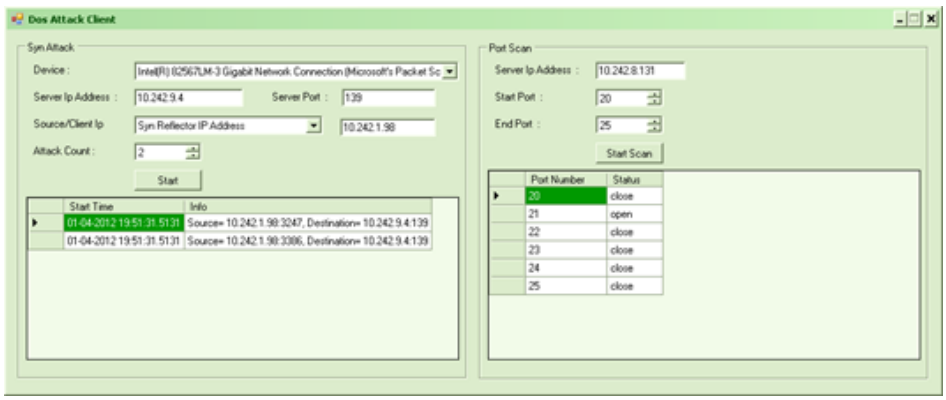

Figure 5.7. DDoS attack client.

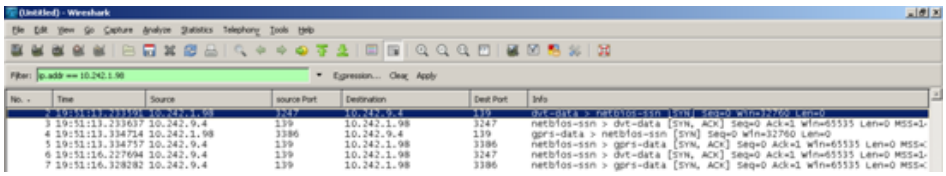

Figure 5.8. DDoS attack client (wireshark logs).

5.2. DDoS Counter Measures

There are two types of counter measure meachanisms are examined in this project. These are:

(i) IDS Mechanims
(ii) IPS Mechanisms

These mechanisms work for mitigation, deterrence and prevetion for SYN attacks. Basically, these kind of prevention tools are running on the server side. We will discuss them more detailed in this section.

5.2.1. Intrusion Detection Systems (IDS)

The main purpose of IDS tool is to prophesy the possible attacks and give notice to the user and administrator about them. The tool is composed of the following components:

- A listener port that runs on promiscuous mode
- Connection recorder
- Database system
- Data mining algorithm
- Decision mechanism

The tool works as follows, firstly we need a port runs in promiscuous mode to listen all the traffic that flows on the Ethernet card on the server device. All the network traffic is recorded to the database with packet details. The packet details are client IP address, packet sequence number, flag information (SYN, SYN_ACK, FIN etc.) they are all recorded to the database.

There is a decision mechanisms to decide which client is malicious and which is not.

The process works as follows, when a client first comes to the server, it sends a SYN packet which catches by the tool and recorded to the database. Furthermore, based on the server application type, tool is waiting for particular time for the termination of this connection. If the connection does not terminate within the particular predefined period, this connection is considered as an open connection, and the user is warned about it.

Another feature of the IDS tool is building a base line according to the previous network traffic of the server applications. It is also possible to calculate a base line for specific port numbers and specific applications. If current traffic of the server exceeds the base line by 20%, the user is warned about this situation also.

The calculation of the base line can be explained as follows. The calculation is done for the last seven days. The connection count is calculated for the time period the connection time periods, there is a milisecond time grouping exists here. If for a moment there is no new connection comes to that server port, then this moment is not added to the total or any average calculations. In conclusion, there is an average value calculation is done including the moments that at least one connection comes to the server.

After this baseline determined, the graph and the log file are starting to be generated. The red line indicates the baseline of the server including whole ports that runs on the server side. In addition, it is also possible to pick a specific port and focusing on the connections and baseline of it. When the current connection count exceeds the baseline threshold value by the rate of twenty percent, the users are warned about this situation through the log file. Information of the number of open connections on the server is given to the users via log file. To decide about the connection is open or not, the tool is checking the flag status and the connection start time in the database. If the predefined time limit which is one minute for the example is exceeded and there is not any FIN or RST packet comes from the client yet, it is counted as an open connection. Proposed method is shown below:

The last feature of the IDS tool is monitoring connection counts in real time. Not only server bases but also per port bases graphics are available in the IDS tool.

In Figure 5.11, the connections of the whole server is monitoring in real time. It is possible to determine a specific port and trace the connection count of it. The method is explained below:

If you only want to trace a specific port, you can give this port number via configuration tab as an input. Therefore, after starting the tool, it only listens the traffic of that port number on the server. On the other hand, you can leave blank the port selection text box empty, and the tool starts to listen all the traffic of the server. In fact, you can filter any port after the trace starts, with the drop down list control

WHILE the promiscuous mode socket service is opened
IF a new connection arrives
Parse the incoming packet and get source IP address, destination IP address
Source port, destination port, flag information
Store the packet information into the database
Wait for the new packets to come about the related connection
IF a new packet arrives for the connection which is already stored
into the database
IF the FIN or RST flag is set
IF the elapsed time value is less than one minute
Update FIN flag information in the database for the
related connection
ELSE
Warn the user about the open connection through the
log file
END IF
END IF
END IF
END IF
END WHILE

Figure 5.9. IDS work flow.

WHILE the promiscuous mode socket service is opened
IF a new connection arrives
Parse the incoming packet and get source IP address, destination
IP address, source port, destination port, connection count
Store the packet information to the database with the connection time value
IF one second is passed
Check the database
Calculate a historical baseline (how many connection exists
on which port)
Calculate an instant value (how many connection exists on
which port)
Draw a new baseline
Draw a real time graphic for the server ports
END IF
END IF
END WHILE

Figure 5.10. IDS online monitoring work flow.

on the top right side of the page.

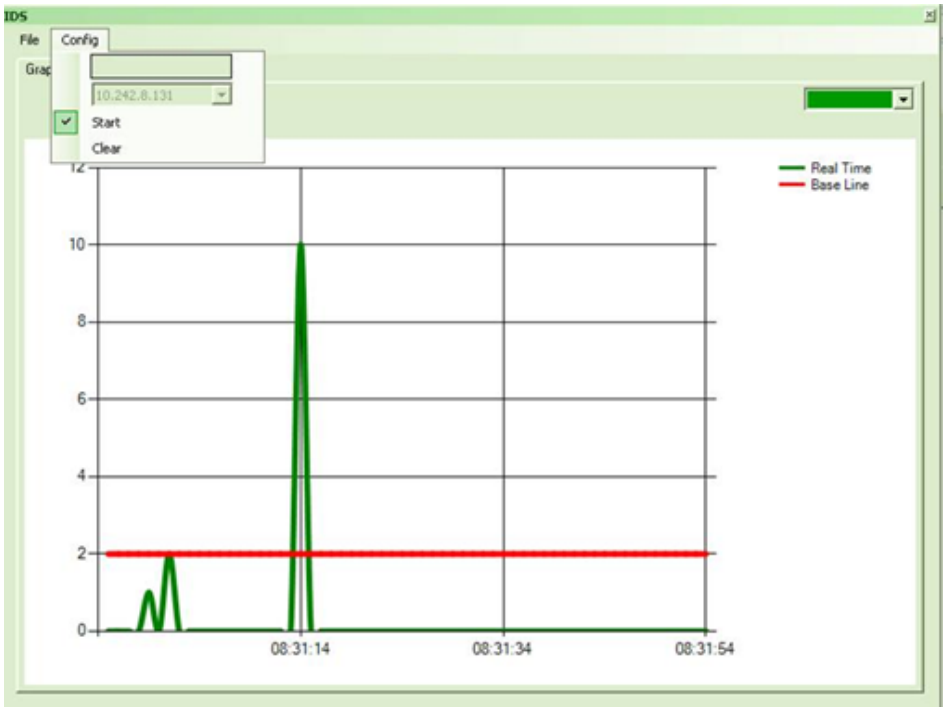

Figure 5.11. IDS tool (real time connection monitoring feature).

5.2.2. Intrusion Prevention Systems (IPS)

There are 3 types of SYN attack IPS mechanisms are discussed and implemented in this chapter. The mechanisms are:

(i) SYN Proxy
(ii) SYN Killer Transparent Gateway
(iii) SYN Killer Evil IP Address

These IPS mechanisms help to the server (victim) device to prevent the SYN attacks in real time. These mechanisms reduce the vulnerability of the server against to the SYN attacks.

Although there are three different types of IPS mechanisms are deployed to the tool, the required inputs of the programmes are in common. The inputs are:

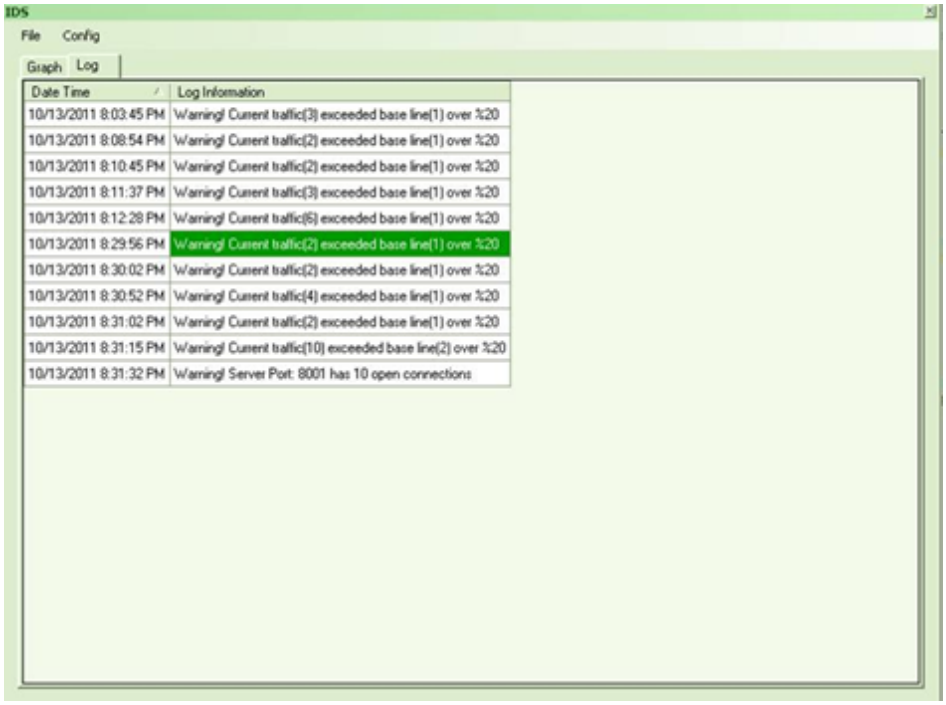

Figure 5.12. IDS tool (user warning feature).

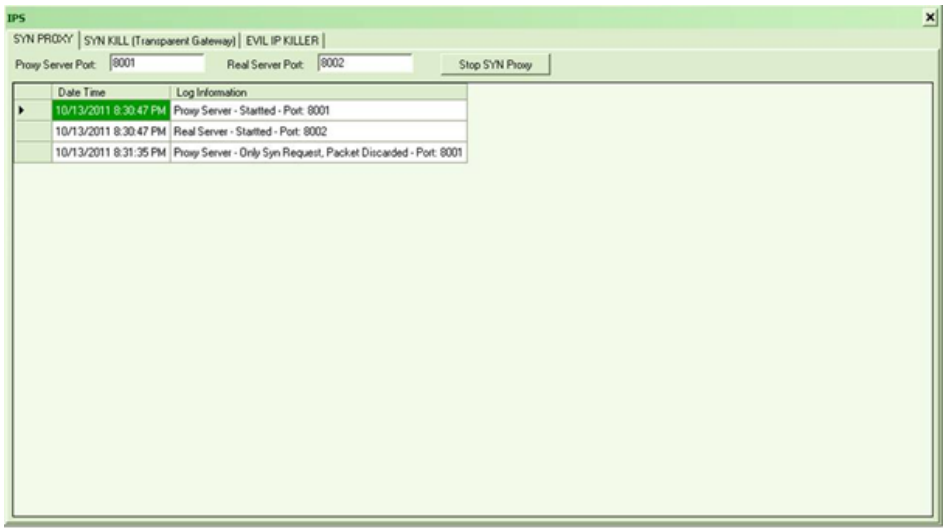

Figure 5.13. IPS tool (general view).

Proxy Server Port: Proxy server port is used for welcoming the client request firstly. It decides the client is malicious or not.

Real Server Port: It runs after the proxy server. If the proxy server believes and decide that the client is trustable, then it forwards the connection request to the real server.

Interface Card: It is possible have more than one Ethernet card for one server. In that case, this is important to choose the right interface card.

The IPS tool opens an promiscuous port on the ethernet card and listens all the traffic that going on the card. The software application is designed and coded as a stream socket at the TCP/IP application layer. When it is necessary to send a SYN or RST packet, it generates this packets like the other programmes which explained in previous parts.

The most important thing while generating this packet is the sequence numbers. While generating SYN packet, it is possible to give a random value to the sequence number of the SYN packet, because SYN packet is the first packet of the a connection, and not calculated from any other packets. On the other hand, sending RST packet or ACK packet is a little bit different from this. To be able to send a RST packet truly, it is important to calculate the sequence number of the packet like it is a real RST packet coming from the client side. To manage this, all sequence number of the specific packets are recorded to the database, and the sequence number of the next RST packet is calculated from them.

5.2.2.1. SYN Proxy. SYN Proxy mode feature is working like an additional layer between the target and the attacker or the real client. The client who is only sends TCP SYN packet, can not be considered as it establishes a connection, when the TCP three-way handshake is completed between two sides, then it can be counted as a real connection. Due to the fact that, SYN proxy mechanisms can be used.

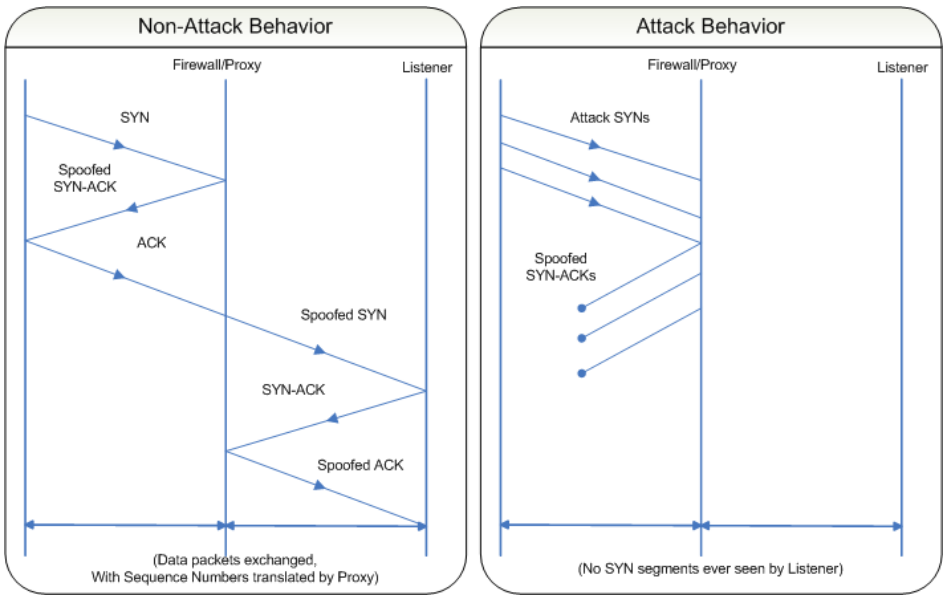

Figure 5.14. SYN proxy work flow.

In this mechanism, proxy device takes the SYN packet over itself and not let the real server know about this connection request until three-way handshake completes with the client. The proxy tool sends SYN_ACK packet to the client, If the client does not send acknowledge packet back to the proxy tool with in a specific time period (calculated considered the retransmission mechanism), this client and the SYN packet are discarded by the proxy server. On the other hand, if the client sends ACK packet back to the proxy server, it forwards this request to the real server without cutting the data streaming between them.

The session table and the backlog queue of the real server is protected by this feature. Furthermore, indirectly it affects the server devices availability and accessibility.

The general view of the SYN proxy mode IPS tool is above, the input values are virtual server and real server port numbers. On the other hand the output values are, logs which are the real time events and the specific time value that this event occurs. If the proxy server accepts a new connection requests or discard a client because of it does not meet the reliable client requirements etc., the events are logged and shown to

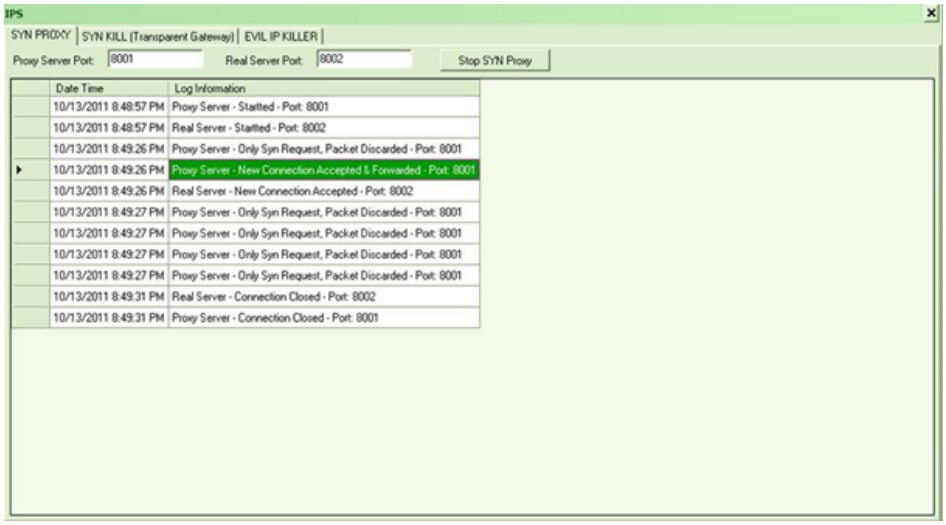

Figure 5.15. SYN proxy tool.

the users. The method is explained below:

5.2.2.2. SYN Killer (Transparent Gateway). IPS SYN killer transparent gateway mode is also good way to preventing SYN flood attacks. In this type of IPS mechanism, the SYN killer device is also working between the client and the server.

When a SYN_ACK packet is seemed by the SYN killer device, it sends ACK packet back to the server machine like the real client sends this packet. Therefore, the connection state in the server (victim or target) device is "Established" and ready for application to take the control of the connection. On the other hand, it also prevents unnecessary SYN_ACK packet retransmission network traffic that possibly generated by the server.

SYN killer algorithm checks, if there is a real ACK packet coming from the client within a specific time period. In reality, this time period can be caused because of network round trip time or a problem in the client side. If there is no ACK packet coming from the client, it means this client is malicious and SYN killer sends RST packet to the real server like client sends. The server, who gets the RST packet from the client, releases all the resource that related with the connection and drops it. SYN

WHILE the proxy port is opened
IF a new connection arrives
Store all the packet information into the database and send SYN_ACK
packet to the client (pretend to be the server)
IF the destination port is the server port
and the ACK flag is set
and the elapsed time value is less than two seconds
and the source IP address
and port are equal to the specific client which is already stored
in the database
Forward the connection to the real server
ELSE
Send RST to the client (pretend to be the server)
Ignore the connection
Delete from the database
END IF
END IF
END WHILE

Figure 5.16. SYN Proxy work flow.

Killer application is working like a gateway between the client and the real server, it protects the server from the unreliable clients. The application should send ACK packet and probably needs to send RST packet while it pretends the be the client against the server. The software code that sending these packets, is really important, because it should generate packets with the true parameter for not to be discarded by the server TCP/IP stack.

In conclusion, with the help of this IPS module, a malicious client who sends only SYN packet and not send any other data or ACK packet back to the client, can never access to the server in reality.

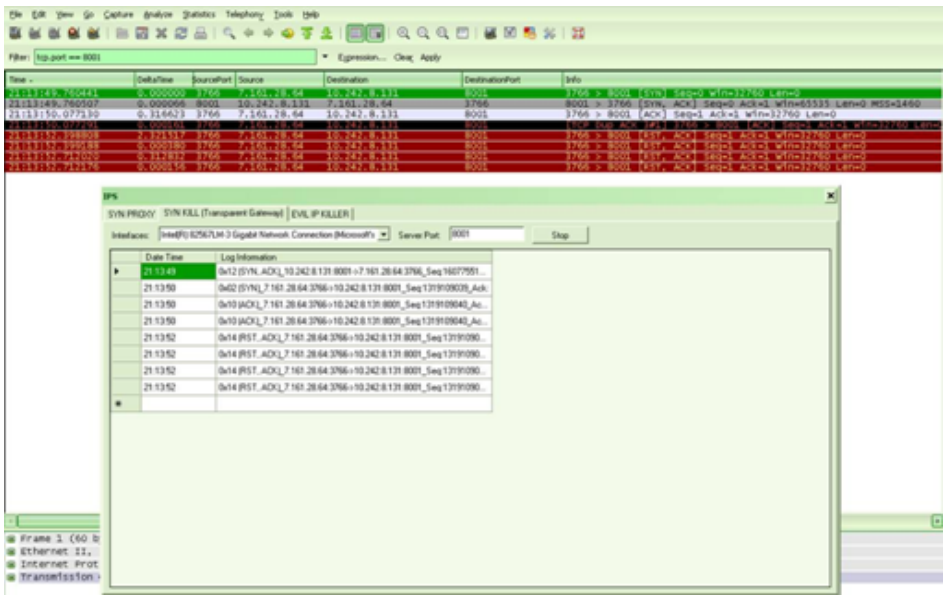

Figure 5.17. SYN killer transparent gateway tool.

The general view of the SYN Killer transparent gateway mode of the IPS tool is above. The inputs of the IPS tool are real server port number and ethernet card of the server. The tool is monitoring the event that occurs on the given server port, such as SYN packet of a client etc. When a client sends SYN to the server, then the server sends SYN_ACK response back to the client. These packet can be seen in the log information tab of the tool with the specific time values. On the other hand, at this time, the tool sends ACK response back to the server like a client does. Therefore, the server supposes that the client sends this ACK, and makes “Established” the status of

the connection. If the real client does not send any ACK packet within a particular predefined time period, the tool again sends RST packet to the server like a client does. Therefore, the server drops the connection after it gets this RST request packet. These packets also can be seen in the log information tab. With the help of this tab, it is possible to trace the network packet flow and connection details. The method is explained below:

5.2.2.3. SYN Killer (Evil IP). IPS SYN killer evil IP mode works like SYN killer transparent gateway mode, but there is a small differences between them. When a client sends a SYN packet to the real server, the real server sends SYN_ACK packet back to the client immediately. The SYN IP killer records this instant time values and starting to wait for a specific time period to expire. When the time is up, it checks there is any ACK response back from the client or not. If there is no reply from the client, it sends RST packet to the server like client sends it. Therefore, like other methods do, it also helps to prevent the server to not holding resources unnecessarily.

The general view of the SYN killer evil IP mode of the IPS tool is above. The input parameters of the tool are the ethernet card of the server, and the server port number. The application is tracing the connection of the real server instantly by listening the ethernet card like a sniffer. The tool controls the open connections on the server port every second (adjustable predefined value), If there is a connection that has been opened for ten seconds (adjustable predefined value) thinks that there is an SYN attack trying to be performed against the server, and close all those open connections with sending RST packet to the server, with appropriate TCP and connection parameters. The most important thing is deciding the timeout values, following criteria should be concerned while giving this timeout limitations:

- Network round trip time (NRTT)
- Server application
- Client and server performance
- General application usage and user response time (end to end)

WHILE the proxy port is opened
IF a new connection arrives
Store all the packet information into the database
and wait for the server to send the SYN_ACK packet
IF the source port equals to the server port
and the SYN_ACK flag is set
and the destination IP address
and port are equal to the specific client which is already stored in the
database
Send ACK packet to the server (pretend to be the client)
Connection state is ESTABLISHED
Wait for 2 seconds
IF the destination port is the server port
and the ACK flag is set
and the elapsed time value is less than two seconds
and the source IP address
and port are equal to the specific client which is already stored
in the database
Forward the connection to the real server
ELSE
Send RST to the server (pretend to be the client)
Ignore the connection
Delete from the database
END IF
END IF
END IF
END WHILE

Figure 5.18. SYN Killer Transparent Gateway work flow.

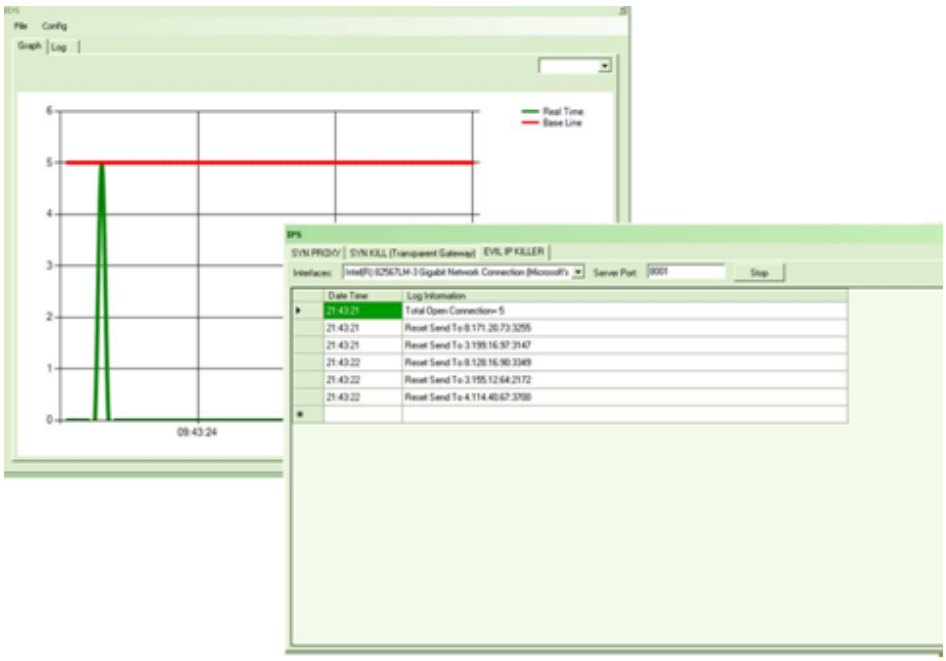

Figure 5.19. SYN killer evil IP tool.

Otherwise, regular client connections might be dropped instead of malicious attackers. The method is explained below:

5.3. DDoS Attacks Tool Properties in Terms of Literature

We mentioned about DDoS Attack taxonomies in the previous chapters. Now, we will discuss about how custom SYN client attack tool suits to this taxonomy.

In Terms of Architecture Base

SYN client tool has a feature named reflector, with this feature attack tool abuses an infected machine to send SYN packet to the victim server machine. Taxonomy corresponding:

- Reflector Model

In Terms of Degree of Automation Base

WHILE the proxy port is opened
IF a new connection arrives
Store all the packet information into the database and wait for the server
to send the SYN_ACK packet
IF the source port equals to the server port
and the SYN_ACK flag is set
and the destination IP address
and port are equal to the specific client which is already stored
in the database
Wait for 2 seconds
IF the destination port is the server port
and the ACK flag is set
and the elapsed time value is less than two seconds
and the source IP address
and port are equal to the specific client which is already stored
in the database
Ignore the connection
Delete from the database
ELSE
Send RST to the server (pretend to be the client)
Ignore the connection
Delete from the database
END IF
END IF
END IF
END WHILE

Figure 5.20. SYN Killer Evil IP Address work flow.

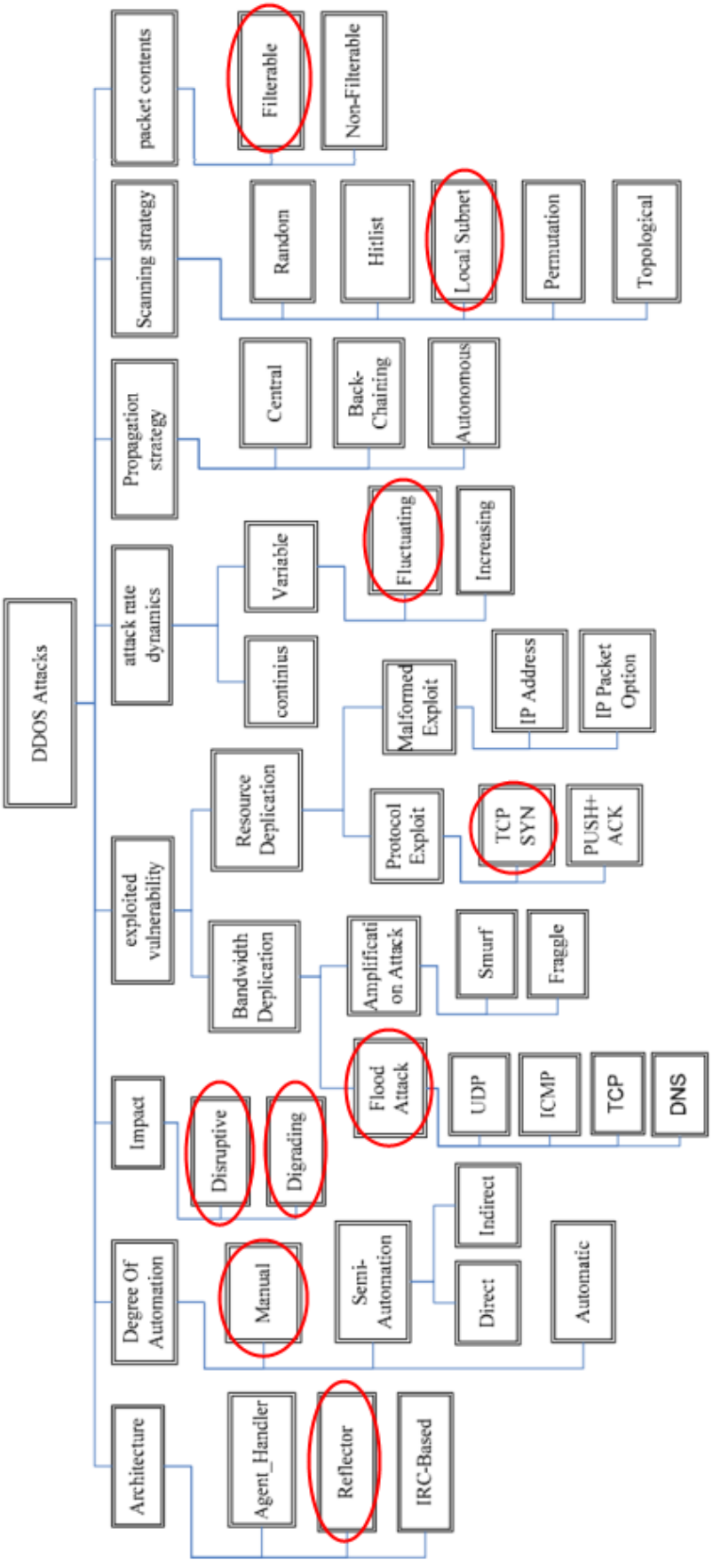

Figure 5.21. SYN client tool and DDoS taxonomy.

SYN client tool needs to take an action for from the user manually to start the attack to the server (victim) machine. Therefore, it is not an automated system, the process initiation should be done manually. Taxonomy corresponding:

- Manual Model

In Terms of Impact Base

SYN client tool is developed to have an ability to send multiple SYN packet at the sametime. The only limitation is the hardware or software performance of the attacker machine where the SYN tool runs on. Because of this reason, it is possible to create a huge volume of SYN flood. This can also cause to deplete all the bandwidth of the server (victim or target) network. Taxonomy corresponding:

- Disruptive and Degrading

In Terms of Vulnerability Base

The SYN client tool is an attacker machine that can cause to crash the victim system because of the saturated network bandwidth. Furthermore, the SYN attack client tool exploits the TCP protocol structure. Taxonomy corresponding:

- Bandwidth Depletion Attacks – Flood Attacks
- Resource Depletion Attacks – Protocol Exploit Attacks

In Terms of Attack Rate Dynamic Base

It is possible in the client tool to decide how many TCP SYN packet should send to the server system. Therefore, the attack rate is adjustable by the attacker. Taxonomy corresponding:

- Variable – Fluctuating

5.4. DDoS Defense Tool Properties in Terms of Literature

We mentioned about DDoS defense taxonomies in the previous chapters. Now, we will discuss about how custom IPS and IDS mechanism tools suit to this taxonomy.

In Terms of Mitigation

Intrusion prevention and also intrusion detection proposed models works like a middle secure layer between the client (possible attacker) and the server (the victim or the target). They have many smart decision algorithms based on predefined calculations. According to the conclusions of the calculations, it can forward or drop the connection. In addition, sometime they can warn the users. Therefore, they can protect the real server from malicious clients. The tool also has an ability to warning the users about traffic anomalies. Taxonomy corresponding:

- Server – Overlay
- Server- DDoS Aware Algorithms
- Server- Resource Accounting

In terms of Deterrence The tool stores the source IP addresses of the clients. Therefore, the network status estimation is available. Furthermore, the traffic performance and instant status are always being monitored by the tool and recorded. Therefore, historical data also exists and network performance tracing can be done any time. Taxonomy corresponding:

- Forensics (storing fake IP addresses, tracing network performance)

In Terms of Prevention

The intrusion prevention tools have many properties that helps you to prevent

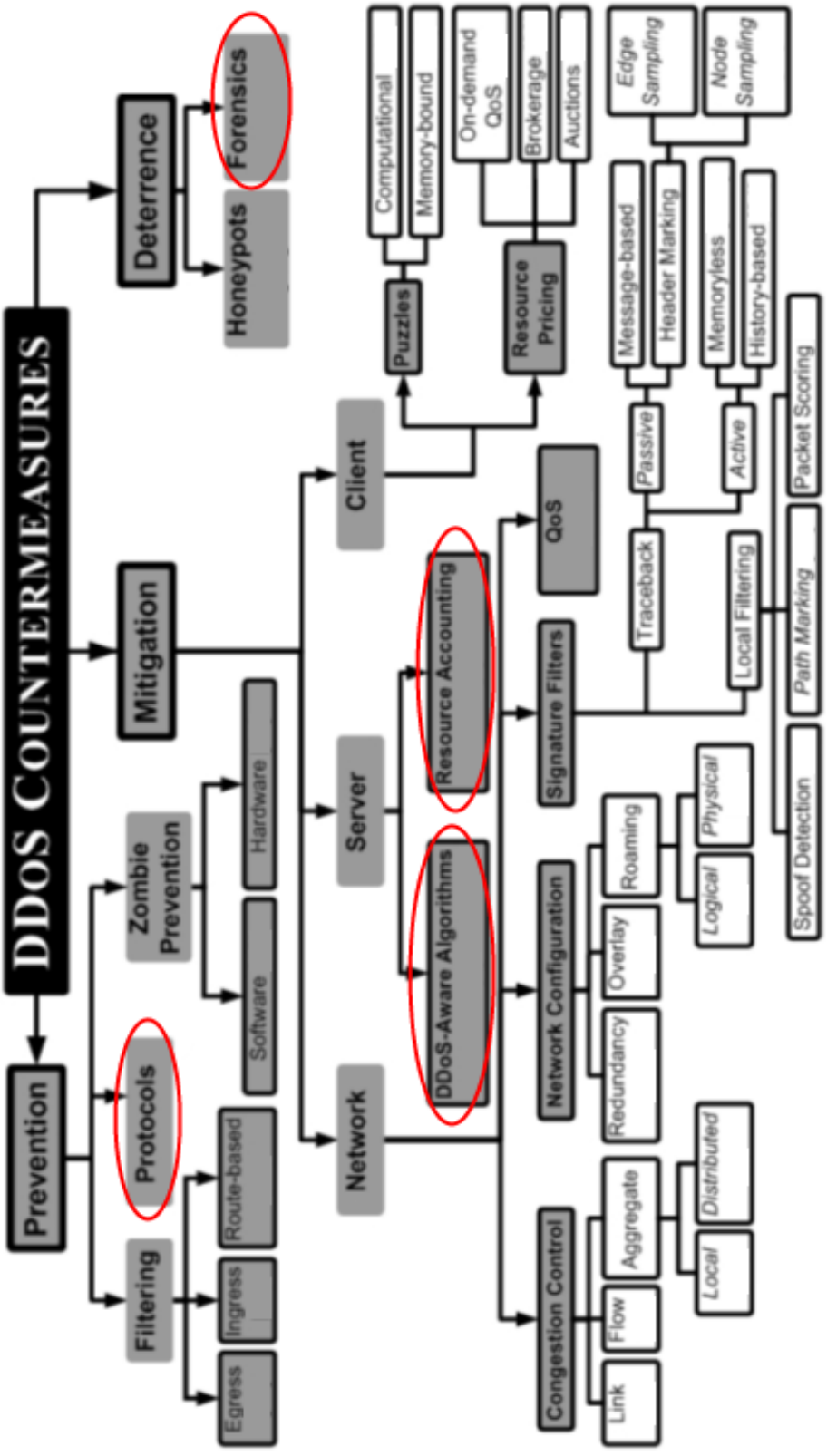

Figure 5.22. IPS/IDS mechanisms tool and taxonomy.

SYN attacks. Therefore, they help the server (the victim or the target) machine to protect its session table, backlog queue and also network badwidth. Taxonomy corresponding:

- Protocol

5.5. Virtual Server with New Perspective

Generaly, custom virtual server works like a server in terms of functionality. It has a port number and IP address. It issues regular socket API functions like listen, bind, accept etc. The virtual server program is a socket application program and the socket type of the server which is also called the listener is a stream socket.

To run the virtual server, we need the real server information, a service port number, and the threshold value. The details of these inputs are below:

- Real server information: The IP address and the port number of the real server.
- Threshold value: The limit of the connection requests that can come simultenously
- Service port number: The port number of the virtual server

When a virtual server starts, it opens a port on the system. When a client wants to connect to that port, desicion mechanism comes to the stage. Virtual server never accepts the connection, before it gets positive feedback from the decision side. Assume that, desicion mechanism sends positive reponse to the virtual server, then it accepts the connection and forward it to the real server. Thus, the mission is completed without a problem. On the other hand, if it gets negative response from the desicion algorithm then it rejects the client and never forwards that connection to the real server.

The system components provide the virtual server to decide both an instant connection rate and historical values. That makes the virtual server more reliable, and helps to protect the real server itself.

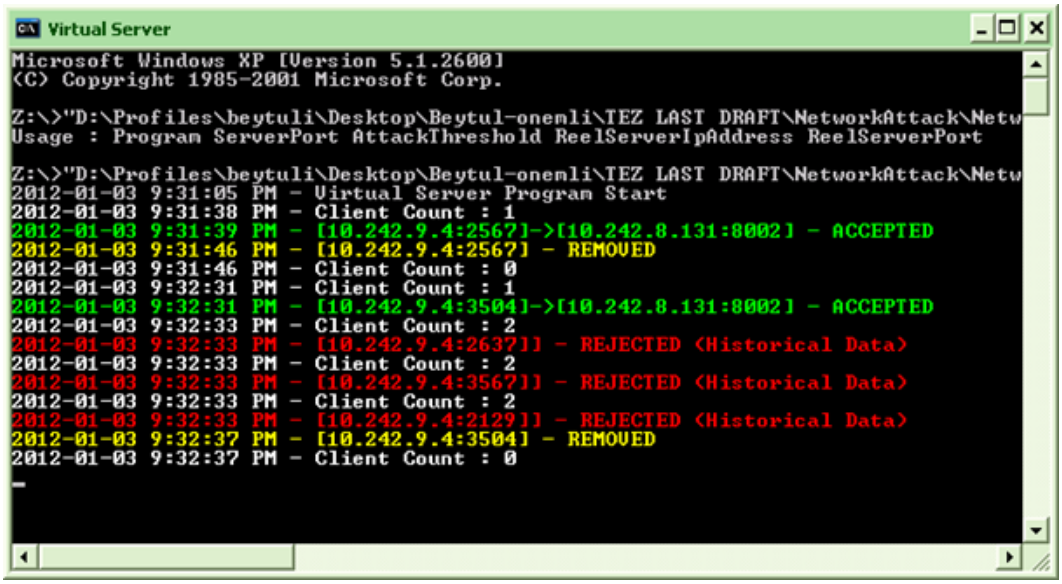

Figure 5.23. New proposed virtual server.

The main purpose of the decision algorithm is to understand the reliability of the client.

The algorithm compose of the following components:

- Historical database: Trusted client information is recorded here
- Instant connection threshold value: To decide at the first sight for group of connection
- Data mining: To calculate the historical trusted threshold values for new connections

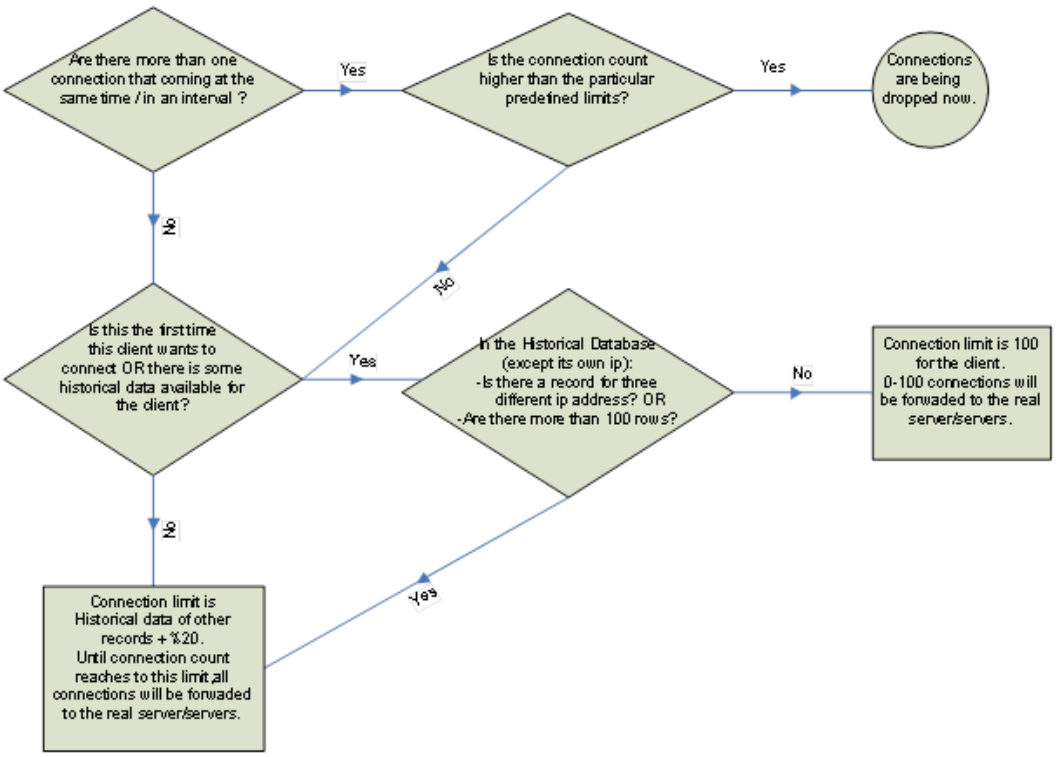

Figure 5.24. Decision algorithm flow chart.

6. CONCLUSION AND FUTURE WORKS

DDoS Attack, detection and prevention chain is covered with every components in this work. Furthermore, a virtual server application developed with a new perspestive.

We developed an custom attack tool that has four different features. The features are port scanning, SYN flood attack with local source IP address, SYN flood attack with manipulated source IP address and SYN flood attack with pre-defined reflector IP address. The tool is more powerful and flexible SYN attack tool. To run this tool only one computer can be enough for a regular server to remain unresponsive. If the target or in other words the victim system is enhanced, the SYN attack tool random IP feature may be used. Therefore, it is almost impossible for the victim to traceback of the attacker's IP address. The tool also has a feature that called port scanning that checks the server ports to find out which ports are available. The result of the scan can be used to decided on which port will be attack target. We also developed an intrusion detection tool, that has two important features. The first one is tracing SYN-FIN pair and examining the result of these pairs on deciding the open connections and warning users about them. The second one is calculating a base line according to the historical database. It is possible to decide the current traffic rate is normal or abnormal using the results of the calculated baseline. The main important difference the new IDS tool is recording all parameters about the connections to the database. Therefore, desicion mechanisms is more accurate according to the other proposed IDS models. It is almost no chance to mix a connection with another connection that coming from the same source.

In addition, we also developed a new intrusion prevention system tool. The tool has three different features. These features are SYN proxy, SYN killer transparent gateway and SYN killer evil IP address. These mechanisms are implemented and making simulation about them. We see that, how the prevention mechanisms are working against the SYN flood attacks, and compare the results about them. Lastly, we developed a custom server application, which is called a virtual server. The server

is a multithread socket application programme. The most important feature of the virtual server is the decision mechanism. It has an enhanced algorithms, that barely no way to forward malicious traffic to the real server side.

In conclusion, we need to fully compare our schemes with the existing detection mechanisms in the feature. Furthermore, to be able to make more comprehensive data mining we should improve the database structure. In addition, we are planning to install these new intrusion prevention and intrusion detection applications on a web server which runs the Internet environment and see the results. According to the results, we plan to improve our distributed systems.

REFERENCES

1. Parziale, L., D. T. Britt, C. Davis, J. Forrester, C. M. W. Liu and N. Rosselot, *TCP/IP Tutorial and Technical Overview*, Prentice-Hall, Inc., New Jersey, 2006.

2. Asosheh, A., Dr. and N. Ramezani, "A comprehensive taxonomy of DDOS attacks and defense mechanism applying in a smart classification", *WSEAS Transaction on Computers*, Vol. 7, pp. 281–290, Apr. 2008.

3. Champagne, D. and R. B. Lee, "Scope of DDoS Countermeasures: Taxonomy of Proposed Solutions and Design Goals for Real-World Deployment", *8th International Symposium on Systems and Information Security (SSI'2006)*.

4. Casad, J. and D. Newland, *MCSE Training Guide: Networking Essentials*, New Riders, 1997.

5. Stallings, W., *Network Security Esessentials Applications and Standards*, Pearson Education, New Jersey, 2003.

6. Stein, L. D. and J. N. Stewart, *The World Wide Web Security FAQ*, 2002, `http://www.w3.org/Security/Faq`, accessed at February 2011.

7. Houle, K. J. and G. M. Weaver, *Trends in Denial of Service Attack Technology*, 2001, `http://www.cert.org/archive/pdf/DoS_trends.pdf`, accessed at October 2011.

8. DOE, *L-040: The Ramen Worm*, 2001, `http://www.ciac.org/ciac/bulletins/l-040.shtml`, accessed at February 2011.

9. Houle, K. J. and G. M. Weaver, *CERT Advisory CA-2001-19 Code Red Worm Exploiting Buffer Overflow in IIS Indexing Service DLL*, 2001, `http://www.cert.org/advisories/CA-2001-19.html`, accessed at October 2011.

10. Mirkovic, J. and P. Reiher, "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms", *ACM SIGCOMM Computer Communication Review*, Vol. 34, pp. 39–53.

11. Specht, S. M., "Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures", *Proceedings of the 17th International Conference*, pp. 543–550, 2004.

12. Douligeris, C. and A. Mitrokotsa, "DDoS Attacks and Defense Mechanisms: Classification and State-of-the-art", *Computer Networks*, Vol. 44, No. 5, pp. 643 – 666, Apr. 2004.

13. Tariq, U. and M. Hang, "A Comprehensive Categorization of DDoS Attack and DDoS DefenseTechniques", *ADMA'06*, pp. 1025–1036, 2006.

14. Mahajan, R., S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson and S. Shenker, "Controlling high bandwidth aggregates in the network", *ACM Computer Communication Review*, Vol. 32, pp. 62–73, 2002.

15. Carpenter, J. and C. Dougherty, *Continuing Threats to Home Users*, 2001, `http://www.cert.org/advisories/CA-2001-20.html`, accessed at September 2011.

16. Bysin and Knight.c, *Packet Storm Security*, 2001, `http://packetstormsecurity.nl/distributed/knight.c`, accessed at March 2011.

17. Dietrich, S., N. Goddard and N. Long, "Analyzing Distributed Denial of Service Tools: The Shaft Case", *In Proceedings of USENIX LISA'2000*, pp. 329–339, 2000.

18. Hussain, A., J. Heidemann and C. Papadopoulos, "A Framework for Classifying Denial of Service Attacks", *In Proceedings of ACM Special Interest Group on Data Communications*, pp. 99–110, 2003.

19. Kumar, V. A., "Sophistication in Distributed Denial-of-service Attacks on the Internet", *Current Science Journal*, Vol. 87, Oct. 2004.

20. Bellovin, S. M., "Security Problems in the TCP/IP Protocol Suite", *Computer Communications Review*, Vol. 19, pp. 32–48, 1989.

21. Systems, C., *Defining Strategies to Protect Against TCP SYN Denial of Service Attacks*, 2002, `http://www.cisco.com/en/US/tech/tk828/technologies_tech_note09186a00800f67d5.shtml`, accessed at October 2011.

22. Moore, D. and C. Shannon, *The Spread of the Code Red Worm (crv2)*, 2008, `http://www.caida.org/research/security/code-red/coderedv2_analysis.xml`, accessed at February 2011.

23. Weaver, N., *Warhol Worms, the Potential for Very Fast Internet Plagues*, 2001, `http://www.cs.berkeley.edu/ nweaver/warhol.html`, accessed at February 2011.

24. Schuchter, M., *Distributed Denial of Service (DDoS) Attack*, 2010, `http://www.parabon.com/faqs/ddos-timeline.html`, accessed at March 2011.

25. Sangpachatanaruk, C., S. M. Khattab, T. Znati, R. G. Melhem and D. Mossé, *Design and Analysis of a Replicated Elusive Server Scheme for Mitigating Denial of Service Attacks*, 2004, `http://dx.doi.org/10.1016/j.jss.2003.09.012`, accessed at March 2011.

26. Brustoloni, J., "Protecting Electronic Commerce from Distributed Denial of Service Attacks", *11th International Conference on World Wide Web*, May 2002.

27. Spatscheck, O. and L. L. Peterson, "Defending against Denial of Service Attacks in Scout", *In Proceedings of the 1999 Usenix/ACM Symposium on Opreating system*

Desing and Implementation, pp. 59–72, 1999.

28. Keromytis, A., V. Misra and D. Rubenstein, "SOS: Secure Overlay Services", *In Proceedings of ACM Special Interest Group on Data Communications*, pp. 61–72, 2002.

29. Druschel, P. and G. Banga, "Lazy Receiver Processing (LRP): A Network Subsystem Architecture for Server Systems", *SIGOPS Oper. Syst. Rev.*, Vol. 30, pp. 261–275, 1996.

30. Bernstein, D. J., *SYN Cookies*, 2004, `http://cr.yp.to/syncookies.html`, accessed at February 2011.

31. Spitzner, L., *Honeypots: Tracking Hackers*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.

32. CERT CC., *Erkms and Li0n worms*, 2001, `http://www.cert.org/incident_notes /IN2001-03.html`, accessed at May 2011.

33. Labovitz, C., D. McPherson, S. Iekel-Johnson and M. Hollyman, *Internet Traffic Trends - A View from 67 ISPs*, 2008, `http://www.nanog.org/meetings/nanog43/presentations/Labovitz_internetstats_N43.pdf`, accessed at May 2011.

34. Wang, H., D. Zhang and K. G. Shin, "Detecting SYN Flooding Attacks", *In Proceedings of the IEEE Infocom*, pp. 1530–1539, IEEE, 2002.

35. Kompella, R. R., S. Singh and G. Varghese, "On scalable attack detection in the network", *In IMC '04: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, pp. 187–200, ACM Press, 2004.

36. Sun, C., J. Fan and B. Liu, "A Robust Scheme to Detect SYN Flooding At-

tacks", *International Conference on Communications and Networking in China*, Aug. 2007.

37. Chen, W. and D.-Y. Yeung, "Defending against TCP SYN Flooding Attacks under Different Types of IP Spoofing", *Fifth International Conference on Networking*, 2006.