MITIGATING DoS/DDoS ATTACKS USING IPTABLES

Bahaa Qasim M. AL-Musawi
College of Engineering
University Of Kufa , An Najaf, Iraq
bahaaqm@yahoo.com

Abstract

Denial-of-Service (DoS) is a network security problem that constitutes a serious challenge to reliability of services deployed on the servers. The aim of DoS attacks is to exhaust a resource in the target system, reducing or completely subverting the availability of the service provided. Threat of DoS attacks has become even more severe with DDoS (Distributed Denial-of-Service) attack. It is an attempt by malicious users to carry out DoS attack indirectly with the help of many compromised computers on the Internet.

Service providers are under mounting pressure to prevent, monitor and mitigate DoS/DDoS attacks directed toward their customers and their infrastructure. Defending against those types of attacks is not a trivial job, mainly due to the use of IP spoofing and the destination-based routing of the Internet, though there are many proposed methods which aim to alleviate the problem like Firewalls, Intrusion Detection Systems, Ingress filtering, IP Traceback, SYN Proxy etc.

This paper discusses the efficient packet filtering technique using firewall to defend against DoS/DDoS attacks. Firewall scripts are written using command-line tool iptables in Linux to deny the suspicious traffic. Packet analyzer tool used to showcase the effectiveness of the scripts in mitigating the various kinds of DoS/DDoS attacks.

Keywords: **DoS attacks, DDoS attacks, iptables.**

1. Introduction

Internet grows rapidly since it was created. Via the Internet infrastructure, hosts can not only share their files, but also complete tasks cooperatively by contributing their computing resources. Moreover, an end host can easily join the network and communicate with any other host by changing packets. These are the encouraging features of the Internet, openness and scalability. However, the attackers can also take advantage of this to launch attacks that are more powerful than those launched by a single machine. Denial-of Service Attack is one type of such attacks [1].

A Denial of Service (DoS) attack is a type of attack focused on disrupting availability

of service. Such an attack can take many shapes, ranging from an attack on the physical IT environment, to the overloading of network connection capacity, or through exploiting application weaknesses. Gligor et al. [2] defined DoS as: "a group of otherwise-authorized users of a specific service is said to deny service to another group of authorized users if the former group makes the specified service unavailable to the latter group for a period of time which exceeds the intended (and advertised) waiting time." Internet-facing and other networked infrastructure components are at risk of DoS for two primary reasons:

1. Resources such as bandwidth, processing power, and storage capacities are not unlimited and so DoS attacks target these resources in order to disrupt systems and networks.
2. Internet security is highly interdependent and the weakest link in the chain may be controlled by someone else thus taking away the ability to be self reliant [1,2].

In Distributed Denial of Service (DDoS) attacks, attackers do not use a single host for their attacks but a cluster of several dozens or even hundreds of computers to do a coordinated strike. From the beginning the evolution of solutions to resolve the occurrence of attacks promoted the evolution of the attacks itself, and nowadays DoS attacks have been superseded by DDoS attacks [3].

There are two major reasons making DDoS attacks attractive for attackers. The first is that there are effective automatic tools available for attacking any victim, i.e., expertise is not necessarily required. The second is that it is usually impossible to locate an attacker without extensive human interaction or without new features in most routers of the Internet [4,5].

The paper organized as follows: section 2 present a survey of mitigating DoS/DDoS attacks. Section 3 illustrates types of DoS/DDoS attacks. Section 4 describes iptable. Section 5 presents the experimental setup and measurements and finally, section 6 presents the conclusions of this work.

2. Related Work

Mitigating DoS/ DDoS attacks at the origin or within the core of the internet seems to be an impossible task due to the distributed and authorization-free nature of the IP based network. Various approaches to find the source IP of attacker using filtering mechanisms have been proposed. [6,7,8].

Luo et al. [9] suggested a scheme to detect Pulsing DoS (PDoS) attacks at the victim's network using two anomalies caused by PDoS attacks, namely the fluctuation of the incoming data traffic, and the decline of outgoing TCP ACK traffic.

S. Kumar and R. S. Gade evaluated the effectiveness of a security device called Netscreen 5GT (or NS-5GT) from Juniper Networks under Layer-4 flood attacks at different attack loads. This security device NS-

5GT comes with a feature called TCP-SYN proxy protection to protect against TCP-SYN based DDoS attacks, and UDP protection feature to protect against UDP flood attacks.

In their work, they conducted real experiments to measure the performance of this security device NS-5GT under the TCP SYN and UDP flood attacks and test the performance of these protection features. It was found that the Juniper's NS-5GT mitigated the effect of DDoS traffic to some extent especially when the attack of lower intensity. However, the device was unable to provide any protection against Layer4 flood attacks when the load exceeded 40Mbps [10].

K. W. M. Ghazali and R. Hassan made a study reviews about recent researches on flood attacks and their mitigation, classifying such attacks as either high-rate flood or low-rate flood. Finally, the attacks are compared against criteria related to their characteristics, methods and impacts. They found Denial-of-service flood attacks vary in their rates, traffic, targets, goals and impacts. However, they have general similarities that are the methods used are flooding and the main purpose is to achieve denial of service to the target [11].

In this work, capability of firewall is explored to defend against this attack. To determine whether the network traffic is legitimate or not, a firewall relies on a set of rules it contains that are predefined by a network or system administrator. These rules tell the firewall whether to consider as legitimate and what to do with the network traffic coming from a certain source, going to a certain destination, or having a certain protocol type.

Rashid Rehman and Sheikh Obaid Ur Rahman made testing and analysis of personal firewall, they tested and analyzed the security firewalls against TCP ACK, TCP SYN, TCP FIN, TCP Connect, Echo Ping, UDP and Denial of Service attacks (Ping of Death, Teardrop, and Land Attack) to check security issues. They found from the results that Zone Alarm provides the best security results against all attacks and scanning methods. The reason is that it shows all ports are filtered at full security. It also gives warning alert against denial of service attacks and blocks the infected packets, while other firewalls didn't show the same results [12].

3. Types of DoS/DDoS Attacks

In DOS/DDOS attacks the attacker sends packets directly from his computer(s) to the victim's site but the source address of the packets may be forged. There are many tools available to allow this type of attack for a variety of protocols including ICMP, UDP and TCP. Some common tools include stream2, synhose, synk7, synsend, and hping2. Some of the common DDoS attacks are discussed below [6,13].

3.1 UDP Flood Attack

In UDP Flood attack attacker sends large number of UDP packets to a victim system, due to which there is saturation of the network and the depletion of available bandwidth for legitimate service requests to the victim system [14].

A UDP Flood attack is possible when an attacker sends a UDP packet to a random port on the victim system. When the victim system receives a UDP packet, it will determine what application is waiting on the destination port. When it realizes that there is no application that is waiting on the port, it will generate an ICMP packet of "destination unreachable" to the forged source address. If enough UDP packets are delivered to ports of the victim, the system will go down. By the use of a DoS tool the source IP address of the attacking packets can be spoofed and this way the true identity of the secondary victims is prevented from exposure and the return packets from the victim system are not sent back to the zombies [15].

3.2 ICMP Flood Attack

ICMP Flood attacks exploit the Internet Control Message Protocol (ICMP), which enables users to send an echo packet to a remote host to check whether it's alive. More specifically during a DDoS ICMP flood attack the agents send large volumes of ICMP_ECHO_ REPLY packets ("ping") to the victim. These packets request reply from the victim and this results in saturation of the bandwidth of the victim's network connection [16]. During an ICMP flood attack the source IP address may be spoofed.

3.3 SYN Flood Attack

In a SYN Flood attack, the victim is flooded with half open connections. The client system begins by sending a SYN message to the server. The server then acknowledges the SYN message by sending SYN-ACK message to the client. The client then finishes establishing the connection by responding with an ACK message. The connection between the client and the server is then open, and the service-specific data can be exchanged between the client and the server. **Fig.1** shows the view of this message flow [9]:

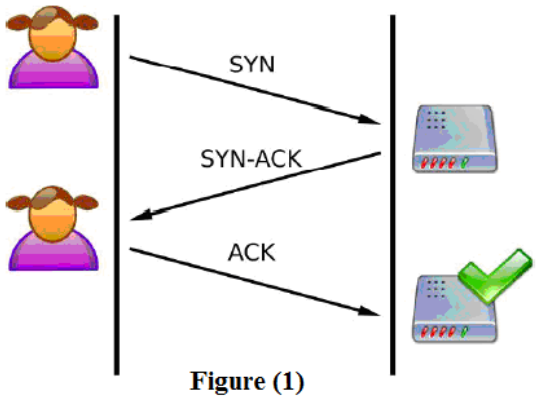

Figure (1)

The potential for abuse arises at the point where the server system has sent an acknowledgment (SYN-ACK) back to client but has not yet received the ACK message. This is known as half-open connection. The server has built in its system memory a data structure

describing all pending connections. This data structure is of finite size, and it can be made to overflow by intentionally creating too many partially-open connections.

Creating half-open connections is easily accomplished with IP spoofing. The attacking system sends SYN messages to the victim server system; these appear to be legitimate but in fact reference a client system that is unable to respond to the SYN-ACK messages. This means that the final ACK message will never be sent to the victim server system as shown in **Fig.2** [9].

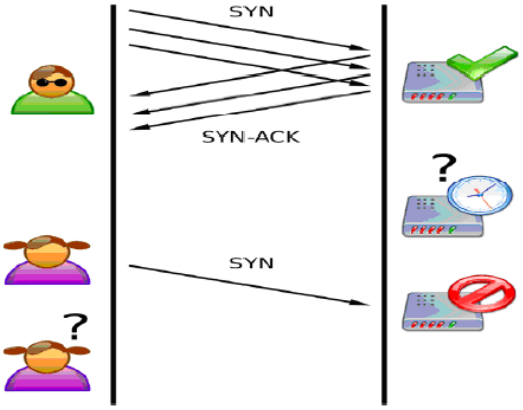

Figure (2)

3.4 Smurf Attack

In a "smurf" attack, the victim is flooded with Internet Control Message Protocol (ICMP) "echo-reply" packets. On IP networks, a packet can be directed to an individual machine or broadcast to an entire network. When a packet is sent to an IP broadcast address from a machine on the local network, that packet is delivered to all machines on that network.

In the "smurf" attack [17], attackers are using ICMP echo request packets directed to IP broadcast addresses from remote locations to generate denial-of-service attacks. When the attackers create these packets, they do not use the IP address of their own machine as the source address. Instead, they create forged packets that contain the spoofed source address of the attacker's intended victim. The result is that when all the machines at the intermediary's site respond to the ICMP echo requests, they send replies to the victim's machine. The victim is subjected to network congestion that could potentially make the network unusable.

3.5 Teardrop Attack

This type of denial of service attack exploits the way that the Internet Protocol (IP) requires a packet that is too large for the next router to handle be divided into fragments. The fragment packet identifies an offset to the beginning of the first packet that enables the entire packet to be reassembled by the receiving system. In the teardrop attack, the attacker's IP puts a confusing offset value in the second or later fragment. If the receiving operating system does not have a plan for this situation, it can cause the system to crash [18].

3.6 Land Attack

The attack involves sending a spoofed TCP SYN packet (connection initiation) with the target host's IP address to an open port as both source and destination. The reason a LAND attack works is because it causes the

machine to reply to itself continuously.

Land attacks have been found in services like Simple Network Management Protocol (SNMP) and Windows 88/tcp (kerberos/global services) which were caused by design flaws where the devices accepted requests on the wire appearing to be from themselves and causing replies repeatedly [19].

4. Iptables

Iptables is part of the Netfilter project. Netfilter is a set of Linux kernel hooks that communicate with the network stack. Iptables is a command and the table structure that contains the rule sets that control the packet filtering.

Iptables is complex. It filters packets by the fields in IP, TCP, UDP, and ICMP packet headers. A number of different actions can be taken on each packet, so the key to iptables happiness is simplicity. Start with the minimum necessary to get the job done, then add rules as you need them. It's not necessary to build vast iptables edifices, and in fact, it's a bad idea, as it makes it difficult to maintain, and will hurt performance [20, 21].

There are three tables in iptables. Any rules or custom chains that you create will go into one of these tables. The filter table is the default, and is the one most used. The filter table contains these built-in chains:

INPUT: Processes incoming packets

FORWARD: Processes packets routed through the host

OUTPUT: Processes outgoing packets

5. Experimental Setup and Measurements

The experimental setup was made by using Linux Ubuntu server version 10.10 and installed Wireshark, which is network protocol analyzer. Three modules attack will be studied in details as shown next:

5.1 Module 1- SYN Flood Attack

The attack was made by Flooding the victim's machine by running following Hping command from attacker's:

 # hping3 --flood –S –p 80 192.168.0.1

Description:
--flood flag sends the packet as fast as possible
-S flag sets the SYN flag on in TCP mode
-p 80 sends the packet to port 80 on victim's machine (192.168.0.1)

On victim machine, capturing and analyzing traffic using Wireshark show that victim machine (192.168.0.1) is responding to SYN packet by sending back packets with SYN, ACK flags set, but attacker's machine (192.168.0.3) is not participating three way handshake by sending back ACK, instead it is sending RST flag set packet thereby resulting in half open connection. When thousands of such connections are made in a few seconds, victim's resources will get exhausted in no time. **Fig.3** shows TCP flow graph using SYN flood attack.

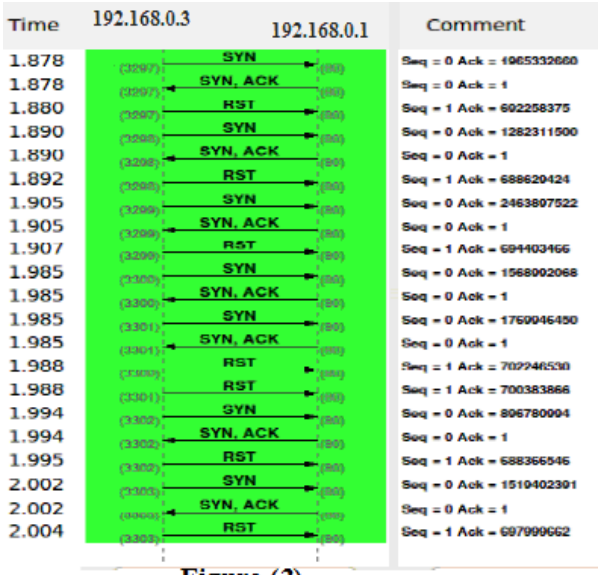

Figure (3)

To defend against SYN Flood Attack, iptables script is writing as bellow.

 # iptables -N syn_flood

 # iptables -A INPUT -p tcp --syn -j syn_flood

 # iptables -A syn_flood -m limit --limit 1/s --limit-burst 3 -j RETURN

 # iptables -A syn_flood -j DROP

Fig.4 shows TCP flow graph after applied the script and recapture packets using Wireshark to test working of the script.

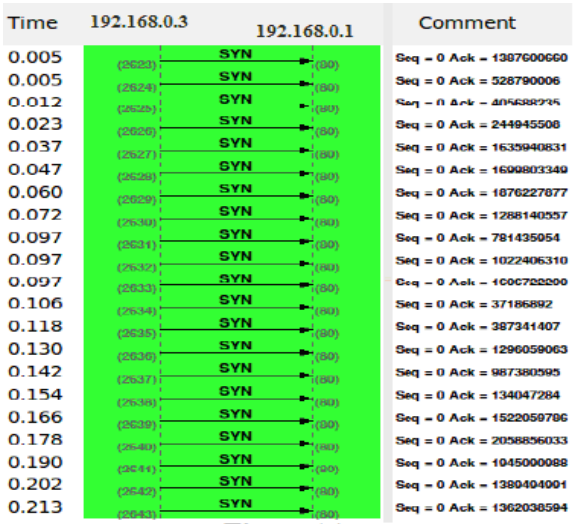

Figure (4)

5.2 Module 2- UDP Flood Attack

The attack was made by Flooding the victim's machine by running following Hping command from attacker's:

 # hping3 –p 80 –i u1000 --udp 192.168.0.1

Description:

-p 80 sends the packet to port 80 on victim's machine (192.168.0.1)

-i u1000 sets the interval between packets as 100 packets per second.

--udp flag sets the udp mode

As seen in **Fig.5** victim's machine (192.168.0.1) is responding with ICMP port unreachable since there is no application running on attacker's machine which sent UDP packet. In this way all of the resources of victim's machine are consumed and legitimate requests will not be served as victim will be busy in serving attacker's invalid requests.

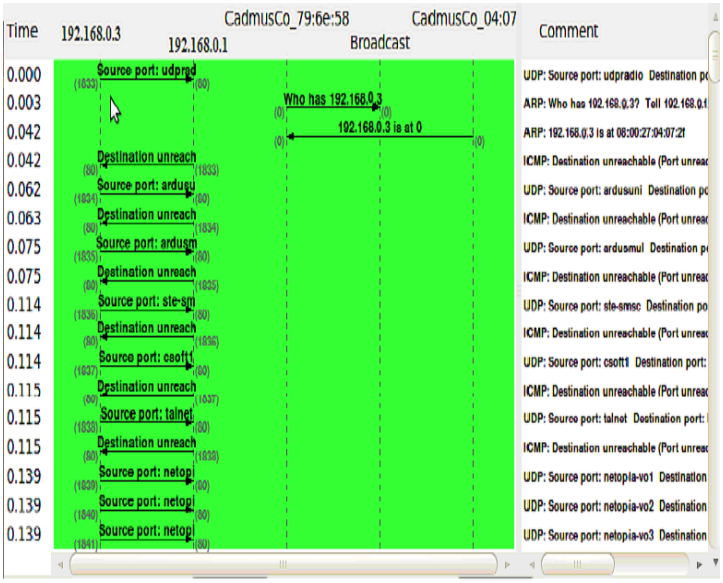

Figure (5)

To defend against UDP Flood Attack, iptables script is writing as bellow:

 # iptables -N udp_flood

 # iptables -A INPUT -p udp -j udp_flood

 # iptables -A udp_flood -m state –state NEW –m recent –update –seconds 1 –hitcount 10 -j RETURN

 # iptables -A udp_flood -j DROP

Fig.6 shows UDP flow graph after applied the script and recapture packets using Wireshark to test working of the script.

Figure (6)

5.3 Module 3-ICMP Flood Attack

The attack was made by Flooding the victim's machine by running following Hping command from attacker's:

 # hping3 –p 80 --flood --icmp 192.168.0.1

Description:

-p 80 sends the packet to port 80 on victim's machine (192.168.0.1)

--flood flag sends the packet as fast as possible

--icmp flag sets the icmp mode

Fig.7 shows victim's machine (192.168.1.2) under ICMP flood attack.

Figure (7)

To defend against ICMP Flood Attack, iptables script is writing as bellow:

 # iptables -N icmp_flood

 # iptables -A INPUT -p icmp -j icmp_flood

 # iptables -A icmp_flood -m limit --limit 1/s --limit-burst 3 -j RETURN

 # iptables -A icmp_flood -j DROP

Fig.8 attacker is sending ICMP Echo Request packets continuously but victim's machine is not responding by sending ICMP Echo Reply packets as all the packets are being dropped by the firewall according to the iptables rules.

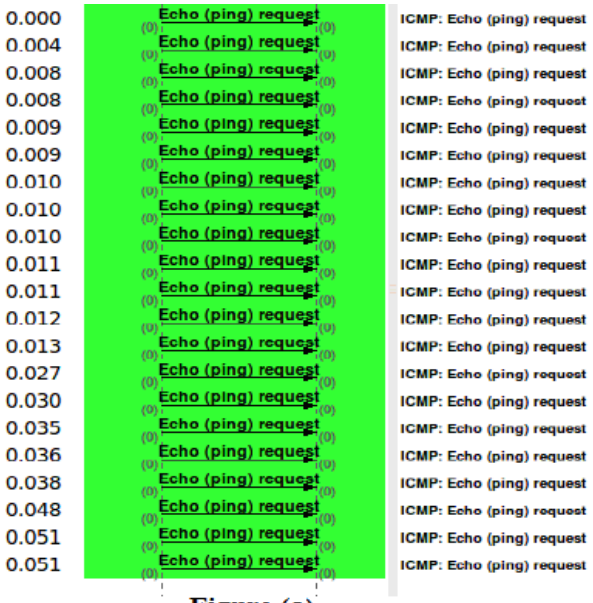

Figure (8)

6. Conclusions

Mitigation of DoS/DDoS attacks is a part of an overall risk management strategy for an organization. Each organization must identify the most important DoS/DDoS risks, and implement a cost-effective set of defense mechanisms against those attack types causing the highest risk for business continuity. Many different defense mechanisms are typically needed to mitigate DoS/DDoS attacks, but it is not cost-effective to blindly choose a large set of defense mechanisms against DoS/DDoS attacks.

In this paper, capability of iptables rules is explored to defend against this attack. To determine whether the network traffic is legitimate or not, a iptable relies on a set of rules it contains that are predefined by a network or system administrator. These rules tell the iptables whether to consider as legitimate and what to do with the network traffic coming from a certain source, going to a certain destination, or having a certain protocol type.

Major concentration of the paper has been on capturing the live traffic using the network protocol analyzer Wireshark and on the basis of analysis scripts using iptables have been developed to allow/deny the network traffic depending upon the traffic rate of any IP address of the computer sending the packets.

As future work, it will be a good idea to apply iptables scripts within openwrt, which is Linux distribution primarily targeted at routing and embedded devices to mitigating DoS/DDoS attacks.

References

[1] Z. Fu, M. Papatriantafilou, P. Tsigas, "**Mitigating Distributed Denial of Service Attacks in Multiparty Applications in the Presence of Clock Drifts**", in 27th IEEE International Symposium on Reliable Distributed Systems (SRDS), Napoli, Italy, 6-8 October, 2008.

[2] Yu, C.-F.; Gligor, V.D ,"**A specification and verification method for preventing denial of service**", IEEE Trans. Software Eng., Volume 16, Issue 6, pp. 581–592, June 1990.

[3] Arun Raj Kumar, P. and S. Selvakumar, "**Distributed Denial of Service Attack Detection using an Ensemble of Neural Classifier**", International Journal of Computer Communications, Elsevier Publications, United Kingdom, Volume 34, Issue 11, 2011, pp. 1328-1341.

[4] Abhilash C. S., Sunil kumar P. V., "**Mitigation of Distributed Denial of Service (DDoS) Threats**", Conference on Advances in Computational Techniques (CACT) 2011.

[5] Chang, R.K.C., "**Defending Against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial**", IEEE Commun. Mag., Volume 40, Issue 10, 2000, pp.42-51.

[6] Ming Li, jun Li and Wei Zhao, "**Simulation Study of Flood Attacking of DDOS**", International Conference on Internet Computing in Science and Engineering (ICICSE),2008.

[7] L.Kavisankar, C.Chellappan, "**Challenging Number Approach for uncovering TCP SYN flooding using SYN spoofing attack**", International Journal of Network Security and its Applications (IJNSA), Vlo.3, No.5, Sep 2011.

[8] A. R. Kumar, P., S. Selvakumar, "**Identifying the Type of High Rate Flooding Attack using a Mixture of Expert Systems**", International Journal Computer Network and Information Security, Volume 1, pp. 1-16, 2012.

[9] X. Luo, R. K. C. Chang, "**On a New Class of Pulsing Denial-of-Service Attacks and the Defense**", Network and Distributed System Security Symposium (NDSS), San Diego, California, USA, 2005.

[10] S. Kumar, R. S. Gade, "**Experimental Evaluation of Juniper Network's Netscreen-5GT Security Device against Layer4 Flood Attacks**", Journal of Information Security, Volume 2, pp.50-58, 2011.

[11] K. W. M. Ghazali, R. Hassan, "**Flooding Distributed Denial of Service Attacks- A review**", Journal of Computer Science, Volume 7, Issue 8, pp.1218-1223, 2011.

[12] Rashid Rehman and Sheikh O. Ur Rahman, "**Testing and Analysis of Personal Firewalls**", M.Sc. thesis, Department of Computer Science and Engineering, University of Gothenburg, 2010.

[13] B. B. Gupta, R.C. Joshi, M. Misra, "**Distributed Denial of Service Prevention Techniques**", International Journal of Computer and Electrical Engineering, Volume 2, Number 2, April 2010.

[14] F. Lau, S. H. Rubin, M. H. Smith and L. Trajkovic, "**Distributed Denial of Service Attacks**" IEEE International Conference on Systems, Man, and Cybernetics, Nashville, 8-11 October 2000, pp. 2275-2280.

[15] J. Markovic, J. Martin, and L. Reiher, "**A Taxonomy of DDoS Attack and DDoS Defense Mechanisms**" ACM SigComm Computer Communication Review, Vol. 34, No. 2, 2004, pp. 39-53.

[16] Felix Lau, Rubin H. Stuart, Smith H. Michael, and et al., "**Distributed Denial of Service Attacks**" in Proceedings of 2000 IEEE International Conference on Systems, Man, and Cybernetics, Nashville, TN, Vol.3, pp.2275-2280, 2000.

[17] CERT advisory CA-1998-01 "**Smurf IP Denial-of-Service Attacks**". Available at http://www.cert.org/advisories/CA-1998-01.html, Jan. 1998.

[18] Hayoung Oh, Inshil Doh, Kijoon Chae, "**Attack Classification Based on Data Mining Technique and its Application for Reliable Medical Sensor**

Communication" International Journal of Computer Science and Applications, Volume 6, No. 3, pp.20-32, 2009.

[19] Vladimir I. Gorodetsky, Igor V. Kotenko, J. Bret Michael, "**Multi-Agent Modeling and Simulation of Distributed Denial of Service Attacks on Computer Networks**", Third International Conference on Navy and Shipbuilding Nowadays (NSN), Krylov Shipbuilding Research Institute (St. Petersburg, Russia, June 2003).

[20] Gregor N. Purdy, "**Linux iptables Pocket Refrence**", CRC O'Reilly Media 2004.

[21] Michael Ras, "**Linux Firewalls Attack Detection and Response With IPTABLES, PSAD, AN*D FWSNORT***", CRC 2007.