ISSN:2249-5789
Manish Saxena et al , International Journal of Computer Science & Communication Networks,Vol 2(5), 607-614

Classification of DDoS Attacks and their Defense Techniques using Intrusion Prevention System

Mohd. Jameel Hashmi[1], Manish Saxena[2] and Dr. Rajesh Saini[3]

1 Research Scholar, Singhania University,
Pacheri Bari, Jhujhunu, Rajasthan, India. Pin - 333515
jam_yas@gmail.com

2 Asst. Professor, MCA Department, FGIET,
Raebareli, UP, India. Pin - 229001
manish.mohan.saxena@gmail.com, URL : www.manishsaxena.in

3 Asst. Professor, Singhania University, CSE Department,
Pacheri Bari, Jhujhunu, Rajasthan, India. Pin - 333515
rajesh.saini4458@gmail.com

Abstract

Distributed Denial of Service (DDoS) Attacks has been increasingly found to be affecting the normal functioning of organizations causing billions of dollars of losses. Organizations are trying their best to minimize their losses from these systems. However, most of the organizations widely use the Intrusion Prevention System (IPS) to observe and manage their networks. One of the major functional areas of a IPS is DDoS detection and DDoS Management. This paper examines how the Network Management Systems could aid in the detection of the DDoS attacks so that the losses from these could be minimized. The classifications of DDoS Attacks and their Defense Techniques have been classified in this paper to have a close look at the DDoS Problem and its severity.

Keywords: *DDoS, Intrusion Prevention System, Classification of DDoS Attacks, Classification of DDoS Defense Systems.*

1. Introduction

One of the Internet's largest security concerns is its intrinsic inability to deal with certain denial-of-service (DoS) type of attacks [1]. The term DoS referring to a situation, where a legitimate requestor of service, or a client, cannot receive the requested service for one reason or the other [2]. DoS attacks can very well be launched both locally and remotely and they range from software exploits to bandwidth consumption attacks. However, targeting network resources attacks are more of a problem. As Houle and Weaver [1] among many others have pointed out, bandwidth consumption attacks are built within the principles of the Internet and thus there is no comprehensive solution to be found. Based on that, it appears that any absolute solution would require a change in the principles themselves.

Distributed denial-of-service (DDoS) attacks are a particular type of DoS attacks and it can cause severe problems in today's computerized world. DDoS, or DDoS attack, is a commonly used term, which refers to a DoS attack using multiple attacking sources and is characterized by coordination [3], [4]. Although not a requisite, DDoS attack is usually aimed to exhaust network resources, which means that DDoS attacks most often are bandwidth consumption attacks. DDoS attacks are now performed by people with fine-tuned objectives in mind. The motives are numerous, such as terrorism, and the possible damages can be severe.

The DDoS field is evolving quickly, and it is becoming increasingly hard to grasp a global view of the problem. This paper strives to introduce some structure to the DDoS field by proposing a classification of DDoS attacks and DDoS defense systems.

This paper is not written to propose or advocate any specific DDoS defense mechanism. Some sections might point out vulnerabilities of certain defense systems, but our purpose is not to criticize but to draw attention to these problems.

After this introduction part rest of the paper is organized as follows: In Section 2 investigation of problem with DDoS attacks is given; in Section 3 their classification has been proposed; in Section 4 solutions to DDoS is given. Finally in Section 5 paper is concluded.

1.1 Objectives to this study

The main purpose of this study is to provide a clear and thorough coverage of the area of DDoS attacks. In principle, this study attempts to aid the DDoS research on the issues related to the field of attack mechanisms. The study is based on a comprehensive literature review, which spans an area of source codes and analyses of DDoS attack tools. The prime objectives of this paper can be summarized to the following:

- Analyse the details of DDoS attack mechanisms and the principles DDoS attacks rely,
- Present the novel classification of DDoS attack mechanisms,
- Discuss a few of the possible evolutions of the DDoS attack mechanisms.

2. The DDoS Attack Problem

The definition provided by [5] is the definition for denial-of-service attack used in this paper:.

"A denial-of-service attack is characterized by an exclusive function of the attack and an explicit attempt by one or more attackers to prevent one or more legitimate users of a service from using that service."

A denial-of-service attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service [5]. A DDoS attack deploys multiple machines to attain this goal. The service is denied by sending a stream of packets to a victim that either consumes some key resource, thus rendering it unavailable to legitimate clients, or provides the attacker with unlimited access to the victim machine so he can inflict arbitrary damage. In Fig. 1 "Ping of Death" type DDoS attack in shown.

2.1 The DDoS Attack Strategy

In order to perform a distributed denial-of-service attack, the attacker needs to recruit the multiple agent (slave) machines. This process is usually performed automatically through scanning of remote machines, looking for security holes that would enable subversion. Vulnerable machines are then exploited by using the discovered vulnerability to gain access to the machine and they are infected with the attack code. The exploit/infection phase is also automated and the infected machines can be used for further recruitment of new agents.

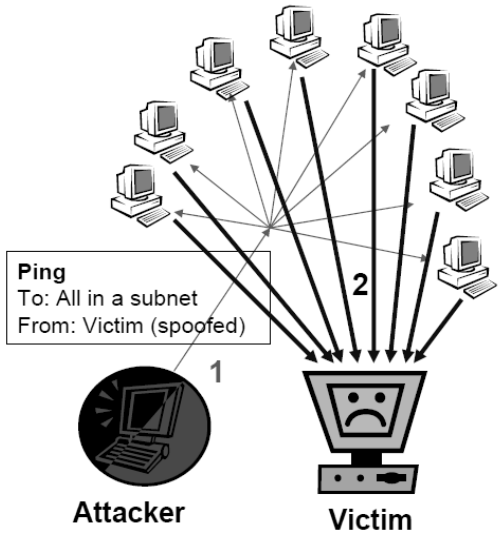

Fig. 1 : A Type of DDoS Attack e.g. "Ping of Death

Agent machines perform the attack against the victim. Attackers usually hide the identity of the agent machines during the attack through spoofing of the source address field in packets. The agent machines can thus be reused for future attacks.

2.2 DDoS Goals

The goal of a DDoS attack is to inflict damage on the victim, either for personal reasons (a significant number of DDoS attacks are against home computers, presumably for purposes of revenge), for material gain (damaging competitor's resources) or for popularity (successful attacks on popular Web servers gain the respect of the hacker community).

3 Classification of DDoS Attacks

To classify the DDoS Attacks, the information on which the classification was built was gathered from live and publicly available DDoS attack tools. The source code of the tools used as references are: [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18] and [19]. Analyses of DDoS attack tools used as references are Trinity (Marchesseau 2000), Shaft (Dietrich, Long and Dittrich 2000), Power bot (Dittrich 2001) and GT bot (GT Bot 2003).

There are three general categories of attacks:

- Against users
- Against hosts
    - fork() bomb

- intentionally generate errors to fill logs, consuming disk space, crashing
- The power switch!!

- Against networks
  - UDP bombing
  - TCP SYN flooding
  - Ping of death
  - Smurf attack

3.1. Classification by Degree of Automation

During the attack preparation, the attacker needs to locate prospective agent machines and infect them with the attack code. Based on the degree of automation of the attack, we differentiate between following:

Manual Attacks

- The attacker scanned remote machines for vulnerabilities, broke into them and installed the attack code, and then commanded the onset of the attack.

Semi-Automatic Attacks

- The DDoS network consists of handler (master) and agent (slave, daemon) machines. The attacker deploys automated scripts for scanning and compromise of those machines and installation of the attack code. He then uses handler machines to specify the attack type and the victim's address and to command the onset of the attack to agents, who send packets to the victim.

Automatic Attacks

- Automatic DDoS attacks additionally automate the attack phase, thus avoiding the need for communication between attacker and agent machines. The time of the onset of the attack, attack type, duration and victim's address is preprogrammed in the attack code. It is obvious that such deployment mechanisms offer minimal exposure to the attacker, since he is only involved in issuing a single command – the start of the attack script. The hardcoded attack specification suggests a single-purpose use of the DDoS network. However, the propagation mechanisms usually leave the backdoor to the compromised machine open, enabling easy future access and modification of the attack code.
- Both semi-automatic and automatic attacks recruit the agent machines by deploying automatic scanning and propagation techniques.

3.2 Classification by Random Scanning

- During random scanning each compromised host probes random addresses in the IP address space, using a different seed. This potentially creates a high traffic volume since many machines probe the same addresses.

Attacks with Hitlist Scanning

- A machine performing 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.

Attacks with Topological Scanning

- Topological scanning uses the information on the compromised host to select new targets. All Email worms use topological scanning, exploiting the information from address books for their spread.

Attacks with Permutation Scanning

- During permutation scanning, all compromised machines share a common pseudo-random permutation of the IP address space; each IP address is mapped to an index in this permutation. A machine begins scanning by using the index computed from its IP address as a starting point. Whenever it sees an already infected machine, it chooses a new random start point. This has the effect of providing a semi-coordinated, comprehensive scan while maintaining the benefits of random probing.

Attacks with Local Subnet Scanning

- Local subnet scanning can be added to any of the previously described techniques to preferentially scan for targets that reside on the same subnet as the compromised host. Using this technique, a single copy of the scanning program can compromise many vulnerable machines behind a firewall. Code Red II [20] and Nimda Worm [21] used local subnet scanning. Based on the attack code propagation mechanism, we differentiate between attacks that deploy central source propagation, back-chaining propagation and autonomous propagation [22].

Attacks with Central Source Propagation

- During central source propagation, the attack code resides on a central server or set of servers. After compromise of the agent machine, the code is downloaded from the central source through a file transfer mechanism. The 1i0n [23] worm operated in this manner.

Attacks with Back-chaining Propagation

- During back-chaining propagation, the attack code is downloaded from the machine that was used to exploit the system. The infected machine then becomes the source for the next propagation step. Back-chaining propagation is more survivable than central-source propagation since it avoids a single point of failure. The

Ramen worm [24] and Morris Worm [25] used backchaining propagation.

Attacks with Autonomous Propagation

- Autonomous propagation avoids the file retrieval step by injecting attack instructions directly into the target host during the exploitation phase. Code Red [26], Warhol Worm [27] and numerous E-mail worms use autonomous propagation.

3.3 Classification by Communication Mechanism

- Based on the communication mechanism deployed between agent and handler machines we divide semi-automatic attacks into attacks with direct communication and attacks with indirect communication.

Attacks with direct communication

- During attacks with direct communication, the agent and handler machines need to know each other's identity in order to communicate. This is achieved by hard-coding the IP address of the handler machines in the attack code that is later installed on the agent. Each agent then reports its readiness to the handlers, who store its IP address in a file for later communication. The obvious drawback of this approach is that discovery of one compromised machine can expose the whole DDoS network. Also, since agents and handlers listen to network connections, they are identifiable by network scanners. A Direct DDoS Attack is shown in Fig. 2

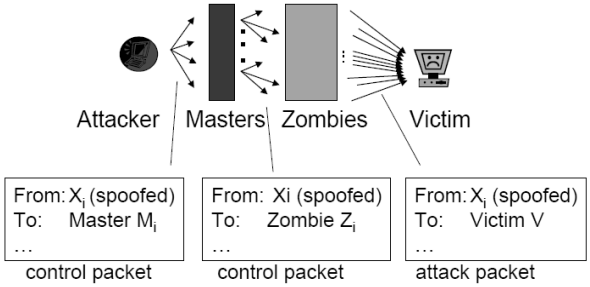

Fig. 2 : A Direct DDoS Attack

Attacks with indirect communication

- Attacks with indirect communication deploy a level of indirection to increase the survivability of a DDoS network. Recent attacks provide the example of using IRC channels [28] for agent/handler communication. The use of IRC services replaces the function of a handler, since the IRC channel offers sufficient anonymity to the attacker. Since DDoS agents establish outbound connections to a standard service port used by a legitimate network service, agent communications to the control point may not be easily differentiated from legitimate network traffic. An attacker controls the agents using IRC communications channels. A Reflector DDoS Attack is shown in Fig. 3

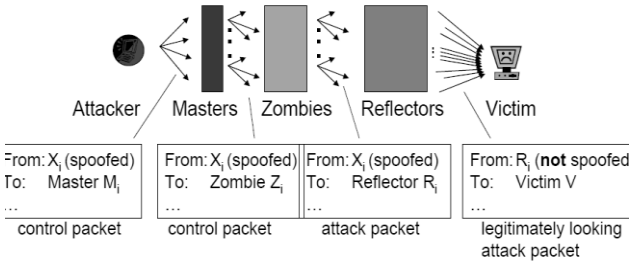

Fig. 3 : A Reflector DDoS Attack

3.4 Classification by Exploited Vulnerability

- DDoS attacks exploit different strategies to deny the service of the victim to its clients. Based on the vulnerability that is targeted during an attack, we differentiate between protocol attacks and brute-force attacks.

Protocol Attacks

- Protocol attacks exploit a specific feature or implementation bug of some protocol installed at the victim in order to consume excess amounts of its resources. Examples include the TCP SYN attack, the CGI request attack and the authentication server attack.
- In the TCP SYN attack, the exploited feature is the allocation of substantial space in a connection queue immediately upon receipt of a TCP SYN request. The attacker initiates multiple connections that are never completed, thus filling up the connection queue indefinitely.
- In the CGI request attack, the attacker consumes the CPU time of the victim by issuing multiple CGI requests.
- In the authentication server attack, the attacker exploits the fact that the signature verification process consumes significantly more resources than bogus signature generation. He sends numerous bogus authentication requests to the server, tying up its resources.

Brute-force Attacks

- Brute-force attacks are performed by initiating a vast amount of seemingly legitimate transactions. Since an upstream network can usually deliver higher traffic volume than the victim network can handle, this exhausts the victim's resources.

Filterable Attacks

- Filterable attacks use bogus packets or packets for non-critical services of the victim's operation, and thus can be filtered by a firewall. Examples of

such attacks are a UDP flood attack or an ICMP request flood attack on a Web server.

Non-filterable Attacks

- Non-filterable attacks use packets that request legitimate services from the victim. Thus, filtering all packets that match the attack signature would lead to an immediate denial of the specified service to both attackers and the legitimate clients. Examples are a HTTP request flood targeting a Web server or a DNS request flood targeting a name server.
- The line between protocol and brute force attacks is thin. Protocol attacks also overwhelm a victim's resources with excess traffic, and badly designed protocol features at remote hosts are frequently used to perform "reflector" brute-force attacks, such as the DNS request attack [29]or the Smurf attack [30]. The difference is that a victim can mitigate the effect of protocol attacks by modifying the deployed protocols at its site, while it is helpless against brute-force attacks due to their misuse of legitimate services (non-filterable attacks) or due to its own limited resources (a victim can do nothing about an attack that swamps its network bandwidth).
- Countering protocol attacks by modifying the deployed protocol pushes the corresponding attack mechanism into the brute-force category. For example, if the victim deploys TCP SYN cookies [31] to combat TCP SYN attacks, it will still be vulnerable to TCP SYN attacks that generate more requests than its network can accommodate.
- It is interesting to note that the variability of attack packet contents is determined by the exploited vulnerability. Packets comprising protocol and non-filterable brute force attacks must specify some valid header fields and possibly some valid contents. For example TCP SYN attack packets cannot vary the protocol or flag field, and HTTP flood packets must belong to an established TCP connection and therefore cannot spoof source addresses, unless they hijack connections from legitimate clients.

3.4 Overview of DDoS Tools

- Attackers follow trends in the network security field and adjust their attacks to defeat current defense mechanisms. We now provide a quick overview of the several well-known DDoS attack tools in order to illustrate the variety of mechanisms deployed.

- **Trinoo** [32] is a simple tool used to launch coordinated UDP flood attacks against one or many IP addresses. The attack uses constant-size UDP packets to target random ports on the victim machine. The handler uses UDP or TCP to communicate with the agents. This channel can be encrypted and password protected as well. Trinoo does not spoof source addresses although it can easily be extended to include this capability.
- **Tribe Flood Network (TFN)** [33] can generate UDP and ICMP echo request floods, TCP SYN floods and ICMP directed broadcast (e.g., Smurf). It can spoof source IP addresses and also randomize the target ports. Communication between handlers and agents occurs exclusively through ICMP_ECHO_REPLY packets.
- **Stacheldraht** [34] combines features of Trinoo (handler/agent architecture) with those of the original TFN (ICMP/TCP/UDP flood and Smurf style attacks). It adds encryption to the communication channels between the attacker and Stacheldraht handlers. Communication is performed through TCP and ICMP packets. It allows automated update of the agents using rcp and a stolen account at some site as a cache. New program versions will have more features and different signatures to avoid detection.
- **TFN2K** [35] is the variant of TFN that includes features designed specifically to make TFN2K traffic difficult to recognize and filter. Targets are attacked via UDP, TCP SYN, ICMP_ECHO flood or Smurf attack, and the attack type can be varied during the attack. Commands are sent from the handler to the agent via TCP, UDP, ICMP, or all three at random. The command packets may be interspersed with any number of decoy packets sent to random IP addresses to avoid detection. TFN2K can forge packets that appear to come from neighboring machines. All communication between handlers and agents is encrypted and base-64 encoded.
- The **mstream** [36] tool uses spoofed TCP packets with the ACK flag set to attack the target. Communication is not encrypted and is performed through TCP and UDP packets. Access to the handler is password protected. This program has a feature not found in other DDoS tools. It informs all connected users of access, successful or not, to the handler(s) by competing parties.
- **Shaft** [37] uses TCP, ICMP or UDP flood to perform the attack, and it can deploy all three styles simultaneously. UDP is used for communication between handlers and agents, and messages are not encrypted. Shaft randomizes the source IP address and the source port in

packets. The size of packets remains fixed during the attack. A new feature is the ability to switch the handler's IP address and port during the attack.

- The **Code Red** [38] worm is self-propagating malicious code that exploits a known vulnerability in Microsoft IIS servers for propagation. It achieves a synchronized attack by preprogramming the onset and abort time of the attack, attack method and target addresses (i.e., no handler/agent architecture is involved).

4. Classification of DDoS Defence Mechanisms

- The seriousness of the DDoS problem and the increased frequency of DDoS attacks have led to the advent of numerous DDoS defense mechanisms. Some of these mechanisms address a specific kind of DDoS attack such as attacks on Web servers or authentication servers. Other approaches attempt to solve the entire generic DDoS problem. Most of the proposed approaches require certain features to achieve their peak performance, and will perform quite differently if deployed in an environment where these requirements are not met.
- We need to understand not only each existing DDoS defense approach, but also how those approaches might be combined together to effectively and completely solve the problem. The proposed classification may help us reach this goal.

4.1. Classifications by Activity Level

Preventive Mechanisms

- The goal of preventive mechanisms is either to eliminate the possibility of DDoS attacks altogether or to enable potential victims to endure the attack without denying services to legitimate clients. According to these goals we further divide preventive mechanisms into attack prevention and denial-of-service prevention mechanisms.

Attack Prevention Mechanisms

- Attack prevention mechanisms modify the system configuration to eliminate the possibility of a DDoS attack.

System security mechanisms

- Increase the overall security of the system, guarding against illegitimate accesses to the machine, removing application bugs and updating protocol installations to prevent intrusions and misuse of the system. DDoS attacks owe their power to large numbers of subverted machines that cooperatively generate the attack streams. If these machines were secured, the attackers would lose their army and the DDoS threat would then disappear.

Protocol Security Mechanisms

- Protocol security mechanisms address the problem of bad protocol design. Many protocols contain operations that are cheap for the client but expensive for the server. Such protocols can be misused to exhaust the resources of a server by initiating large numbers of simultaneous transactions. Classic misuse examples are the TCP SYN attack, the authentication server attack, and the fragmented packet attack, in which the attacker bombards the victim with malformed packet fragments forcing it to waste its resources on reassembling attempts.

Reactive Mechanisms

- Reactive mechanisms strive to alleviate the impact of an attack on the victim. In order to attain this goal they need to detect the attack and respond to it. The goal of attack detection is to detect every attempted DDoS attack as early as possible and to have a low degree of false positives.

Mechanisms with Pattern Attack Detection

- Mechanisms that deploy pattern detection store the signatures of known attacks in a database. Each communication is monitored and compared with database entries to discover occurrences of DDoS attacks. Occasionally, the database is updated with new attack signatures. The obvious drawback of this detection mechanism is that it can only detect known attacks, and it is usually helpless against new attacks or even slight variations of old attacks that cannot be matched to the stored signature. On the other hand, known attacks are easily and reliably detected, and no false positives are encountered.

Mechanisms with Anomaly Attack Detection

- Mechanisms that deploy anomaly detection have a model of normal system behaviour, such as a model of normal traffic dynamics or expected system performance. The current state of the system is periodically compared with the models to detect anomalies.

Mechanisms with Hybrid Attack Detection

- Mechanisms that deploy hybrid detection combine the pattern-based and anomaly-based detection, using data about attacks discovered through an anomaly detection mechanism to devise new attack signatures and update the database.

Mechanisms with Third-Party Attack Detection

- Mechanisms that deploy third-party detection do not handle the detection process themselves, but rely on an external message that signals the occurrence of the attack and provides attack characterization.

Agent Identification Mechanisms

- Agent identification mechanisms provide the victim with information about the identity of the machines that are performing the attack. This information can then be combined with other response approaches to alleviate the impact of the attack.

Filtering Mechanisms

- Filtering mechanisms use the characterization provided by a detection mechanism to filter out the attack stream completely.

Autonomous Mechanisms

- Autonomous mechanisms perform independent attack detection and response. They are usually deployed at a single point in the Internet and act locally. Firewalls and intrusion detection systems provide an easy example of autonomous mechanisms.

4.2. Classification by Deployment Location

Victim-Network Mechanisms

- DDoS defense mechanisms deployed at the victim network protect this network from DDoS attacks and respond to detected attacks by alleviating the impact on the victim. Historically, most defense systems were located at the victim since it suffered the greatest impact of the attack and was therefore the most motivated to sacrifice some resources for increased security.

Intermediate-Network Mechanisms

- DDoS defense mechanisms deployed at the intermediate network provide infrastructural service to a large number of Internet hosts. Victims of DDoS attacks can contact the infrastructure and request the service, possibly providing adequate compensation.

Source-Network Mechanisms

- The goal of DDoS defense mechanisms deployed at the source network is to prevent customers using this network from generating DDoS attacks. Such mechanisms are necessary and desirable, but motivation for their deployment is low since it is unclear who would pay the expenses associated with this service.

5. Conclusion

Distributed denial of service attacks is a complex and serious problem and consequently, numerous approaches have been proposed to counter them. The multitude of current attack and defense mechanisms obscures the global view of the DDoS problem. It is important to recognize and understand trends in attack technology in order to effectively and appropriately evolve defense and response strategies.

The classifications described here are intended to think about the threats we face and the measures we can use to counter those threats. We do not claim that these classifications are complete and all-encompassing. Many more attack possibilities exist and must be addressed before we can completely handle the DDoS threat, and some of them are likely to be outside the current boundaries of the classification presented here. Thus, these taxonomies are likely to require expansion and refinement as new threats and defense mechanisms are discovered. The DDoS attack and DDoS defense classifications outlined in this paper are useful to the extent that they clarify our thinking and guide us to more effective solutions to the problem of DDoS. The ultimate value of the work described here will thus be in the degree of discussion and future research that it provokes.

References:

[1] Houle K. J. and Weaver G. M., "Trends in Denial of Service Attack Technology," CERT Coordination Center, Oct. 2001.

[2] Howard J., "An Analysis of security incidents on the Internet 1989 – 1995," Carnegie Mellon University, Carnegie Institute of Technology, <http://www.cert.org/research/JHThesis/Start.html>, Apr. 1997.

[3] Mirkovic J., Martin J. and Reiher P., "A Taxonomy of DDoS Attacks and DDoS defence Mechanisms," UCLA Computer Science Department, Technical report no. 020018. <http://www.lasr.cs.ucla.edu/ddos/ucla_tech_report_020018.pdf>, 2002.

[4] Spech S. and Lee R., "Taxonomies of Distributed Denial of Service Attacks, Tools and Countermeasures," Princeton University Department of Electrical Engineering, Technical report CE-L2003-004, May 2003.

[5] CERT Coordination Center, "Denial of Service Attacks," <http://www.cert.org/tech_tips/denial_of_ service.html>, Jun 2001.

[6] Blitznet, <http://www.packetstormsecurity.org/distributed/blitznet.tgz>, 1999.

[7] DOSnet.c, <http://www.packetstormsecurity.org/distributed/DOSnet.c>, 2002.

[8] Distributed DNS Flooder v0.1b (ddnsf), <http://www.packetstormsecurity.org/distributed/ddnsf.tar.gz>, 2001.

[9] Flitz, <http://www.packetstormsecurity.org/distributed/

flitz-0.1.tgz>.
[10] Kaiten, <http://www.packetstormsecurity.org/irc/kaiten.c>.
[11] Knigth, <http://www.packetstormsecurity.org/distributed/knight.c>.
[12] Mstream, <http://www.packetstormsecurity.org/distributed/mstream.txt>.
[13] Omega v3 Beta,<http://www.packetstormsecurity.org/distributed/omegav3.tgz>.
[14] Peer-to-peer UDP Distributed Denial of Service (PUD), <http://www.packetstormsecurity.org/distributed/ pud.tgz>.
[15] Skydance v3.6, <http://www.packetstormsecurity.org/distributed/skd36.zip>.
[16] StacheldrahtV4,<http://www.packetstormsecurity.org/distributed/stachel.tgz>.
[17] Tribe Flood Network (TFN), <http://packetstormsecurity.org/groups/mixter/tfn.tgz>.
[18] Tribe FloodNet – 2k edition (TFN2k), <http://packetstormsecurity.org/distributed/tfn2k.tgz>.
[19] Trin00, <http://www.packetstormsecurity.org/distributed/trinoo.tgz>,
[20] CERT Coordination Center, "Code Red II," http://www.cert.org/incident_notes/IN-2001-09.html
[21] CERT Coordination Center, "Nimda worm," http://www.cert.org/advisories/CA-2001-26.html
[22] CERT Coordination Center, "Trends in Denial of Service Attack Technology," October 2001,
[23] CERT Coordination Center, "erkms and li0n worms," http://www.cert.org/incident_notes/IN-2001-03.html
[24] CERT Coordination Center, "Ramen worm," http://www.cert.org/incident_notes/IN-2001-01.html
[25] K. Hafner and J. Markoff, Cyberpunk: Outlaws and hackers on the computer frontier, Simon & Schuster, 1991.
[26] CERT Coordination Center, "Code Red," http://www.cert.org/incident_notes/IN-2001-08.html
[27] N. Weaver, "Warhol Worm,"http://www.cs.berkeley .edu/ ~nweaver/warhol.html
[28] CERT Coordination Center, "Trends in Denial of Service Attack Technology," October 2001, http://www.cert.org/archive/pdf/DoS_trends.pdf
[29] CERT Coordination Center, "DoS using nameservers," http://www.cert.org/incident_notes/IN-2000-04.html
[30] CERT Coordination Center, "Smurf attack," http://www.cert.org/advisories/CA-1998-01.html
[31] CERT Coordination Center, "TCP SYN flooding and IP spoofing attacks," http://www.cert.org/advisories/CA-1996-21.html
[32] D. Dittrich, "The DoS Project's 'trinoo' distributed denial of service attack tool," http://staff.washington.edu/dittrich/misc/trinoo.analysis
[33] D. Dittrich, "The 'Tribe Flood Network' distributed denial of service attack tool," http://staff.washington.edu/dittrich/misc/tfn.analysis.txt
[34] D. Dittrich, "The 'Stacheldraht' distributed denial of service attack tool," http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.txt
[35] CERT Coordination Center, "CERT Advisory CA-1999-17 Denial-Of-Service Tools," http://www.cert.org/advisories/CA-1999-17.html
[36] D. Dittrich, "The 'mstream' distributed denial of service attack tool, "http://staff.washington.edu/dittrich/misc/mstream.analysis.txt
[37] S.Dietrich, N. Long and D. Dittrich, "An Analysis of the "Shaft" distributed denial of service tool," http://www.adelphi.edu/~spock/shaft_analysis.txt
[38] CERT Coordination Center, " CERT Advisory CA-2001-19 'Code Red' Worm Exploiting Buffer Overflow In IIS Indexing Service DLL," http://www.cert.org/advisories/CA-2001-19.html