Sophistication in distributed denial-of-service attacks on the Internet

V. Anil Kumar

Denial-of-Service (DoS) attacks pose major threat to the secure and reliable operation of the Internet. This paper discusses the evolution and sophistication of different generations of DoS attacks as well as some of their common characteristics. A highly disastrous and massively distributed DoS attack in which millions of innocent Internet servers deployed at various parts of the Internet are exploited as packet reflectors to facilitate the attack is analysed. The paper is concluded by emphasizing the needs and highlighting the complexity in protecting innocent Internet servers from being exploited as intermediate launching pads of the attack against somebody.

DENIAL-of-service (DoS) attacks on the Internet are malicious attempts aiming to limit or deny service availability to legitimate users. They continue to be a major threat to the secure and reliable operation of the Internet. During the past two decades, along with the tremendous growth of the Internet, we could also observe that unethical activities on the Internet have been increasing steeply. As shown in Table 1, the statistics available from CERT (Computer Emergency Response Team) coordination centre indicates that the number of malicious incidents reported to the centre per year has increased[1] from less than 10 in the year 1988 to 114,855 in 2003. More interestingly, in an attempt to measure the prevalence of DoS attacks on the Internet, the San Diego Supercomputer Center has observed more than 12,000 backscatters (unsolicited traffic generated as a side-effect of DoS attacks) in a short span of three weeks[2] in February 2001. These figures reveal that, irrespective of the motivations behind such attacks, they are widespread on the Internet and their frequency of occurrence is also on the increase.

Before we discuss the evolution of different generation DoS attacks on the Internet, we review some of their common characteristics. First of all, they are not intended for information theft, system penetration or breaking crypto codes. A high priority item in the agenda of the attacker is to make sure that its anonymity is preserved throughout the attack. This is achieved through IP address (a unique 32 bit identification of Internet hosts) spoofing, a method by which the attacker replaces the source address of the attack packet with that of some innocent party. This also makes the instant tracing of the attacker to its physical location nearly impossible. Most of the widely seen DoS attacks are flooding attacks, where the attacker overwhelms the victim's scarce and valuable resources like CPU, memory or network bandwidth. Often, bandwidth of an organization's Internet access link is the target, because by simply exhausting this, the attacker can virtually detach an entire local network (probably with large number of computers) from the Internet. In the rest of the document, by DoS attacks we mean flooding DoS attacks.

The first-generation DoS attacks which appeared on the Internet in mid 90s were simple and designed to exploit some known vulnerabilities of the communication protocol. These attacks, in general, involve two entities: the attacker and the victim. The attacker generates flood traffic as rapidly as possible and directs it towards the victim. Of course, such attacks are effective if the attacker is at least as powerful as the victim. For example, it is not feasible for an attacker sitting behind a dial-up connection to launch the attack against well-connected targets. Examples of first-generation DoS attacks are UDP flood[3], SYN-flood[4] and ping flood attacks.

The Internet, undoubtedly, has become more robust against traditional first-generation DoS attacks. How vulnerable is a system or network to conventional attacks mainly depends on the availability of resources at the victim's side. Building blocks of the Internet have been gaining significant enhancements in their information-handling capabilities. The end-hosts are being equipped with more processing, storing and transmitting capabilities. The LAN technologies which connect the end-hosts together, the WAN technologies which integrate the LANs, and the interconnecting equipments like routers have also undergone considerable revision. As a result, conventional DoS attacks of a single attacker need not be always effective against most of today's well-connected systems.

The limitations of first-generation attacks prompted the attacker to search for novel techniques to devise new mechanisms to make the attacks more disastrous. The result is a number of more powerful second-generation attacks, commonly known as distributed denial-of-service (DDoS) attacks. They started to appear on the Internet around

V. Anil Kumar is in CSIR Centre for Mathematical Modelling and Computer Simulation, Bangalore 560 037, India
e-mail: anil@cmmacs.ernet.in

Table 1. Malicious incidents on the Internet as reported to CERT from 1988 to 2003

Year	No. of incidents	Year	No. of incidents	Year	No. of incidents	Year	No. of incidents
1988	6	1992	773	1996	2573	2000	21,756
1989	132	1993	1334	1997	2134	2001	52,658
1990	252	1994	2340	1998	3734	2002	82,094
1991	406	1995	2412	1999	9859	2003	114,855

1999. In contrast to the first-generation attacks, the DDoS attack combines the transmission power of multiple machines against a single victim, and this makes the attack more powerful. Launching such attacks involves two distinct phases. The first phase is the setting up of an attack network. The attacker scans the Internet to identify hosts with known vulnerabilities and compromises these hosts to install the attack program. The attacker along with a set of compromised machines hosting the attack program, forms an attack network. The attacker's machine or a compromised machine designated by the attacker acts as the master and the compromised machines running the attack program as slaves/zombies. The attack programs are designed in such a way that the master can remotely control them by sending instructions through the Internet. The attack network deployed once can be used at any time later, unless the zombies are rescued. In the second phase, the master, who is the coordinator of the attack, instructs the zombies to generate spoofed flood traffic towards the victim whose address is specified by the master. There are a number of readymade tools capable of launching DDoS attacks in this fashion. Trinoo[5], TFN[6], TFN2K and Stacheldraht[7] are some of the most popular members of this category. Figure 1 shows the typical architecture of a DDoS attack network.

Widespread deployment of Trinoo, TFN and Stacheldraht-based DDoS attack networks on the Internet was detected in the second half[8] of 1999. During the last three years, several powerful sites were subjected to DDoS attacks. Most noticeable are: a three-hour long attack against yahoo.com[9] on 7 February 2000, and the attack against Microsoft[10] on 14 August 2003.

While these DDoS attacks are indeed powerful and effective in paralysing even strong targets, the attackers' search for sophistication does not end here. In a recent attack, the attacker(s) demonstrated that the normal operation of innocent Internet servers serving genuine clients, can be exploited to facilitate massively distributed DoS attack without compromising the servers. These third-generation DDoS attacks are called reflector-based DDoS attacks, because during the attack the servers act as packet reflectors. Any host on the Internet which returns a packet in response to an incoming packet is a good reflector[11]. The Internet has millions of such reflectors. A typical example is TCP servers providing popular Internet services like www, e-mail, telnet, ftp, etc. The realization of the fact that TCP servers can be used to facilitate DDoS attack, is apparently a great news among the hacker community. Let us next review how a TCP server located somewhere on the Internet can be turned to a reflector for facilitating DDoS attack.

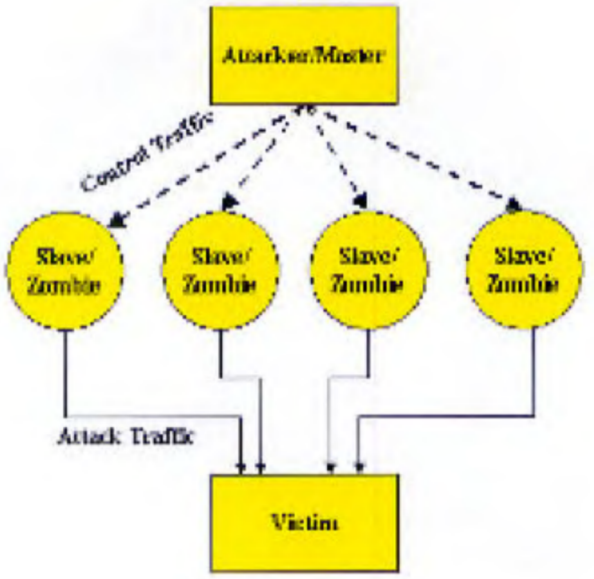

Figure 1. Architecture of distributed denial-of-service attack.

TCP is a connection-oriented transport-layer protocol responsible for byte delivery between end-systems connected through the Internet. It follows the well-known client–server architecture, where one end of the communication acts as client and the other end as server. Being a connection-oriented protocol, the client and server TCP has to first establish a logical connection prior to the beginning of the actual application data transfer. This is performed using a three-way handshake (Figure 2 *a*). The client, which needs to access service from the server, initiates the process by sending a special TCP segment (packet at transport layer) called SYN request. Through this the client informs the server its willingness to access a service from the server, and its initial sequence number (ISN) which will be used as an offset to index the subsequent application data, if any, from the client to the server. The server, upon receiving the SYN request, responds with a SYN/ACK segment containing the server's ISN and the acknowledgement to SYN of the client. In the third and final step, the client, after receiving the SYN/ACK from the server, sends the ACK segment acknowledging the SYN of the server and the connection is now established. The

three-way handshake explained above is an indispensable part of all data transfer using TCP. Web access, e-mail transfer, telnet and ftp access over the Internet are all performed only after a successful three-way handshake.

The handshake process can be easily exploited to turn an innocent TCP server to a reflector to facilitate a massive DDoS attack. The attacker (rather the zombies on behalf of the attacker; as shown in Figure 2 *b*) fabricates an SYN request in which the source address is replaced with that of the attack target and sends this request to the TCP server. The server, which is ignorant that the source address of the request is spoofed, will send a SYN/ACK segment to the target. The attacker generates and sends such spoofed requests to millions of innocent and well-behaving servers which are physically located at different parts of the Internet. The result is a massively distributed and highly intensive DoS attack against the victim. As shown in Figure 3, the zombies, instead of sending the flood directly to the victim, force the servers to flood the victim. The large number of SYN/ACK (here the attack flood) is unsolicited traffic for the victim and according to TCP specification[12], the victim will *try* to respond with RST (reset) segments to inform the sender about the arrival of unsolicited segments. However, in the case of a massive attack, since the victim's Internet access link is likely to be under flood, it is unlikely that the RST segment will reach the reflector. The RST is significant here, because an intelligent reflector, which consecutively receives RST as a response to its SYN/ACK, may be able to conclude that it is being exploited as a reflector and could stop sending new SYN/ACK responses to SYN request containing the victim's address as source. However, it is important to note that TCP specification explicitly recommends that RST segments should be used by either end of the communication to inform the other end about its unexpected unavailability to continue the ongoing communication. For example, if the client application crashes after sending the SYN, the client TCP will send an RST, which will help the server to free the resources reserved (after receiving the SYN) for serving the client. Hence arrival of RST segments after sending SYN/ACK is not completely unusual under normal operations, and cannot be taken as an exclusive signature of being used as a reflector.

While TCP three-way handshake is the easiest to exploit for reflector attack, there are other states of TCP which can also be exploited for the same purpose. For example, if a TCP server waiting for connection request receives anything other than SYN request, it will respond with RST. So the attacker can send even junk data with the source address of the victim and force the TCP server to flood the victim with RST segments. However, such attempts are more likely to be detected by the reflector in the initial stage of the attack itself. A well-configured security-monitoring mechanism like firewall or intrusion detection system in the reflector's domain may not even allow such meaningless segments to reach the server. Also, the three-way handshake used for reflection attack may act like a packet multiplier, which is an incentive to the attacker. If the reflector does not receive the RST as a response to its SYN/ACK in a stipulated period of time, it will assume that the SYN/ACK is lost and re-transmit it. This process will be repeated for a certain time, causing each SYN to trigger multiple SYN/ACK packets which will intensify the attack.

One of the first known and well-documented reflector-based attacks launched by exploiting the TCP three-way

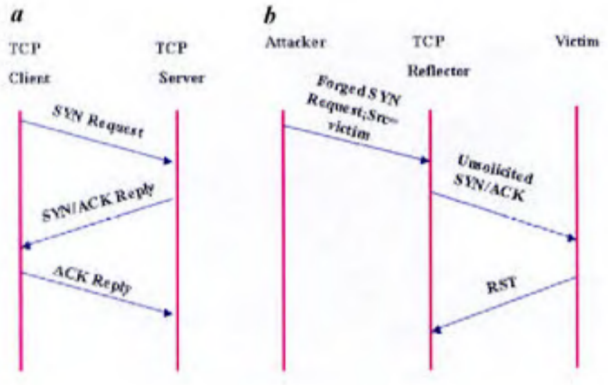

Figure 2. *a*, Normal three-way handshake; *b*, Three-way handshake during reflection attack.

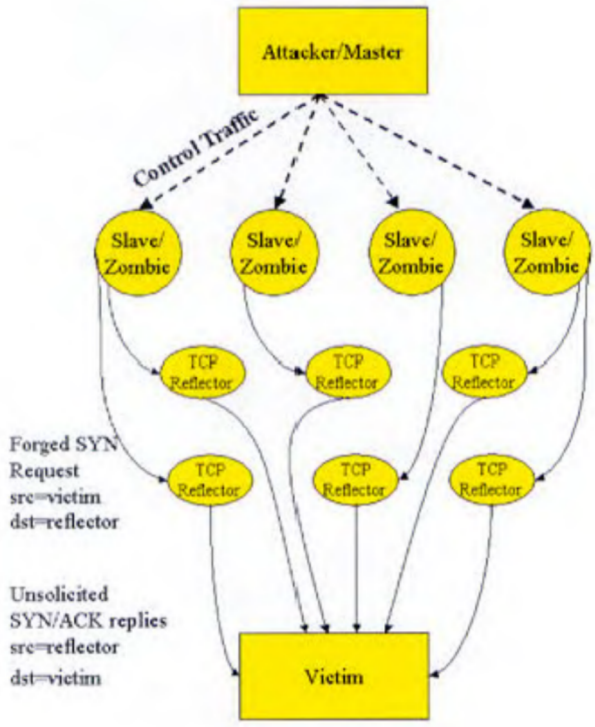

Figure 3. TCP servers acting as reflectors in massively DDoS attack.

handshake happened on 11 January 2002 at 2:00 a.m. against Gibson Research Corporation (GRC), USA[13]. The Internet access link of the company was completely flooded with SYN/ACK segments which seemed to be completely legitimate, and the victim was virtually detached from the Internet. The attack lasted for several hours, and according to the report there were more than one billion SYN/ACK packets competing with each other to flood GRC. Analysis of the forensic traces revealed that the apparent sources of the flood were hundreds of core routers and web servers of the Internet belonging to well-connected and prestigious commercial and research organizations. What really happened was, malicious hackers from somewhere on the Internet were using these well-connected TCP servers as reflectors to facilitate the attack.

While reflector-based DDoS attacks are a real nightmare to even well equipped and resource-rich organizations, vulnerability of resource-constrained networks to such attacks need not be emphasized separately. Though research has been progressing towards defending against such sophisticated attacks, effective and satisfactory solutions are yet to emerge. While protecting valuable resources from malicious attackers continues to be a difficult task, it is equally challenging, if not more, for any prestigious organization to ensure that their resources are not being exploited to facilitate such attacks without their knowledge. The emerging conclusion is that connectivity to the Internet poses two concerns: Are we the target of attackers? And are we the intermediate launching pad (of attack) against somebody?

1. CERT homepage, http://www.cert.org/stats/cert_stats.html.
2. Moore, D., Voelker, G. and Savage, S., Inferring Internet denial of service activity. 10th USENIX Security Symposium, August 2001.
3. CERT Advisory CA-1996-01 UDP port denial-of-service attack. http://www.cert.org/advisories/CA-1996-01.html.
4. Schuba, C., Krsul, I., Kuhn, M., Spafford, E., Sundaram, A. and Zamboni, D., Analysis of a denial of service attack on TCP. In proceedings of IEEE Symposium on Security and Privacy, 1997.
5. Dittrich, D., The DoS project's 'trinoo' distributed denial of service attack tool. http://staff.washington.edu/dittrich/misc/trinoo.analysis.
6. Dittrich, D., The 'tribe flood network' distributed denial of service attack tool. http://staff.washington.edu/dittrich/misc/tfn.analysis.
7. Dittrich, D., The 'Stacheldraht' distributed denial of service attack tool. http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.
8. Houle, K. J. and Weaver, G. M., Trends in denial of service attack technology. http://www.cert.org/archive/pdf/DoS_trends.pdf.
9. Williams, M., Immense network assault takes down Yahoo. http//edition.cnn.com/2000/TECH/computing/02/08/yahoo.assault.idg/
10. Roberts, P., Microsoft.com falls to DoS attack. http://www.infoworld.com/article/03/08/15/HNmsfalls_1.html.
11. Paxson, V., An analysis of using reflectors for distributed denial-of-service attacks. *ACM Comput. Commun. Rev.*, July 2001, **31**.
12. Jon Postel, Transmission control protocol. RFC 793, September 1981.
13. Steve, G., Distributed reflection denial of service. http://grc.com/dos/drdos.htm.

ACKNOWLEDGEMENT. I thank Fraunhofer FOKUS, Berlin and DAAD, Germany for providing facilities and financial support for this work. I am grateful to Dr Eckhard Moeller, FOKUS for valuable suggestions and providing logistic support.

Received 3 November 2003; accepted 28 June 2004