Global Information Assurance Certification Paper

Analyzing Distributed Denial of Service Attacks

SANS GSEC Practical version 1.4 Option 1
Val Pipenko
August 23, 2002

Abstract

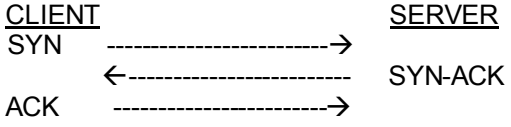
The first signs of Distributed Denial of Service (DDoS) attacks were seen in June and July 1999. Three years have passed since the first attacks, yet limited progress has been made in eliminating or containing the threat DDoS attacks pose to organizations on the Internet. The intent of this paper is to examine DDoS attacks in detail. It is imperative to realize how the attacks are committed, as well as understand the tools used to perform these attacks. This paper will build on the theory behind DDoS attack methodology and shed insight in why defending against DDOS attacks remains a problem. The paper will conclude by exploring the countermeasures available to defend your network against the DDoS threat today and in the future.

Overview

A "denial-of-service" attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service.(1) During a distributed denial of service (DDoS) attack multiple computers are used to disrupt the availability of a service from genuine users. The Internet was designed on the premise of sharing information and not security. As such even the fastest and most redundant load balanced computers and networks have limited resources that can be exhausted. During a denial of service attack a victim's computer or network resources are flooded with illegitimate requests or traffic thus preventing genuine clients from accessing available services. DDOS attacks primarily use three protocols during attempts to exhaust computer and network resources: TCP SYN flooding, UDP flooding, and ICMP flooding.

The Transmission Control Protocol or TCP is known as a connection-oriented protocol. On the Internet Web Servers, FTP Servers, Telnet Servers and Email Servers all rely on the TCP protocol to establish and maintain connections between clients requesting services and servers, which provide the service. Normal communications between clients and servers establish what is referred to as the TCP Three Way Handshake. When a client attempts an initial communication with a server an initial TCP packet is sent to the server with a TCP SYN flag set. The SYN flag is used to indicate a request to open a connection to the server. The server then responds with a SYN-ACK packet to the client, which indicates the server is ready to communicate. The server then waits for an ACK packet from the client and normal communication commences.

TCP Three Way Handshake



TCP SYN flooding involves creating thousands of half-open connections on a server. 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 what we mean by 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. (2) During a TCP SYN flood an attacker sends multiple SYN requests to a server and normally spoofs the source IP address of the packet to represent a machine that is not live on the Internet. The server responds to the SYN packets with SYN-ACK packets however sends the packet to the spoofed source address. Because the source address is spoofed the server never receives an ACK response, which it need to complete the connection and start normal communications. The flood of half open connections occupies the server's resources and prevents the server from responding to legitimate requests. An attacker normally spoofs the source IP address of the TCP SYN packets with computers that are not live on the Internet to prevent detection and also prevent live computers on the Internet from responding with a TCP RST packet. TCP RST packets would clear the attacked servers half open connections and would be sent by a computer who receives a SYN-ACK packet but never initiated a SYN request.

TCP SYN Flood

CLIENT		SERVER		SPOOFED IP
SYN	----------→	SYN-ACK	----------→	No response
SYN	----------→	SYN-ACK	----------→	No response
SYN	----------→	SYN-ACK	----------→	No response

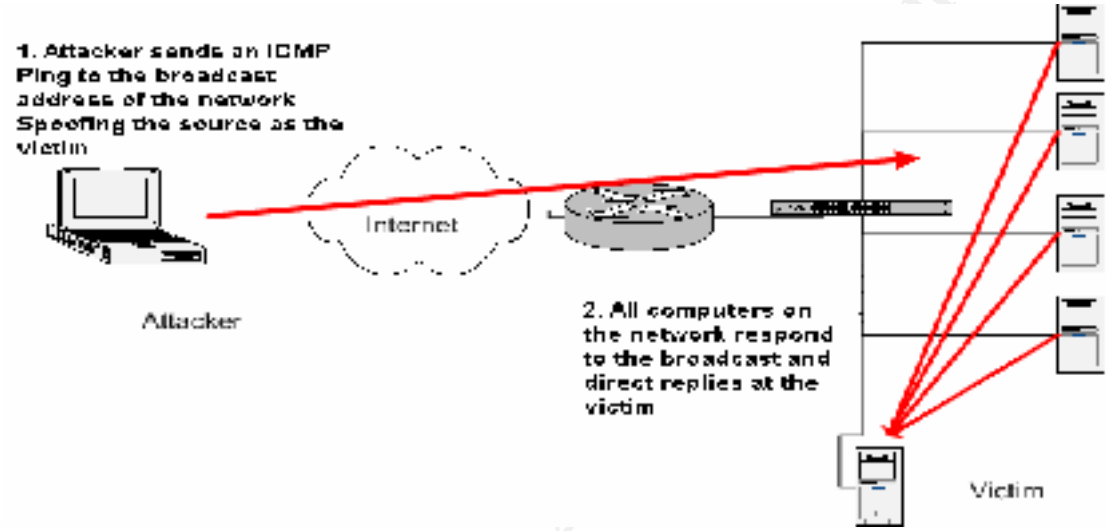
During a DDoS TCP SYN flood attack client requests are initiated from multiple computers there by amplifying the magnitude server resources are consumed.

The User Datagram Protocol or UDP is a connectionless protocol that does not require any connection procedure to transfer data. UDP Flood attacks exploit UDP services, which are known to reply to packets. A hacker is armed with a list of broadcast addresses, to which he/she sends spoofed UDP packets. Usually the packets are directed to port 7 on the target machines, which is the echo port. Other times, it is directed to the chargen port (a port that generates a number of characters when queried). Sometimes a hacker is able to set up a loop between the echo and chargen ports, generating all that much more network traffic. (3) The UDP flood attack generates large amounts of network traffic causing networks to slow to a crawl and can cause individual computers to stop responding.

The Internet Control Message Protocol or ICMP is normally used for testing and troubleshooting connectivity between hosts. During an ICMP flood attack the attacker sends a flood of ICMP ping requests at the target. By overwhelming the network with streams of ICMP packets both server resources and network bandwidth can be exhausted causing a denial of service. A variant of the ICMP flood is called the Smurf attack, which also uses the ICMP protocol.

During a Smurf attack the attacker sends spoofed ICMP ping packets directed to the broadcast address of the network. The source of the address is spoofed to appear to be originating from the victims machine. The ping to the broadcast address of the network will cause all machines on the network to flood the victims' machine with reply packets. This attack causes both the victim machine and network bandwidth to exhaust resources and limits their ability to respond to legitimate traffic.

Smurf ICMP Flood Attack



The disruption to the availability of services or networks can be but are not always the primary goal of attackers. DDoS attacks can be launched in coordination with other exploit techniques as a way of silencing or averting attention from the primary goal of the attacker. Even on there own, DDoS attacks have significant impact to business especially for those companies which rely on their Internet presence as a form of revenue generation.

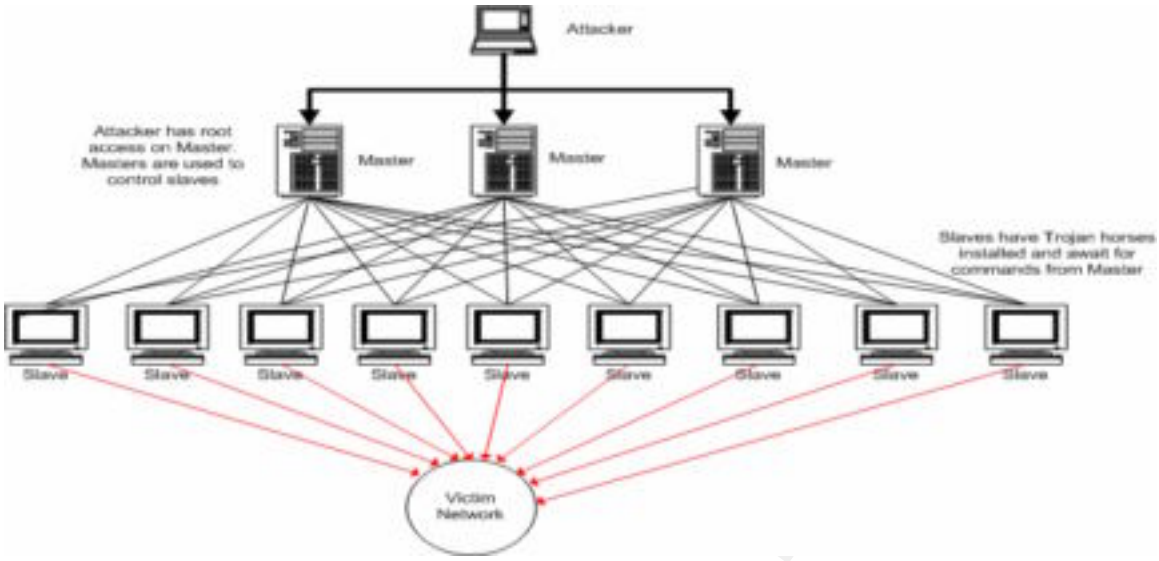
How is it perpetrated?

An attackers first step is to identify one or more systems on the Internet that can be compromised and exploited. The goal of the attacker is to first gain root access to a couple primary systems, which will then be used to gain access to other computers. The attacker wants to avoid detection and will use the initial compromised systems as a launch pad for future exploits. The initial exploits are generally accomplished using a stolen account on a system with a large number of users and/or inattentive administrators, preferably with a high-bandwidth connection to the Internet. The compromised system is loaded with any number of hacking and cracking tools such as scanners, exploit tools, operating system detectors, root kits, and DoS/DDoS programs. (4) These computers are commonly referred to as "Masters" or "Handlers" because they will be used to control the computers, which will perpetrate the DDoS attack.

The second step in a DDOS attack involves recruiting small armies of computers, which the attacker will later use to perpetrate the DDOS attacks. The

attacker must be able to gain control of hundreds of computers, which is commonly done with the installation of Trojan horse software. The term Trojan horse comes from the Trojan War where the Greeks gave a large wooden horse to the citizen of Troy. The wooden horse was just a disguise and secretly contained warriors that conquered the citizens of Troy after they emerged from the wooden horse at night. An attacker uses a Trojan horse very similar to the ones the Greeks used. In DDoS attacks the term Trojan horse refers to a piece of software, which secretly contains malicious code. When the software is executed, remote control software like Sub Seven, Tribal Flood Network, or Trinoo is silently installed in the background of unsuspecting Internet users. Trojan horses once installed on a computer can give the attacker almost complete control of the resources of the infected system. Trojan horses infect computers in one of two methods. First, Trojan horses can be emailed to unsuspecting people and in the form of a joke executable or program that infects the system once the application is launched. The second method of infection is much stealthier as it does not require a user to actually launch an application. An attacker will commonly use a port and vulnerability scanner to generate a list of computers with known buffer overflow exploits. A scan is performed on large ranges of network blocks to identify potential targets. Targets would include Unix or Linux systems running various services known to have remotely exploitable buffer overflow security bugs, such as wu-ftpd, RPC services for "cmsd", "statd", "ttdbserverd", "amd", etc. (4) If the attacker is attempting to install a Windows based Trojan horse the attacker may scan networks that belong to AOL. By scanning AOL network address ranges the attacker will likely find hundreds if not thousands of windows computers with default installations, improperly patched and configured systems. Once the attacker has generated a list of systems with a specific vulnerability, the attacker can use a script to install the Trojan on the identified target. After the Trojan is installed the attacker normally takes steps to hide evidence that the system has been compromised. Altering system logs and replacing programs like netstat with a custom program that will hide the fact that new communications ports are open on a system. These computers are commonly referred to as “Slaves” or “Zombies”. Slave computers run a process referred to as a daemon, which is responsible for carrying out the commands issued by the “Master” or “Handler”.

An attacker is now armed with a list of compromised systems in his army and is able to launch a coordinated distributed denial or service attack at any victim on the Internet.



Tools Used to Perpetrate DDOS

The following section is a technical description of four commonly used Trojan horses attackers use during DDoS attacks.

Trinoo

The trinoo network is comprised of an attacker connecting to a computer that has the trinoo master software installed on it. An attacker would instruct the trinoo master to carry out a DDoS attack on a specified IP address or addresses for a specified period of time. The Trinoo master will instruct the trinoo daemons (slaves) to launch the DDoS.

Trinoo Ports
27665/tcp > Destination port the attacker uses to communicate with the master.
27444/udp > Destination port the master uses to communicate with the daemon.
31335/udp > Destination port the daemon uses to communicate with the master.
Random > Daemons use random UDP ports to DDoS the victim.

When the trinoo daemon is initialized on a slave system it will have the IP address of its master or masters. The trinoo daemon will notify the master it is available by sending out a UDP packet containing the string “*HELLO*”. Trinoo masters keep track of daemons by keeping an encrypted list of compromised systems in the same directory as the master binary file. Trinoo masters are able to send a broadcast request to all daemons on the network. Daemons receiving the broadcast respond to the master with a UDP packet containing the string "PONG". (6) Communications between the attacker and the master on TCP port 27665 are password protected. Communications between the master and the daemon on UDP port 27444 require the packet to contain the string “l44”. Although trinoo requires password to communicate and run the binary, all communication are in clear text. Trinoo currently does not spoof the source address during a UDP DDoS attack, but this capability could be built in.

Trinoo default passwords:

"l44adsl"	trinoo daemon password
"gOrave"	trinoo master server startup ("?? " prompt)
"betaalmostdone"	trinoo master remote interface password

List of names Trinoo daemon is installed under in Linux and Unix environments:

ns	rpc.listen	irix
http	trinix	
rpc.trinoo	rpc.irix	

Trinoo has also been ported to the windows operating system. Infected windows systems will contain the follow file and registry key:

C:\WINDOWS\SYSTEM\service.exe

HKLM\Software\Microsoft\Windows\CurrentVersion\Run\"System Services"="service.exe"

TFN

Tribal Flood Network (TFN) is made up of a client binary called “tribe.c” and daemon binary called “td.c”. An attacker will communicate with multiple clients which intern instructs daemons to launch DDoS attacks on specified victims. An attacker is able to control system with the client software install through remote shell bound to a TCP port, UDP based client/server remote shells, ICMP based client/server shells such as LOKI, SSH terminal sessions, or normal "telnet" TCP terminal sessions. The “tribe.c” client software does not require a password but you do need root access on the system. The client software tracks systems that have been exploited with daemon software through a file called “iplist”. The client software uses ICMP_ECHOREPLY packets to communicate with daemons. Daemons do not communicate with clients in a TFN network.

Command syntax for TFN (7)

usage: ./tfn <iplist> <type> [ip] [port]

<iplist>	contains a list of numerical hosts that are ready to flood
<type>	-1 for spoofmask type (specify 0-3), -2 for packet size, is 0 for stop/status, 1 for udp, 2 for syn, 3 for icmp, 4 to bind a rootshell (specify port) 5 to smurf, first ip is target, further ips are broadcasts
[ip]	target ip[s], separated by @ if more than one
[**port]**	must be given for a syn flood, 0 = RANDOM

A newer version of TFN called TFN2K is also propagating. The new version has also been ported to windows systems.

Stacheldraht

Stacheldraht comes from the German meaning “Barbed Wire”. A Stacheldraht network is compromised of a master (handler) and slave (daemon), which are similar to the TFN and Trinoo network architectures. Stacheldraht is capable of unleashing TCP SYN flooding, UDP flooding, ICMP flooding and smurf DDoS attacks. Stacheldraht differs from its predecessors by employing an encrypted communications medium from the attacker to the master (handler) using a program named “telnet alike”. The Stacheldraht master runs the “mserv.c” binary aswell as the “telnetc” binary. The “mserv.c” is responsible for communications to the slaves while the “telnet.c” will listen for connections from the attacker. An attacker would load a binary called “client.c” on there systems in order to connect to the master(s) “telnet.c” binary. “client.c” and “telnet.c” are the two components of the “telnet alike” program that enables encrypted communications between the attacker and master(s). The slaves (daemons) run a binary called “leaf/td.c” which listen for instructions from the masters. The mserv.c code has a limit of 1000 slaves that it is able to control. Communication in a Stacheldraht network use TCP and ICMP ports between attackers, masters, and slaves.

Stacheldraht Ports

16660/tcp> Destination port the attacker uses to communicate with the master.
65000/tcp, ICMP_ECHOREPLY> Port the master uses to communicate with the daemon.
65000/tcp, ICMP_ECHOREPLY > Port the daemon uses to communicate with the master.
Random > Daemons use random UDP, TCP, and ICMP ports to DDoS the victim.

Stacheldraht Commands (8)

.**distro user server**
Instructs the agent to install and run a new copy of itself
using the Berkeley "rcp" command, on the system "server",
using the account "user" (e.g., "rcp user@server:linux.bin ttymon")

.help Prints a list of supported commands.

.killall Kills all active agents.

.**madd ip1[:ip2[:ipN]]** Add IP addresses to list of attack victims.

.mdie Sends die request to all agents.

.mdos Begins DoS attack.

.**micmp ip1[:ip2[:ipN]]** Begin ICMP flood attack against specified hosts.

.**mlist** List IP addresses of hosts being DoS attacked at the moment.

.mping Pings all agents (bcasts) to see if they are alive.

.**msadd** Adds a new master server (handler) to the list of available servers.

.msort Sort out dead/alive agents (bcasts). (Sends pings and shows counts/percentage of live agents).

.**mstop ip1[:ip2[:ipN]] all** Stop attacking specific IP addresses, or all.

.msrem Removes a master server (handler) from the list of available servers.

.msyn ip1[:ip2[:ipN]] Begin SYN flood attack against specified hosts.

.**mtimer seconds** Set timer for attack duration. (No checks on this value.)

.**mudp ip1[:ip2[:ipN]]** Begin UDP flood attack against specified hosts.

.setisize Sets size of ICMP packets for flooding. (max:1024, default:1024).

.setusize Sets size of UDP packets for flooding (max:1024, default:1024).

.showalive Shows all "alive" agents (bcasts).

.showdead Shows all "dead" agents (bcasts).

.sprange lowport-highport Sets the range of ports for SYN flooding

Stacheldraht uses a password to connect masters to slaves; the default is “sicken”. One feature of stacheldraht not shared by trinoo or TFN is the ability to upgrade the agents on demand. This feature employs the Berkeley "rcp" command (514/tcp), using a stolen account at some site as a cache. On demand, all agents are instructed to delete the current program image, go out and get a new copy (either Linux- or Solaris-specific binary) from a site/account using "rcp", start running this new image with "nohup", and then exit. (9)

Problems associated with defending against DDOS

Regardless of how well defended your assets may be, your susceptibility to many types of attacks, particularly DDoS attacks, depends on the state of security on the rest of the global Internet. (10) The Internet connects millions of weak or improperly secured systems, which attackers can use to launch attacks at even the most secure network environments. With no geographical boundaries and an abundance of computers that have permanent Internet connections and are rarely secured, attackers have the tipped the scale of resources in their favor.

Attack technology is also evolving at a rapid rate. Open source code and collaborative thinking from around the world is creating more sophisticated attack tools. Trinoo, one of the first attack tools used UDP flooding to target victims. From there TFN progressed to use TCP,UDP, ICMP and Smurf attacks. Even more evolution is found in Stacheldraht in the form of encrypted communications between the attack and masters.

The lack of deterrent to commit cyber crimes posses a problem for organizations trying to defend against DDoS attacks. Because of limited legislation and authority to deal with a crime that can be committed from

anywhere in the world, the apprehension and prosecution of an attacker is improbable. Even with the US Patriot Act, a bill that extends the rights and tools used by US authorities to protect the American I.T. infrastructure. Attackers are able to exploit American companies from countries that have no legal or political ties to the United States which enforcement of US law impossible.

With the increased use of the Internet in daily business, a scarcity is forming in properly trained system administrators. The average level of system administrator technical competence has decreased dramatically in the last 5 years as non-technical people are pressed into service as system administrators.(11) The growing dependency of business to be on the internet coupled with newbie system administrator creates a recipe for opportunity to mis-configure systems and expose a company to Internet attacks.

Countermeasures today

DDoS attack tools frequently employee IP address spoofing in an attempt to evade tracking and hide the identities of the computers carrying out the attack. Some DDoS attacks like the Smurf attack require the attacker to spoof the source of the packet as the victims IP address. Restricting the ability of an attacker to disguise the origin of an attack will not prevent all attacks from happening but will drastically shorten the time required to locate the attackers. By using Egress filtering on perimeter routers of private networks and Internet Service Providers, the Internet community can insure packets arriving on their network came from the source specified in the packet. Egress filtering insures that only addresses that are assigned to a specific network can be routed from that network. It is also important to ensure that packets attempting to leave a site with a non-routable source address as those listed in RFC 1918 should not be permitted to route through the Internet. Egress filtering is only effective if collaboratively deployed by the Internet community. Deploying Egress filtering at your site can ensure that your systems are not used as slaves in a DDOS attack on another organization.

During a Smurf attack, broadcast amplification is used in conjunction with IP spoofing during the attack. An attacker generates packets (ICMP echo_request) with the source address of the victim and then sends those packets to the network address of a large network. Every system on that network then floods the victim with responses. Echo and chargen ports are often used in a similar UDP attack by creating a loop between the echo and chargen ports of two systems. By disabling the forwarding of directed broadcast and multicast traffic from your network you will protect your system from these attacks. Chargen and echo services should also be disabled on systems unless there is a specific need for the service, as goes for any service that is not needed.

Many organizations do not respond to complaints of attacks originating from their sites or to attacks against their sites, or respond in a haphazard manner. This makes containment and eradication of attacks difficult. Further, many organizations fail to share information about attacks, giving the attacker

community the advantage of better intelligence sharing. (12) Private organizations need to develop policies, which outline how Internet threats should be addressed. The collaboration and sharing of information amongst the private sector, vendors, ISP's, and authorities are essential to in the prevention of future attacks and suppression minimization of current threats.

Unpatched and mis-configured systems remain one of the biggest challenges in defending against DDoS attacks. Vulnerable systems produce an abundant resource for attacker to prey on. By deploying "best practice" and well-known system hardening techniques, DDoS tools can be prevented from being installed on systems.

1) System administrator can begin by scanning all Internet connected computers for DDoS software and other malicious software.
2) Antivirus programs should be kept up dated on all system to help prevent infection.
3) Security patch levels on all machines, firewalls, routers and other Internet accessible equipment should be kept up to date.
4) Unneeded services should be turned off all machines.
5) Organizations should deploy a multi-layered approach to security including Firewalls with restrictive rule bases, and Intrusion Detection Systems both host and network based.
6) Default security settings that ship from most vendors create many avenue of attack. Hardening scripts should be used by administrator to ensure production and internet connected systems are not susceptible to exploitation.

Countermeasures tomorrow

The future of controlling and even eliminating DDoS attacks lies in the advances of three areas, protocol developments and research and analysis.

The development of new protocols will help address some of the inadequacies attackers exploit during DDoS attacks. ICMP trace (itrace) is a proposed protocol to help trace the origins of packets being forwarded by routers. Using the itrace protocol, routers would sample a few packets of the total traffic being passed and add a "Traceback" message that would be forwarded to the final destination. The proposed itrace protocol could help eliminate the dilemma of locating the real source of spoofed packets with the "Traceback" field attached to the packet. A possible problem with itrace is probability. Itrace uses a mathematical equation to randomly sample router traffic and attach the "Traceback" field. It is possible that during an attack legitimate packets get tagged with the "Traceback" field while the attackers packets get forwarded by the router untouched.

IPSEC, which stands for Internet protocol security, is currently being used in virtual private network architecture. IPSEC adds the ability to authenticate a packets origin or AH authentication header. IPSEC also uses ESP encapsulating security payload, which can also ensure the integrity of the packet as well as encrypt the packet between two points. The adoption of IPSEC into all Internet communications could insure the authentication of the origin of every packet on the Internet. The use of IPSEC today comes with the price of additional overhead

to accommodate the information in the larger IPSEC packets. IPSEC communications also require a manual setup between two peers (normal VPN peers) before communication take place. Future incorporation of IPSEC into everyday traffic could prevent spoofing in DDoS attacks.

Pushback is designed to detect and control high bandwidth aggregates in the network. An aggregate is a collection of packets with a common property. For instance, with the destination prefix as the common property, all packets with a matching prefix define an aggregate. (13) Routers using the pushback protocol could control floods of traffic in a DDoS attack by limiting the traffic to the specific destination (the aggregate). The router could also request upstream routers to limit the amount of traffic sent to the specific destination. Pushback only employees a rate limiting mechanism instead of blocking the traffic, in an effort to allow legitimate traffic in to the destination. During a DDoS attack pushback will attempt to notify routers far enough back the upstream traffic to prevent the attacks at their source.

IPv6 is the next generation of Internet Protocol, which is intended to replace the current IPv4. IPv6 will incorporate features that will address many of the deficiencies with IPv4. IPv6 will use IPSEC to ensure authentication of packets, which will prevent spoofing of addresses. Congestion control features are all built into IPv6, which can be used to thwart the bottlenecks caused by DDoS attacks. Another idea under discussion is ingress filtering at the customer edge of ISPs. By enforcing topological correctness at the first hop, the attacker will be unable to spoof their source addresses. (14) There is currently a lot of pressure and effort being focused on adopting IPv6.

Conclusion

Distributed denial of service attacks remain a complex and serious problem. This paper has examined DDoS attacks; the method attackers use, as well as the tools the attackers are armed with. By examining DDoS in depth, a clear view of the problems associated with defending against DDoS has been seen. No silver bullet or security-spending budget exists to completely eliminate the threat of DDoS attacks from any organization. What is known is all organizations on the Internet are interdependent on each other when defending against DDoS attacks. Combating DDoS is a complex issue that requires a cooperative effort by the entire Internet community. Prospects to minimize the impact of DDoS attacks are feasible for the future with continued research, and protocol development. It is my belief that if attackers can cooperate to develop Trojans and launch DDoS attacks, Internet users can do the same to defend against them. It is every Internet users and organizations responsibility to be a good NET-citizen.

References

(1) CERT Coordination Center, “Denial of Service Attacks” Feb 12, 1999
URL: http://www.cert.org/tech_tips/denial_of_service.html

(2) CERT Coordination Center, “Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks” Nov. 29, 2000
URL: http://www.cert.org/advisories/CA-1996-21.html

(3) SAINT Corporation, “Packet Flooding Problems” Jan 1, 2002
URL: http://www.wwdsi.com/demo/saint_tutorials/packet_flooding_problems.html

(4) Advanced Networking Management Lab (ANML) “Distributed Denial of Service Attacks(DDoS) Resources” June 17,2002
URL: http://www.anml.iu.edu/ddos/howtowork.html

(5) David Dittrich “The DoS Project's "trinoo" distributed denial of service attack tool” Oct 21, 1999
URL: http://staff.washington.edu/dittrich/misc/trinoo.analysis.txt

(6) CERT Coordination Center, “Incident Note IN-99-07” January 15, 2001
URL: http://www.cert.org/incident_notes/IN-99-07.html

(7) David Dittrich, “The "Tribe Flood Network" distributed denial of service attack tool” Oct 21, 1999
URL: http://staff.washington.edu/dittrich/misc/tfn.analysis.txt

(8) David Dittrich, “The "stacheldraht" distributed denial of service attack tool” Dec 31, 1999
URL: http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.txt

(9) David Dittrich, “The "stacheldraht" distributed denial of service attack tool” Dec 31, 1999
URL: http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.txt

(10) Kevin J. Houle, CERT/CC, George M. Weaver, CERT/CC “Trends in Denial of Service Attack Technology” October 2001
URL: http://www.cert.org/archive/pdf/DoS_trends.pdf

(11) Kevin J. Houle, CERT/CC, George M. Weaver, CERT/CC “Trends in Denial of Service Attack Technology” October 2001
URL: http://www.cert.org/archive/pdf/DoS_trends.pdf

(12) Rich Pethia, Alan Paller, Gene Spafford “Consensus Roadmap for Defeating Distributed Denial of Service Attacks A Project of the Partnership for Critical Infrastructure Security Version 1.10” February 23, 2000
URL: http://www.sans.org/ddos_roadmap.htm

(13) Sally Floyd, Steve Bellovin, John Ioannidis, Kireeti Kompella, Ratul Mahajan, Vern Paxson “Pushback Messages for Controlling Aggregates in the Network” July, 2001
URL: http://www.icir.org/floyd/papers/draft-floyd-pushback-messages-00.txt

(14) Allen Householder, Art Manion, Linda Pesante,George M. Weaver “Managing the Threat of Denial-of-Service Attacks” October 2001
URL: http://www.cert.org/archive/pdf/Managing_DoS.pdf