



DDoS: In Depth - by badpack3t

Date: Sunday, December 14 @ 14:59:58 EST

Topic: Security

Distributed Denial of Service Attacks have recently emerged as one of the most newsworthy, if not the greatest, weaknesses of the Internet. Overview Distributed Denial of Service (DDoS) attacks are a relatively new development; reports of the first DDoS attacks surfaced in mid-1999, with the highest-profile attacks coming in early 2000 against sites like Amazon.com, CNN.com, eBay and E-Trade. Clearly, the challenge these attacks present is a serious one. While you alone can't do much to protect yourself, as a community we can improve the situation. The victims were unreachable for several hours each. A brief note on usage: the network where these attacks are taking place is called the "Internet", with a capital "I"; it is the public network shared by people all over the world. An "internet", with a lower-case "i", is a collection of networks interconnected; many organizations have private internets. The Internet is the result of inter-connecting a gigantic number of private internets.

The advent of DDoS marked an escalation in Internet Relay Chat (IRC) wars. Relying on networks of linked servers, IRC offers channels, or chat rooms, that users can join to exchange ideas, pictures, sounds, and programs. Channel operator (ruling) status is assigned by default to a channel's creator, to someone who inherits channel operator privileges, or to someone who simply asks for it (assuming there is no current channel operator).

Explanation of DDoS attacks

DDoS attacks involve breaking into hundreds or thousands of machines all over the Internet. Then the attacker installs DDoS programs on them, allowing them to control all these exploited machines to launch coordinated attacks on victim sites. These attacks typically exhaust bandwidth, router processing capacity, or network stack resources, breaking network connectivity to the victims.

Hacker (or, if you prefer, Cracker) starts by breaking into weakly secured computers, using well-known exploits in standard network service programs, and common weak configurations in operating systems. On each system, once they break in, they perform some additional steps. First, they install software to conceal the fact of the break-in, and to hide the traces of their subsequent activity. For example, the standard commands for displaying running processes are replaced with versions that fail to display the attacker's processes. These replacement tools are collectively called a "rootkit", since they are installed once you have "cracked root", taken over system administrator privileges, to keep other "root users" from being able to find you. Then they install a special process, used to remotely-control the burgled machine. This process accepts commands from over the Internet, and in response to those commands it launches an attack over the Internet against some designated victim's site. And finally, they can have there so called "Bots or Zombies" report to private chat rooms on IRC. A cautious hacker will begin by breaking into just a few sites, then using them to break into some more, and repeating this cycle for several

steps, to reduce the chance they are caught during this, the riskiest part of the operation. By the time they are ready to mount the kind of attacks we've seen recently they have taken over thousands of machines and assembled them into a DDoS network; this just means they all have the attack software installed on them, and the attacker knows all their addresses.

Now its time for the attack. The attacker runs a single command, which sends command packets to all the captured machines, instructing them to launch a particular attack against a specific victim. When the attacker decides to stop the attack, they send another single command.

The packets used in today's DDoS attacks use forged (or Spoofed) source addresses; they are lying about where the packet comes from. The very first router to receive the packet can very easily catch the lie; it has to know what addresses lie on every network attached to it, so that it can correctly route packets to them. If a packet arrives, and the source address doesn't match the network it's coming from, the router should dump the packet. This style of packet checking is called variously Ingress or Egress filtering, depending on the point of view; it is Egress from the customer network, or Ingress to the heart of the Internet. If the packet is allowed past the border, catching the lie is nearly impossible. Returning to our analogy, if you hand a letter to a letter-carrier who delivers to your home, there's a good chance he could notice if the return address is not your own. If you deposit a letter in the corner letter-box, the mail gets handled in sacks, and routed via high-volume automated sorters; it will never again get the close and individual attention required to make any intelligent judgments about the accuracy of the return address. Likewise with forged source addresses on internet packets: let them past the first border router, and they are unlikely to be detected.

Today there's no possibility of performing more than a few back-traces at most, in as little as a few hours. Even that would require some luck to favor your efforts. So as long as the attacker turns their attack off after at most a few hours, you are unlikely to find more than a few of the thousands of machines used to launch the attack; the remainder will remain available for further attacks. And the compromised machines that are found will contain no evidence that can be used to locate the original attacker; your trace will stop with them.

Tools of the Trade

Many tools are available to perpetrate DDoS attacks. Because source code is available for a number of these tools, many of the findings about a particular set of DDoS tools change over time. In fact, the characteristics that are seen "in the wild" often do not match those seen by analysis of the available source code. DDoS tools typically follow a three-tier architecture, known as a DDoS constellation. The attacker (controlling console) is used to issue commands to the master controller layer. The master controllers are then responsible for controlling a given number of agents that do the actual labor of the attack. The attacker can control a large number of masters, and each master can control a large number of agents. Since any traceback of flooding traffic to ascertain the source of the attack will result in an agent system, finding the master controllers is very difficult, and finding the attacker consoles is even more difficult.

There are basically five methods of attack that are supported by known DDoS tools:

- Smurf -- ICMP (Internet Control Message Protocol) ping requests to a directed broadcast address. The forged source address of the request is the target of the attack. The recipients of the directed broadcast ping request respond to the request and flood the target's network.
- ICMP flood -- Similar to Smurf, but without the amplification caused by requests to a directed broadcast address.
- UDP flood -- Sending large numbers of UDP (User Datagram Protocol) packets to the target system, thus tying up network resources.
- TCP flood -- Sending large numbers of TCP packets to the target system, thus tying up network resources.
- TCP SYN flood -- Sending large numbers of TCP connection initiation requests to the target. The target system must consume resources to keep track of these partially opened connections.

The most prominently seen DDoS tools vary by their methods of attack, communication between master and agents, and the system privileges needed to execute an attack. The more recent and sophisticated DDoS tools even come with functionality to update software automatically, easing the burden of running a large DDoS constellation. Seven families of DDoS tools have been seen in the wild. The more common families are trinoo, Tribe Flood Network (TFN and TFN2K) and Stacheldraht.

Trinoo, an early DDoS tool, is relatively unsophisticated by current standards. It initiates only a UDP flood attack. Communication between the master and agents uses unencrypted TCP and UDP. Root/administrator privileges are not needed to use trinoo. This means that any regular user can deploy a trinoo constellation without having to compromise a systems administration account. Given trinoo's relative simplicity, it is easier to detect and combat than more recently developed tools.

TFN and TFN2K use multiple attack types, including UDP, ICMP and TCP SYN floods. It can also emulate a Smurf attack. Communication between the master and the agents uses ICMP_ECHOREPLY packets. Commands and arguments are sent as part of the ICMP ID field and in the data portion of the packets. The main difference between TFN2K and TFN is that the agent is silent in TFN2K, making it more difficult to detect. The master sends multiple commands to the agent and relies on the probability that at least one will get through. In addition, the command packets are mixed with a number of decoy packets sent to random destinations. As TFN evolves, it becomes easier to cause outages and more difficult to detect. TFN and TFN2K are more difficult to deploy than trinoo, because they require root or administrator privileges on the system running the agent.

Like TFN, Stacheldraht has multiple attack options, including UDP, ICMP, TCP SYN and broadcast ping floods. Its use of ICMP_ECHOREPLY is similar to TFN's, but Stacheldraht can encrypt the console-to-master TCP session. Stacheldraht also has an auto-update feature. Like TFN and TFN2K, Stacheldraht requires root or admin privileges on the system running the agent as well as the master.

Key Trends and Factors

The recent attacks against e-commerce sites demonstrate the opportunities that attackers now have because of several Internet trends and related

factors:

- Attack technology is developing in an open-source environment and is evolving rapidly. Technology producers, system administrators, and users are improving their ability to react to emerging problems, but they are behind and significant damage to systems and infrastructure can occur before effective defenses can be implemented. As long as defensive strategies are reactionary, this situation will worsen.
- Currently, there are tens of thousands - perhaps even millions - of systems with weak security connected to the Internet. Attackers are (and will) compromising these machines and building attack networks. Attack technology takes advantage of the power of the Internet to exploit its own weaknesses and overcome defenses.
- Increasingly complex software is being written by programmers who have no training in writing secure code and are working in organizations that sacrifice the safety of their clients for speed to market. This complex software is then being deployed in security-critical environments and applications, to the detriment of all users.
- User demand for new software features instead of safety, coupled with industry response to that demand, has resulted in software that is increasingly supportive of subversion, computer viruses, data theft, and other malicious acts.
- Because of the scope and variety of the Internet, changing any particular piece of technology usually cannot eliminate newly emerging problems; broad community action is required. While point solutions can help dampen the effects of attacks, robust solutions will come only with concentrated effort over several years.
- The explosion in use of the Internet is straining our scarce technical talent. 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. Additionally, there has been little organized support of higher education programs that can train and produce new scientists and educators with meaningful experience and expertise in this emerging discipline.
- The evolution of attack technology and the deployment of attack tools transcend geography and national boundaries. Solutions must be international in scope.
- The difficulty of criminal investigation of cybercrime coupled with the complexity of international law mean that successful apprehension and prosecution of computer crime is unlikely, and thus little deterrent value is realized.

- The number of directly connected homes, schools, libraries and other venues without trained system administration and security staff is rapidly increasing. These "always-on, rarely-protected" systems allow attackers to continue to add new systems to their arsenal of captured weapons.

Resource Consumption

An intruder may also be able to consume all the available bandwidth on your network by generating a large number of packets directed to your network. Typically, these packets are ICMP ECHO packets, but in principle they may be anything. Further, the intruder need not be operating from a single machine; he may be able to coordinate or co-opt several machines on different networks to achieve the same effect.

In addition to network bandwidth, intruders may be able to consume other resources that your systems need in order to operate. For example, in many systems, a limited number of data structures are available to hold process information (process identifiers, process table entries, process slots, etc.). An intruder may be able to consume these data structures by writing a simple program or script that does nothing but repeatedly create copies of itself. Many modern operating systems have quota facilities to protect against this problem, but not all do. Further, even if the process table is not filled, the CPU may be consumed by a large number of processes and the associated time spent switching between processes. Consult your operating system vendor or operating system manuals for details on available quota facilities for your system.

Security Considerations

The primary intent of this document is to inherently increase security practices and awareness for the Internet community as a whole; as more Internet Providers and corporate network administrators implement ingress filtering, the opportunity for an attacker to use forged source addresses as an attack methodology will significantly lessen. Tracking the source of an attack is simplified when the source is more likely to be "valid." By reducing the number and frequency of attacks in the Internet as a whole, there will be more resources for tracking the attacks which ultimately do occur.

Thoughts

On closing, I just wanted to make some comments regarding security. Try to subscribe to a couple of security alert digests so that you are alerted to new exploits and try to keep up on bugs that effect your systems (SANS, CERT, and SecurityFocus.com (Bugtraq) Security-Protocols.com are a few good security sites with digests) and visit your operating system's site for current information regarding your specific system. As for the research done, I have really enjoyed it and learned a lot about DoS and DDoS.

Thanks,

badpack3t

founder

www.security-protocols.com

References:

Cert. "Denial of Service Attacks." June, 2001.

Url: http://www.cert.org/tech_tips/denial_of_service.html

Staff Washington Education. DDoS - Is There Really a Threat? 1998

Url: <http://staff.washington.edu/dittrich/talks/sec2000/>

Secure Computing. "Analysis and Partial Solutions."

Url: <http://www.securecomputing.com/index.cfm?sKey=416>

CERT's stacheldraht advisory, CA-99-17

Url: www.cert.org/advisories/CA-99-17-denial-of-service-tools.html

SANS Global Incident Analysis Center

Url: http://www.sans.org/ddos_roadmap.htm

RFC 2267. "Defeating Denial of Service Attacks which Employ IP Source Address Spoofing,"

Url: www.landfield.com/rfc/rfc2267.html

This article comes from .: [Security-Protocols] .:

<http://security-protocols.com>

The URL for this story is:

<http://security-protocols.com/article.php?sid=1657>