# Denial-Of-Service attacks

Article URL
Denial-Of-Service attacks

Author
SecurityHome.eu

Published: 12 September 2006

Last updated on 22 May 2007.

Â

There is nothing you can do to prevent or stop D.O.S.-attacks, but there are some things you can do to make it a bit harder.

First you need to understand how D.O.S.-Attacks works.
The goal of D.O.S.-attacks is to take down your server (computer) or, if that is not possible, make it look like your server is down by taking all the bandwidth of your network (internet) connection. There are several different kind of attacks:

* TCP SYN Flooding
* ping Flooding
* UDP flooding
* ICMP Redirect Bombs
* Other attacks

TCP SYN Flooding

A TCP SYN flooding attack consumes you system resources until no more incoming TCP connection sare

possible. The attack makes use of the basic TCP three-way handshaking protocol during connection establishment, in conjunction with IP address spoofing.

The attacker spoofs his or her source address and initiates a connection to one of your TCP-based services. As a client attempting to open a TCP connection, the attacker sends you a SYN message. Your machine responds by sending an acknowledgment, a SYN-ACK. However, in this case the address you're replying to isn't the attacker's address. It's a nonexistent address. The final stage of the TCP connection establishment, receiving an ACK in response, will never happen. Consequently, finite network connection resources are consumed. The connection remains in a half-opened state until the connection attempts times out. The hacker floods your port with connection request, faster then the TCP timeouts release the resources. If this continues, all resources will be in use and no more incoming connection requests can be accepted. If the target is your smtp port, you can't receive email. If the target is your http-port, people can't connect to your site.

Several aids are available to Linux users. The first is the source address filtering. This filters out the most commonly used spoofed source address. The second is to compile your kernel with SYN cookies enabled; this is a specific retardant to SYN flooding (default in RedHat 6.0).

ping Flooding

Any message that elicits a response from your machine can be used to degrade your network connection by forcing the system to spend most of its time responding. The ICMP echo request message sent by ping is a common culprit.

Additionally, an older exploit called the Ping of Death involved sending very large ping packets. Vulnerable systems could crash as a result. Linux is not vulnerable to this exploit, nor many other current UNIX operating systems.

ping is very useful, basic networking tool. You might not want to disable ping altogether. In today's internet environment, conservative folks recommend disabling incoming ping, or at least severely limiting whom you accept echo requests from.

Because of ping's history of involvement in denial-of-service attacks, many sites no longer respond to external ping requests.

UDP Flooding

The UDP protocol is especially useful as a denial-of-service tool. Unlike TCP, UDP is stateless. Flow control mechanisms aren't included. There are no connection state flags. Datagram sequence numbers aren't used. No information is maintained on which packet is expected next. It's relative easy to keep a system so busy responding to incoming UDP probes that no bandwidth is left for legitimate network traffic.

Because UDP services are inherently less secure than TCP service, many sites disable all UDP ports that aren't absolutely necessary. Almost all common Internet services are TCP-based.

ICMP Redirect Bombs

ICMP redirect message type 5 tells the target system to change its routing tables in favor of shorter route. If you run routed or gated and honor redirect messages, it's possible for a hacker to fool your system into

thinking that the hacker's machine is one of your local machines or one of your ISP's machines, or even fool your system into forwarding all traffic to some other remote host.

Other attacks

Network connectivity isn't the only concern in denial-of-service attacks. Here are some examples:

* Your file system can overflow if your system is forced to write enormous numbers of messages to the error logs, or if your system is flooded with many copies of large email messages. You may want to configure resource limits and set up a separated partition for rapidly growing or changing file systems.

* Server can crash if large amounts of data are sent to them and overflow the input buffers, or if unexpected data is sent to them. CGI scripts are especially vulnerable unless you take precautions. Many of the current vulnerabilities in servers are due to buffer overflows. It's important to keep up-to-date and install all the newest patches and software revisions.

* System memory, process table slots, CPU cycles, and other resources can be exhausted by repeated, rapid invocations of network services. There's little you can do about this other than setting any configurable limits for each individual service, enabling SYN cookies, and denying rather than rejecting packets sent to unsupported service ports.

What can you do?

First you should install a firewall
When you configure the firewall you should use deny

instead of reject

. The difference? With reject you send a message back saying 'This is not allowed', or something. With deny nothing is sended back.
I you use Linux you should enable SYN-cookies. Standard in RedHat 6.0 and above.
Limit the services run on your server.
Closing all the ports that you don't use. (TCP, UDP and ICMP).
You shouldn't see any IP's of these ranges:

* Class A 10.0.0.0 - 10.255.255.255 - Private addresses

* Class B 172.16.0.0 - 172.31.255.255 - Private addresses

* Class C 192.168.0.0 - 192.168.255.255 - Private addresses

* Class D 224.0.0.0 - 239.255.255.255 - Multicast IP-addresses (Multicast networks Broadband, audio - video cast)

* Class E 240.0.0.0 - 247.255.255.255 - Set aside for future and experimental use

* lo 127.0.0.0 - 127.255.255.255 - Private network used by UNIX (and other OS) for local, network based services

To prevent that you disk gets full by error-message and emails, you should make a separate partition for this. In linux I recommend that /var

and /home

even /tmp

are on a separated partition from the root /

In windows you can limited the file size of your log files. Don't make them to small, you need the information to figure out what happend.