Fighting DDoS (2):
Types of Simple DoS

by M. E. Kabay, PhD, CISSP
Associate Professor, Computer Information Systems
Norwich University, Northfield VT

In the first article of this series, we looked at the importance of denial-of-service (DoS) attacks. In this article, I present a brief review of some basic types of simple DoS techniques.

* * *

Denial of service (DoS) is hardly new. Unplugging a computer system or burning it to cinders causes denial of service; more interesting, so does holding down the ENTER key on a terminal that is not yet logged on but is connected to certain kinds of mainframe and minicomputers. This latter case is instructive: the reason that such a simple act can bring these computers to a halt is that pressing the ENTER key initiates a device-recognition routine in the operating system that runs at high priority; keeping the key pressed generates so much processing to recognize the device that the process can easily consume almost 100% of the processor capacity.

The key (no pun intended) to understanding this DoS is that an action has caused an unexpectedly high resource consumption because the operating system function is unable to distinguish a legitimate demand (asking to initiate a logon) from an abusive demand (keeping the ENTER key pressed). In this case, the operating system function involved is _stateless_; i.e., there is no mechanism to change the response as a function of whether the stimulus has already occurred in a period of time that would allow the system to recognize abuse.

Instead of using up all the CPU cycles, other kinds of DoS attack saturate other fixed resources. For example, in a normal session initiation, a system on the Internet initiates a request for a connection to a specific host using a SYN (synchronization) packet. The host responds with a SYN-ACK (synchronization acknowledgement) packet to the initiating system's IP address and sets an entry in a table for pending session initiations with a reasonable timeout parameter (originally around 15 seconds). Finally, in a normal connection, the requesting system responds with and ACK (acknowledgement) packet to complete the session initiation. In the SYN flooding attack, the attacker sends as many SYN packets as it can and uses one or more false IP addresses these fraudulent SYN packets. The SYN-ACK responses go into the bit bucket and the host never receives any ACK packets for the pending sessions. Therefore the session-initiation table quickly fills up to its maximum and no further requests for new sessions -- including legitimate requests -- can be processed until the SYN flood terminates. Two of the methods for reducing susceptibility to such attacks were (1) to increase the size of the session-initiation table and (2) to reduce the timeout period to free up entries more quickly.

A similar saturation attack was perpetrated on bulletin board systems (BBSs) back in the 1980s when abusive posters sent hundreds or thousands of messages to the board and used up all available slots for new messages and sometimes completely replaced all the old ones, entirely destroying the message base. System operators responded by configuring strict limits on how many messages a given user could post per day, whereupon the abusers began using multiple aliases to mask the origin of the spurious messages.

Another nasty denial of service attack is possible when repeated logon errors cause an ID to be locked out of a system or network but there are no delays imposed before allowing another logon. By logging on to every ID in turn and deliberately entering invalid passwords, a single criminal hacker can shut down access to all the IDs on a system. Even with a delay, it is still possible to interfere with access in this manner if the attacker uses a script or program for automatically trying all IDs quickly.

Finally, another class of DoS attacks relies on crashing systems. In 1981, for example, I found that typing a particular parameter in a logon to an MPE III operating system would crash any HP3000 in the world even without having a legitimate logon ID. The Teardrop attack is another example of causing DoS through a system crash: Teardrop attacks use malformed packets to cause the IP process stack on the target machine to crash. Similarly, buffer overflows can also cause system crashes (buffer overflows are problems in which long input strings are not blocked by an input edit function before they are processed by the operating system or application program).

* * *

In the next article in this series, we will look at distributed denial-of-service attack methods.

* * *

M. E. Kabay, PhD, CISSP is Associate Professor in the Department of Computer Information Systems at Norwich University in Northfield, VT. Mich can be reached by e-mail at < mkabay@compuserve.com >. He invites inquiries about his information security and operations management courses and consulting services.