DOS ATTACKS - A PRESENT DANGER

DOS Attacks are what are known as Denial-of-Service attacks and are aimed at denying legitimate users access to services or resources. This can be done out of pure spite, or can be the precursor for some more sophisticated attack. While these attacks generally require very little technical skill, they are very difficult to defend against or counteract, simply because many of them use normal traffic and because the attacker is usually not worried about any response, the source addresses can be spoofed. Lets go through some of the more common types..

Buffer Overflow
All programs need buffers in memory into which they can accept input, and the programmer makes assumptions about how much space is needed. An overflow occurs when more data then what is expected is received and the program does not know what to do about it. These could be oversized ICMP packets -the "Ping of Death"- overlong user names, or even SMB logon requests with wrongly identified data size.

SYN Attacks
This attack exploits the usage of the machines buffer during the 3-way handshake when setting up a normal TCP connection. The receiving machine tracks SYN packets in a queue for processing connection requests. A SYN attack floods this buffer with connection requests thus stopping it from processing any other connection requests. You can combat this by increasing the buffer size or decreasing the time before state entries are purged.

Teardrop Attack
You must first understand what fragmentation is, this is when a router receives a packet which is too large for any of it's attached segments. It then fragments the packet, breaking it down into manageable pieces, and then sending it on. The fragmentation field in the packet is used so that the receiving system can correctly reassemble the packet. Messing with the correct functioning of this process -whether by false fragmentation field settings, bad reassembly flags, etc- can cause an error condition, which could be unexpected and could thus possibly cause the receiving system to crash

Smurf and Fraggle Attacks
The *Smurf* attacks uses both spoofing and ICMP replies to cause excessive to the victim system. The attacker sends a spoofed echo request to the broadcast address of a large network, the echo request has the spoofed source address of the attackers intended victim. Now all hosts on the targeted network send echo replies back to the victim, and if the attacker chose a large enough network -the amplification network- then the victims system is flooded with echo reply packets. You can prevent partaking in this type of attack by preventing spoofing with perimeter sanity checks. The *Fraggle* is a *Smurf* attack but it uses UDP rather than TCP.

Land Attack
This attacks also uses spoofing, but this time the attacker crafts a packet which has the same source and destination addresses. The purpose of this is to try to trick the victim machine into entering into an infinite loop by responding to itself. If this happens then all available resources will be quickly exhausted and no more connections will be able to be processed.

ICMP Flood
This attack is very simply a constant and steady high volume flow of echo request packets, the attacker tries to use all the bandwidth the victim system has. While implementing such an attack is easy, it is generally only useful if the attacker has the same, or more, bandwidth as the victim. The solution is also very simple, just disallow any ICMP traffic across your network perimeter or if you must allow it, then limit the amount of traffic you can receive.

TCP Connect Flooding
This attack uses a full 3-way connection to open up as many TCP connections as possible to the victim's machine. This is done in an attempt to consume all the possible sockets available on the victim system, and thus prevent the processing of proper connections.

Bandwidth Consumption
Many of the mentioned attacks actually accomplish this, but this attack can also be accomplished just as well by the simple expedient of sending more data then what the victim has bandwidth -email messages, web requests, etc-, basically a high volume of normal looking traffic. Attacker can use zombie-nets to accomplish this, the zombies are many compromised victims who will follow the commands of the attacker.

Distributed Denial-of-Service Attacks
These are like normal DOS attacks except that the attack is being launched from multiple infected hosts. The original attacker will infect machines with a client program of some sort which will allow them to control the victim, when they want to launch a DDOS attack, the attacker tells all the infected clients to each launch a DOS attack on the main victim system, this means that the main victim can receive multiple individual DOS attacks. Some incidents have shown attackers using infected network numbering in the thousands.

Distributed Reflective Denial-of-Service
Here an attacker takes advantage of the normal mode of operations of critical Internet functionalities such as DNS, routers, etc. The attacker sends out update data to the core ISP and service centers with a spoofed source of the victim system. When the systems respond to the spoofed update to creates a high-speed, high-volume flood. The nasty thing about this attack, is that if the victim bans the source, they are effectively cutting themselves off from entire segments of the Internet.

Coordinated DOS Attack
Unlike the rest of the attacks, this is not really an attack but it has the same effect as a normal DOS attack. This involves many people all doing the same thing at the same time while using the certain service or system. A common occurrence is known as being "slashdotted", this is when a websites address is given in an article on the popular slashdot website, so all the readers go to have a look and unintentionally DOS the "victim" system.

Unintentional DOS Attack
This is another occurrence of an event that is not usually malicious but yet still has the same effect as a normal DOS attack. This is when some resource is misconfigured in such a way that it causes a loss of service to a company. Think of a technician putting in a router which has not been setup right, it could cause the company to suffer network outages. While this may not be caused intentionally, I personally think it is just as bad to attempt to do something badly.

...and guess what else? It just got worse. With the prevalence of wireless networks, attackers have come up with several new ways to DOS wireless systems, add to this the fact that due to the nature of the RF medium, you actually cannot protect against Layer 1 or Layer 2 wireless DOS attacks. Take a look at these..

Jamming Attacks
As mentioned you can actually do nothing about this attack except try to physically track done the device causing it and stop it. The attack can be done by a normal machine with a wireless card flooding the chosen channels with bogus frames and data. Or it could be a custom built device used to generate frequencies in the 2.4 Ghz range, this can be done using something like a microwave oven's magnetron even.

Spoofed Deassociation or Deaunthentication Frames
Here the attacker sends out deassociation or deauthentication frames to the clients with the spoofed mac address of the AP, if these packets are sent out fast enough, then no-one will be able to use the wireless network because they could not connect for long enough. Again, there is nothing you can do to prevent this other then to keep a careful eye on your network and take preventative measures when something happens.

Spoofed Malformed Authentication Frames
Here the attacker sends a malformed authentication frame to the access point with the spoofed source address of the victim, this causes the access point to send an error back to the victim whom the attacker impersonated. This error causes the client to be disassociated from the wireless network, and can cause other problems when reconnecting.

Overflowing Access Point Buffers
There are many access points that do not have sufficient protection against buffer overflow attacks, and on these access points the attacker can cause them to become unresponsive by creating a flood of established connection or authentication requests. Once this has been successfully done, the device can no longer process other legitimate requests.

EAP DOS Attacks
A wireless network using EAP can be susceptible to a large number of DOS attacks which all revolve around messing with the EAP protocol frames. This includes flooding the network with EAPOL-Start frames, sending out premature EAP Success frames, using malformed EAP frames, etc. Stopping the proper functioning of EAP will - of course- have a negative impact on the functioning of the wireless network.

Is there nothing I can do?
Well, most of the major IT vendors have realized the risk the DOS attacks pose and each have varying options which can be set to help deal with the threat..

Vendor	*Settings*
Cisco	verify unicast reverse-path no ip directed broadcast
Windows	HKLM\system\currentcontrolset\services\tcpip\parameters\ TcpTimedWaitDelay - set to - 96 KeepAliveTime - set to - 30000 TcpMaxHalfOpen - set to - 100 TcpMaxPortsExhausted - set to - 1 TcpMaxHalfOpenRetried - set to - 80 TcpMaxDataRetransmissions - set to - 3 TcpMaxConnectResponseRetransmissions - set to - 2 EnableDeadGWDetect - set to - 0 EnablePMTUBHDetect - set to - 0 EnableICMPRedirects - set to - 0 EnableSecurityFilters - set to - 1 DisableIPSourceRouting - set to - 1 SynAttackProtect - set to - 2 HKLM\system\currentcontrolset\services\afd\ DynamicBacklogGrowthDelta - set to - 10 EnableDynamicBacklog - set to - 1 MaximumDynamicBacklog - set to - 2000 MinimumDynamicBacklog - set to - 20 Interfaces\{InterfaceID} PerformRouterDiscovery - set to - 0
Linux	Use IPTables - use the *-m limit* iptables module to limit connections echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter echo 1 > /proc/sys/net/ipv4/tcp_syncookies echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts echo 120 > /proc/sys/net/ipv4/vs/timeout_synack ##decrease as needed echo 15 > /proc/sys/net/ipv4/vs/timeout_synrecv ##decrease as needed echo 180 > /proc/sys/net/ipv4/tcp_max_syn_backlog ##increase as needed