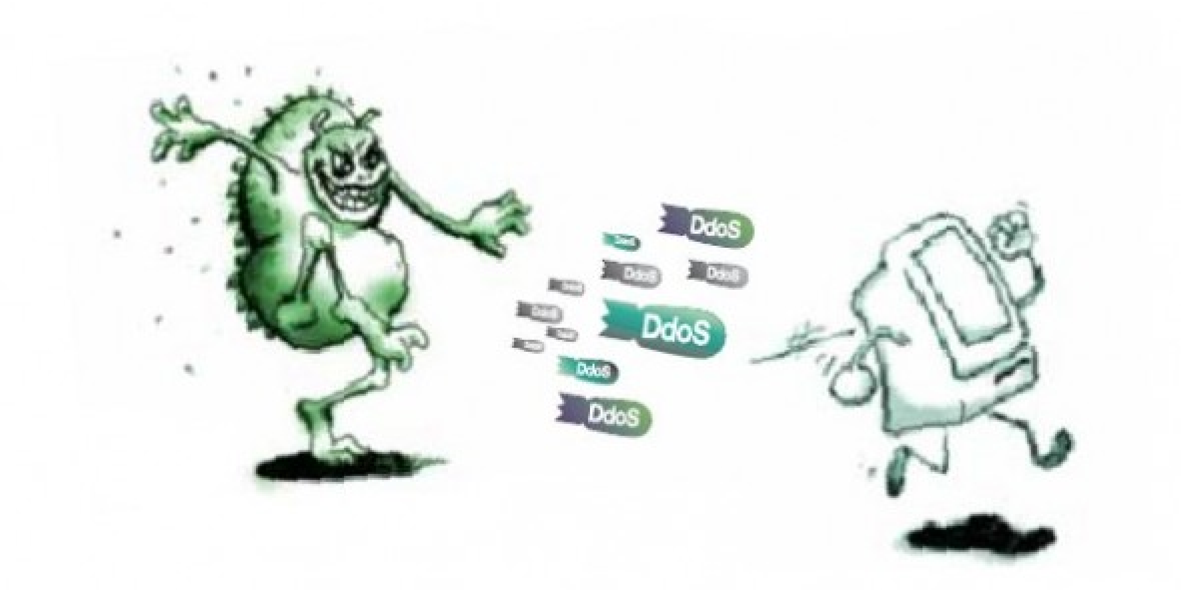
CYBER ATTACKS EXPLAINED: DOS AND DDOS



With this article, we begin a new series on the major kinds of cyber attacks that weaken the IT security infrastructure within organisations. With the rapid spread of Internet technologies and applications, the number of those seeking to break into systems is also increasing — usually to gain fame, money, or to damage the target's reputation. The first to be covered in this series is the Denial of Service attack (DoS) and the distributed version (DDoS). We will learn how these attacks work technically, and discuss ways to stop them at the network entry point.

The fundamental technique behind a DoS attack is to make the target system busy. In a computer server, when a network packet is being received, all components (right from the network interface card or NIC to the application running under the OS) are participating to ensure successful delivery of that packet. The NIC must monitor the Ethernet frames meant for it, align data and pass it to the network card driver, which then adds its own intelligence and passes it to the OS, which takes it to the application.

Each component involved can exhibit some form of vulnerability, and DoS attacks are devised to exploit one or more of these, to penetrate into the system.

Let's now understand the basics of the TCP/IP protocol, which uses a handshake between the sender and receiver. Figure 1 shows how a healthy TCP handshake takes place, and how a SYN flood attack compares with it.

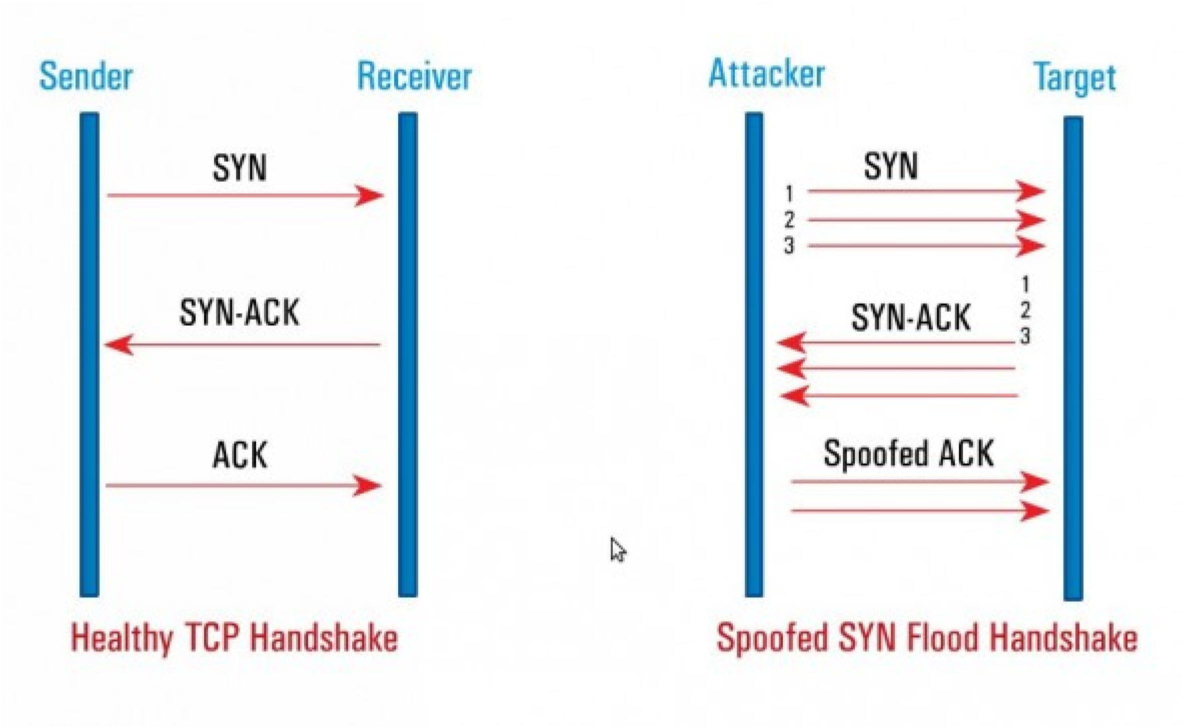


Figure 1: A healthy TCP handshake

When the sender wants to communicate, it sends a SYN packet with its own IP address as the source, and the receiver's IP address as the destination. The receiver responds with a SYN-ACK packet. The sender confirms this by sending an ACK packet. Now, sender and receiver have a guarantee that they can communicate with each other.

The sender then starts sending the actual data in small chunks, and for each data packet received, the receiver sends an ACK back. When the sender sends the final data chunk, it sends a FIN signal, which is acknowledged by the receiver by sending a FIN-ACK back.

If a particular port is not supposed to respond to the request, the receiver responds with an RST packet, which means it is rejecting that request.

As you can see here, the TCP/IP stack software has to deal with complex communication, which does take some CPU and memory resources. To add to this, many handshakes are happening on a server for different source and destination addresses and for various TCP ports. All IP-based protocols such as ICMP ping, telnet, FTP, etc, actually piggyback on this framework to do their job, each working on a different dedicated socket or port.

At the application layers, the OS and the application receiving or transmitting data allocate internal memory buffers and a software process to keep track of what is being sent or received. The OS partially does this job itself, and leaves the rest to the network driver and protocol stack. Each process consumes some CPU time and memory resources.

A DoS attack exploits this situation, by tweaking TCP packets to make the server respond to malicious, fabricated network requests. TCP packets can be forged, or modified to disrupt the basic handshake process, in order to create unexpected network responses. This ultimately results in exhausting all the server resources, which when overwhelmed, stops responding.

There are various ways to do this, each using a different technical approach. Please refer to the following table — it shows you how various DoS techniques map to the OSI model of network layers.

Application	Web DoS, Email Spam
Presentation	Malformed SSL Requests
Session	Telnet DDoS
Transport	SYN Flood, Smurf Attack
Network	ICMP Flooding
Data	MAC Flooding
Physical	Dummy Packet Attack

We will now discuss each of these techniques in more detail.

Network layer DoS attacks

The MAC flood

This is a rare Layer 1 attack, in which the attacker sends multiple dummy Ethernet frames, each with a different MAC address. Network switches treat MAC addresses separately, and hence reserve some resources for each request. When all the memory in a switch is used up, it either shuts down or becomes unresponsive.

In a few types of routers, a MAC flood attack may cause these to drop their entire routing table, thus
disrupting the whole network under its routing domain.

The SYN flood

The attacker sends multiple SYN packets; upon receiving SYN-ACK from the target, it does not send ACK, but instead sends more SYN packets. This leaves the TCP/IP stack on the target to conclude that there is a possible network congestion or disconnect, and it waits for a specific amount of time. Thus, multiple partially-open connections are maintained by the stack for some time, in anticipation of an ACK response.

In another SYN flooding type, the SYN is sent with a spoofed source address, which becomes the destination address for the target’s SYN-ACK packet. However, since the system at the spoofed IP never sent that SYN to begin with, it will never send an ACK to this packet, and will simply drop it. The target system is not aware of this, and keeps track of it in anticipation of an ACK.

In both examples, the connection tables and memory resources on the targeted network components are filled up with bogus entries. When the entire table is filled up, the device stops responding.

The ping of death

In this case, a malformed ping packet flood is sent to the target. Since the TCP stack responds only to a certain type of ping packet, it fails to respond to this, exhausting the system resources.

TCP established connection attacks

This is an extension to the SYN flood, the difference being that it uses a complete 3-way TCP handshake. It does not spoof addresses, nor strip the ACK responses, but just establishes a TCP connection and simply does not send any data. Due to this, connections are maintained till time-out; a flood of such connections results in a DoS on the targeted device.

Since the handshake is gracefully done, this attack is difficult to trace, and needs advanced techniques to detect and stop.

Smurf attacks

This is a famous, widely used Layer-3 attack in which the attacker sends ping traffic to IP broadcast addresses. However, the source address is spoofed, and is of the victims' machines. Thus, routers deliver replies to the victims, which send back with a ping response. On a larger and populated subnet, this can have a devastating effect, and can practically render the routing device non-responsive.

Similar to this is the Fraggle attack, wherein a UDP echo packet is sent instead of TCP

TCP RST attacks

In this new breed of DoS attack, the source IP is spoofed with the victim's IP address, and this malformed packet is sent to a firewall. This forces the firewall to remember this connection for some time.

This attack is rarely used; its sole purpose is to fool the intrusion detection logic on cheaper firewalls. Unlike the host desktop, a firewall with UTM features works in promiscuous mode, and takes note of each and every packet. However, if the anomaly detection logic is not smart enough, it simply results in piling up connections and eventually rendering itself useless.

Application layer DoS attacks

Besides these network-layer attacks, there are a few that deal with the application layer directly. Such attacks are comparatively easy to detect and fix, but if ignored, can result in heavy downtime.

Application-layer attacks exploit vulnerabilities in the OS and the guest application. Listed below are a few such attacks.

Buffer overflows

This very well-known attack pushes the OS to consume resources to such an extent that it starts leaking memory, becomes sluggish or simply stops responding. It is a myth that a buffer overflow is observed only in Windows OS — in fact, it is very much true for Linux distros too. Applications such as a database, email and Web servers are found to have buffer overflow vulnerabilities.

Web and DNS DoS attacks

Web servers running on TCP port 80 are a common target for DoS attacks. Attackers usually send multiple HTTP requests (not malformed at all) targeted to retrieve enormous amounts of data from the backend database server.

Such request floods make the database server busy, keeping the Web server waiting for data. This creates a pile-up on both servers, which become unresponsive to further requests. This can happen unintentionally too, especially when breaking news is posted and everyone tries to access it at the same time. In case of DNS servers, the attack method is similar to Web servers, but it has serious consequences. If a DNS server becomes non-responsive, it can take down the firm’s entire network.

Distributed DoS attacks

In simple terms, a distributed DoS attack (DDoS) combines multiple attackers using the various techniques discussed above, and can result in catastrophic failure. Please refer to Figure 2 to understand how a typical DDoS attack is planned on a website.

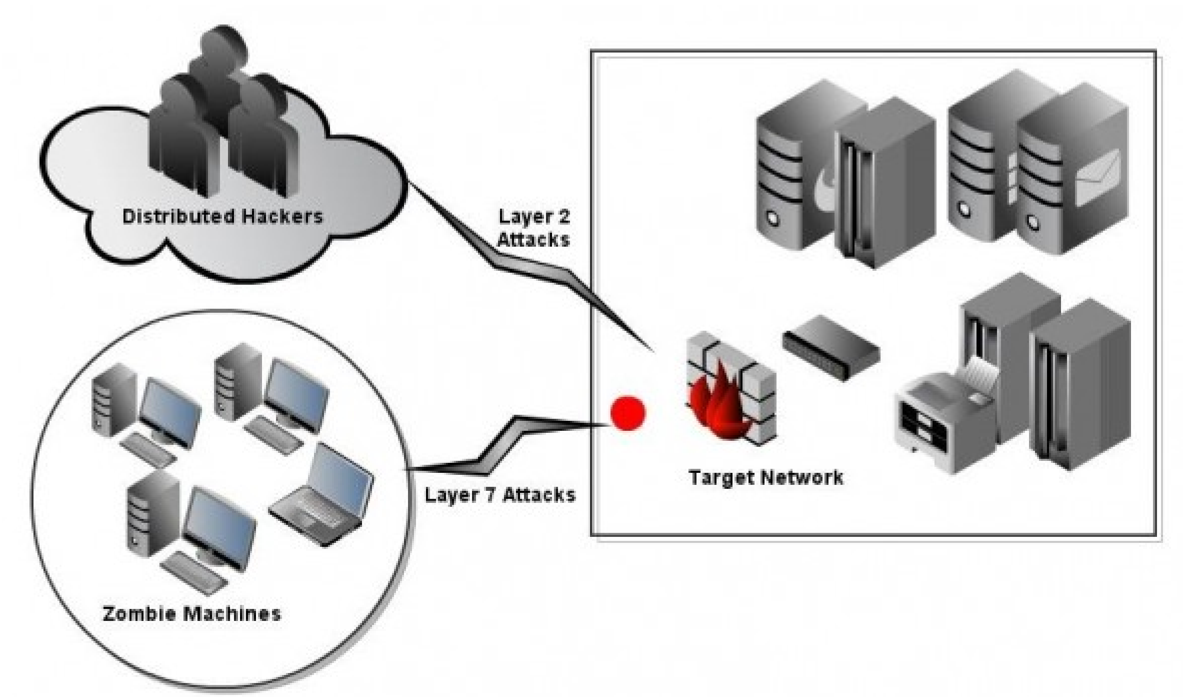


Figure 2: A typical DDoS attack

A lethal combination of spoofed SYN flood and old-style ping-of-death attacks is typically used to disrupt an IT network that is open to the Internet.

In a modern form of DDoS, the attacker injects malicious code into a virus and spreads it to millions of computers, making them zombie machines. On a specific day and time, all infected machines start executing that code, which is usually written to access a website or plant a network-level attack onto a targeted system. Since it is difficult to find out who injected the code into a virus, it is difficult to trace the real attacker.

In another non-hazardous form, a DDoS attack is modified to access the attacker’s website and click advertisements on it, which in turn helps the attacker earn money from ad clicks.

Protecting FOSS systems

Though DoS is one of the oldest and most commonly known attacks, unfortunately, there is no fool-proof solution to stop it because practically, it is difficult to decide which network connection is legitimate, and which one is initiating an attack.

While there are specific tools for a particular type of DoS attack, it boils down to cyber-security design and monitoring to strengthen the network.

Typical symptoms of a DoS attack on a Linux server are a sluggish system or a slow website, sudden and prolonged increase in processor and memory utilisation, excessive disk thrashing without any business activity, slower file transfers, etc.

On a network monitoring system, there could be a large number of TCP packet drops, abnormal TCP resets, broken TCP SYN packets being received, or duplicate ACK packets being sent. Usually, the first-level component impacted is a router, followed by a firewall and other components like switches. Firewalls cannot protect the router, but the firmware of modern routers (e.g., Cisco 7600 or X443) contain patches to protect against DoS attacks. Though modern firewalls have many features to combat DoS attacks, they don’t always help.

Firewalls can certainly protect against network layer-based attacks, but usually fail to protect systems from application-layer attacks like on port 80 (HTTP). Here, an application-level firewall is needed to filter each request and ensure its legitimacy.

For Ubuntu and RHEL, an APF (Advanced Policy Firewall) is a great tool that can help mitigate DoS attacks. Linux FOSS systems are blessed with fantastic network drivers, as well as many built-in features such as a packet-filtering firewall, packet monitoring, network monitoring tools, kernel hardening tools, etc.

For smaller Linux networks, a nice script can be written to SYN Trap open connections and to stop bogus TCP RST connections, as a first line of defence.

For mission-critical corporate Linux networks, deploying an Intrusion Prevention System device (IPS) is the best choice. IPS devices sit on a network in promiscuous mode and use built-in anomaly detection algorithms to intercept and decode each and every packet. Since its intelligence ranges from Layer 2 to Layer 7, it can gauge which packet is legitimate, and which isn’t. It has a great alerting mechanism
to proactively inform and stop DoS and DDoS attacks.

A good combination of IPS devices, UTM firewalls and application layer security can help stop these dreaded attacks.

Source : http://www.opensourceforu.com/2011/11/cyber-attacks-explained-dos-and-ddos/