Hackers have been carrying out **distributed denial-of-service (DDoS)** attacks for more than a decade, and their potency steadily has increased over time.

The Nature of Denial-of-Service Attacks :-

Denial of service is a form of attack on the availability of some service. In the context of computer and communications security, the focus is generally on network services that are attacked over their network connection. We distinguish this form of attack on availability from other attacks, such as the classic acts of god, that cause damage or destruction of IT infrastructure and consequent loss of service.

From this definition, you can see that there are several categories of resources that could be attacked :-

- Network bandwidth
- System resources
- Application resources.

Flooding Attacks :-

Flooding attacks take a variety of forms, based on which network protocol is being used to implement the attack. In all cases the intent is generally to overload the network capacity on some link to a server. The attack may alternatively aim to overload the server's ability to handle and respond to this traffic. These attacks flood the network link to the server with a torrent of malicious packets competing with, and usually overwhelming, valid traffic flowing to the server. In response to the congestion this causes in some routers on the path to the targeted server, many packets will be dropped. Valid traffic has a low probability of surviving discard caused by this flood and hence of accessing the server. This results in the server's ability to respond to network requests being either severely degraded or failing entirely Common flooding attacks use any of the ICMP, UDP, or TCP SYN packet types. It is even possible to flood with some other IP packet type. However, as these are less common and their usage more targeted, it is easier to filter for them and hence hinder or block such attacks.

ICMP Flood :-

This type of ICMP packet was chosen since traditionally network administrators allowed such packets into their networks, as ping is a useful network diagnostic tool. More recently, many organizations have restricted the ability of these packets to pass through their firewalls. In response, attackers have started using other ICMP packet types.

An attacker can generate large volumes of one of these packet types. Because these packets include part of some notional erroneous packet that supposedly caused the error being reported, they can be made comparatively large, increasing their effectiveness in flooding the link.

UDP Flood :-

An alternative to using ICMP packets is to use UDP packets directed to some port number, and hence potential service, on the target system. A common choice was a packet directed at the diagnostic echo service, commonly enabled on many server systems by default.

If the server had this service running, it would respond with a UDP packet back to the claimed source containing the original packet data contents. If the service is not running, then the packet is discarded, and possibly an ICMP destination unreachable packet is returned to the sender. By then the attack has already achieved its goal of occupying capacity on the link to the server. Just about any UDP port number can be used for this end. Any packets generated in response only serve to increase the load on the server and its network links.

TCP SYN Flood :-

Another alternative is to send TCP packets to the target system. Most likely these would be normal TCP connection requests, with either real or spoofed source addresses. They would have an effect similar to the SYN spoofing attack we have described. In this case, though, it is the total volume of packets that is the aim of the attack rather than the system code. This is the difference between a SYN spoofing attack and a **SYN flooding** attack. This attack could also use TCP data packets, which would be rejected by the server as not belonging to any known connection. But again, by this time the attack has already succeeded in flooding the links to the server.

All of these flooding attack variants are limited in the total volume of traffic that can be generated if just a single system is used to launch the attack. The use of a single system also means the attacker is easier to trace.

Distributed denial-of-Service Attacks :-

Recognizing the limitations of flooding attacks generated by a single system, one of the earlier significant developments in DoS attack tools was the use of multiple systems to generate attacks. These systems were typically compromised user workstations or PCs. The attacker uses malware to subvert the system and install an attack agent which they can control. Such systems are known as **zombies**. Large collections of such systems under the control of one attacker can be created, collectively forming a **botnet**, as we discuss in Chapter 6. Such networks of compromised systems are a favorite tool of attacker, and can be used for a variety of purposes, including **distributed denial-of-service (DDoS)** attacks.

While the attacker could command each zombie individually, more generally a control hierarchy is used. A small number of systems act as handlers controlling a much larger number of agent systems, There are a number of advantages to this arrangement. The attacker can send a single command to a handler, which then automatically forwards it to all the agents under its control.

Draw Fig.
DDoS Attack Architecture

Many other DDoS tools have been developed since. Instead of using dedicated handler programs, many now use an IRC4 or similar instant messaging server program, or web-based HTTP servers, to manage communications with the agents. Many of these more recent tools also use cryptographic mechanisms to authenticate the agents to the handlers, in order to hinder analysis of command traffic.

Application-based bandwidth Attacks :-

A potentially effective strategy for denial of service is to force the target to execute resource-consuming operations that are disproportionate to the attack effort. For example, Web sites may engage in lengthy operations such as searches, in response to a simple request. Application-based bandwidth attacks attempt to take advantage of the disproportionally large resource consumption at a server. In this section, we look at two protocols that can be used for such attacks.

Sip Flood :-

The standard protocol used for call setup in VoIP is the Session Initiation Protocol (SIP). SIP is a text-based protocol with a syntax similar to that of HTTP. There are two different types of SIP messages :
requests and responses .

Is a simplified illustration of the operation of the SIP INVITE message, used to establish a media session between user agents. In this case, Alice's user agent runs on a computer, and Bob's user agent runs on a cell phone. Alice's user agent is configured to communicate
with a proxy server (the outbound server) in its domain and begins by sending an INVITE SIP request to the proxy server that indicates its desire to invite Bob's user agent into a session.

A SIP flood attack exploits the fact that a single INVITE request triggers considerable resource consumption. The attacker can flood a SIP proxy with numerous INVITE requests with spoofed IP addresses, or alternately a DDoS attack using a botnet to generate numerous INVITE request.

First, their server resources are depleted in processing the INVITE requests. Second, their network capacity is consumed. Call receivers are also victims of this attack. A target system will be flooded with forged VoIP calls, making the system unavailable for legitimate incoming calls.

1) **HTTP-Based Attacks :-**

An HTTP flood refers to an attack that bombards Web servers
with HTTP requests. Typically, this is a DDoS attack, with HTTP requests coming from many different bots. The requests can be designed to consume considerable resources. For example, an HTTP request to download a large file from the target causes the Web server to read the file from hard disk, store it in memory, convert it into a packet stream, and then transmit the packets. This process consumes memory, processing, and transmission resources.

There are a number of countermeasures that can be taken against Slowloris type attacks, including limiting the rate of incoming connections from a particular host; varying the timeout on connections as a function of the number of connections;

and delayed binding. Delayed binding is performed by load balancing software. In essence, the load balancer performs an HTTP request header completeness check, which means that the HTTP request will not be sent to the appropriate Web server until the final two carriage return and line feeds are sent by the HTTP client.
This is the key bit of information. Basically, delayed binding ensures that your Web server or proxy will never see any of the incomplete requests being sent out by Slowloris.

Reflector and amplifier Attacks :-

The attacker sends a network packet with a spoofed source address to a service running on some network server. The server responds to this packet, sending it to the spoofed source address that belongs to the actual attack target. If the attacker sends a number of requests to a number of servers, all with the same spoofed source address, the resulting flood of responses can overwhelm the target's network link. The fact that normal server systems are being used as intermediaries, and that their handling of the packets is entirely conventional, means these attacks can be easier to deploy and harder to trace back to the actual attacker. There are two basic variants of this type of attack: the simple reflection attack and the amplification attack.

Reflection Attacks :-

The **reflection attack** is a direct implementation of this type of attack. The attacker sends packets to a known service on the intermediary with a spoofed source address of the actual target system. When the intermediary responds, the response is sent to the target. Effectively this reflects the attack off the intermediary, which is termed the reflector, and is why this is called a reflection attack.

The intermediary systems are often chosen to be high-capacity network servers or routers with very good network connections. This means they can generate high volumes of traffic if necessary, and if not, the attack traffic can be obscured in the normal high volumes of traffic flowing through them. If the attacker spreads the attack over a number of intermediaries in a cyclic manner, then the attack traffic flow may well not be easily distinguished from the other traffic flowing from the system.

Defenses against denial-of-service Attacks :-

There are a number of steps that can be taken both to limit the consequences of being the target of a DoS attack and to limit the chance of your systems being compromised and then used to launch DoS attacks. It is important to recognize that these attacks cannot be prevented entirely.

Classically, a posting to the well-known Slashdot news aggregation site often results in overload of the referenced server system. Similarly, when popular sporting events like the Olympics or Soccer World Cup matches occur, sites reporting on them experience very high traffic levels. This has led to the terms **slashdotted**, *flash crowd,* or *flash event* being used to describe such occurrences.

There is very little that can be done to prevent this type of either accidental or deliberate overload without also compromising network performance. The provision of

significant excess network bandwidth and replicated distributed servers is the usual response, particularly when the overload is anticipated. This is regularly done for popular sporting sites. However, this response does have a significant implementation cost.

Defending against attacks on application resources generally requires modification to the applications targeted, such as Web servers. Defenses may involve attempts to identify legitimate, generally human initiated, interactions from automated DoS attacks. These often take the form of a graphical puzzle, a captcha, which is easy for most humans to solve but difficult to automate. This approach is used by many of the large portal sites like Hotmail and Yahoo.