DDoS Attack Types: Glossary of Terms

This Distributed Denial of Service (DDoS) attack glossary is intended to provide a high level overview of the various DDoS attack types and typical DDoS attack characteristics.

Common Attack Name	DDoS Attack Characteristics
SYN Flood	In a SYN Flood, a victim server, firewall or other perimeter defense receives (often spoofed and most often from a botnet) SYN packets at very high packet rates that can overwhelm the victim by consuming its resources to process these incoming packets. In most cases if a server is protected by a firewall, the firewall will become a victim of the SYN flood itself and begin to flush its state-table, knocking all good connections offline or even worse - reboot. Some firewalls in order to remain up and running, will begin to indiscriminately drop all good and bad traffic to the destination server being flooded. Some firewalls perform an Early Random Drop process blocking both good and bad traffic. SYN floods are often used to potentially consume all network bandwidth and negatively impact routers, firewalls, IPS/IDS, SLB, WAF as well as the victim servers.
SYN-ACK Flood	In a SYN-ACK Flood, attackers either flood a network with SYN-ACK packets from a sizeable botnet or spoof a victim's IP address range. Typically, a smaller botnet sends spoofed SYN packets to large numbers of servers and proxies on the Internet that generate large numbers of SYN-ACK packets in response to incoming SYN requests from the spoofed attackers. This SYN-ACK flood is not directed back to the botnet, but instead, is directed back to victim’s network and often exhausts the victim’s firewalls by forcing state-table lookups for every incoming SYN-ACK packet. This denial of service attack can render stateful devices inoperable and can also consume excessive amounts of resources on routers, servers and IPS/IDS devices.
ACK or ACK-PUSH Flood	In an ACK or ACK-PUSH Flood, attackers send spoofed ACK (or ACK-PUSH) packets at very high packet rates that fail to belong to any current session within the firewall's state-table and/or server’s connection list. The ACK (or ACK-PUSH) flood exhausts a victim’s firewalls by forcing state-table lookups and servers by depleting their system resources used to match these incoming packets to an existing flow.
Fragmented ACK Flood	In a fragmented ACK Flood DDoS attack, large fragmented (1500+ byte) packets are sent to consume large amounts of bandwidth, while generating a relatively small packet rate. While the protocols allow for fragmentation these packets usually pass through border routers,

	firewalls and IDS/IPS devices uninspected or can consume excessive resources attempting to reassemble and inspect fragmented packets. The packet contents can be randomized, irrelevant data that can consume resources. However, this method can also be used as an Advanced Evasion Technique designed to bypass deep packet inspection devices altogether. The attacker's goal can be to consume all bandwidth of the victim's network or use fragmentation to hide insidious low-and-slow application-layer DDoS attacks, malware, overflows, brute-force etc.
RST/FIN Flood	In a RST/FIN Flood, attackers send highly-spoofed RST or FIN packets at an extremely high rate that do not belong to any session within the firewall's state-table and/or server's session tables. The RST or FIN flood DDoS attack exhausts a victim's firewalls and/or servers by depleting its system resources used to look up and match these incoming packets to an existing session.
Same Source/Dest Flood (LAND Attack)	In a LAND DDoS Attack, a victim receives spoofed SYN packets at a very high rate that have the victim's IP range in both the Source IP and the Destination IP fields in the IP header. This attack exhausts a victim's firewalls and/or servers by exhausting its system resources used to compute this protocol violation. Although the packet's Source and Destination IP are identically defined within a Same Source/Dest attack, the content of the packets are often irrelevant because the attacker is simply attempting to deplete system resources.
Fake Session Attack	In a Fake Session Denial of Service Attack, an attacker sends forged SYN packets, multiple ACK packets and then one or more FIN/RST packets. When these packets appear together, they look like a valid TCP session from one direction only. Since many modern networks utilize asymmetric routing techniques whereby incoming packets and outgoing packets traverse different internet links to improve cost and performance, this attack is harder to detect. This attack simulates a complete TCP communication and is designed to confuse new attack defense tools that only monitor incoming traffic to the network and not bi-directionally monitoring server responses. There are two common variants of this DDoS attack most often observed: the first variant sends multiple SYNs, then multiple ACKs, followed by one or more FIN/RST packets. The second variant skips the initial SYN and starts by sending multiple ACKs, followed by one or more FIN/RST packets. The slow TCP-SYN rate makes the attack harder to detect than a typical SYN flood.
UDP Flood	In a UDP Flood, DDoS attackers send highly-spoofed UDP packets at a very high packet rate using a large source IP range. The victim's network (routers, firewalls, IPS/IDS, SLB, WAF and/or servers) is overwhelmed by the large number of incoming UDP packets. This attack normally consumes network resources and available bandwidth, exhausting the network until it goes offline. UDP attacks are very difficult to detect and block efficiently and are extremely effective in flooding the network with unwanted traffic. UDP floods can overwhelm a network with packets containing random or fixed source IP addresses.

	They can also be used in a Reflective type of attack scenario where volumes of unsolicited and large DNS responses attack a DNS server or even in VoIP and NTP environments.
UDP Fragmentation	In a UDP Fragmentation attack, attackers send large UDP packets (1500+ bytes) to consume more bandwidth with fewer packets. Since these fragmented packets are normally forged and have no ability to be re-assembled, the victim's resources will receive these packets which can possibly consume significant CPU resources to “reassemble” these often useless packets.
Non-Spoofed UDP Flood	In a Non-Spoofed UDP Flood, attackers send non-spoofed UDP packets at a very high packet rate resulting in networks becoming overwhelmed by the large amount of incoming UDP packets. The attack consumes vast amounts of network resources and bandwidth, exhausting the network and forcing denial of service. The packets contain a valid public IP address of the attacker. This type of attack is harder to identify because it resembles good traffic.
ICMP Flood	In an ICMP Flood, attackers send highly-spoofed ICMP packets at large enough volumes to flood a network. The victim's network resources are overwhelmed by the large number of incoming ICMP packets. The attack consumes resources and available bandwidth, exhausting the network until it goes offline. ICMP floods can overwhelm a network with packets containing random or fixed source IP addresses. This attack is often viewed as a Network-Level volumetric attack and can be defeated by L3/L4 Packet Filtering.
ICMP Fragmentation Flood	In an ICMP Fragmentation Flood, attackers send highly-spoofed, large fragmented ICMP packets (1500+ byte) at a very high packet rate and these packets cannot be reassembled. The large packet size can enlarge the bandwidth of an ICMP attack overall. In addition, it causes wasted CPU resources in an attempt to reassemble useless packets.
Ping Flood	In a Ping Flood, attackers use "ping" which is a variant of an ICMP and send highly-spoofed ping (IMCP echo requests) packets at a very high rate and from random source IP ranges or as the IP address of the victim. Attackers can consume all available network resources and bandwidth exhausting the network until it goes offline. Since the PING requests are most often well-formed and highly-spoofed, a PING flood cannot be easily detected by deep packet inspection or other detection techniques.
TOS Flood	In a TOS (Type of Service) Flood, attackers use the ‘TOS’ field of an IP header. This field has evolved over time and is now used for Explicit Congestion Notification (ECN) and Differentiated Services (DiffServ). While this type of flood isn’t seen too often, there are two types of attacks which may be launched based on this field. In the first, the attacker spoofs ECN packets in order to reduce the throughput of individual connections. This could cause the server to appear out of service or unresponsive to customers. In the second, the attacker utilizes the DiffServ class flags in order to potentially increase the priority of the attack traffic over that of non-attack traffic. Utilizing

	DiffServ flags isn’t a DDoS attack in itself; this function is aimed at increasing the effectiveness of the attack.
IP NULL/TCP NULL Attack	In an IP NULL Attack, attackers send packets whereby the IPv4 header field used to specify which Transport Protocol is being used in its payload (e.g.TCP and/or UDP) and sets this field to a value of zero. Firewalls configured for just TCP, UDP, and ICMP may allow this type of packet through. If these packets arrive as a flood, a victim server’s CPU resources may be wasted handling these packets. In a TCP NULL Attack, attackers send packets that have the no TCP segment flags set (six possible) which is invalid. This type of segment may be used in reconnaissance, such as port scanning.
Smurf/Fraggle Attack	In a Smurf Attack, attackers send large numbers of ICMP packets with the intended victim's spoofed source IP address and are broadcast to a computer network using an IP Broadcast address. This causes all hosts on the network to reply to the ICMP request, causing significant traffic to the victim's computer. In a Fraggle Attack, attackers send spoofed UDP packets instead of ICMP echo reply (ping) packets to the broadcast address of a large network resulting in a denial of service.
DNS Flood DNS Amplified (Reflected)	In a DNS Flood, attackers use DNS as a variant of a UDP flood. Attackers send valid but spoofed DNS request packets at a very high packet rate and from a very large group of source IP addresses. Since these appear as valid requests, the victim's DNS servers proceeds to respond to all requests. The DNS server can be overwhelmed by the vast number of requests. This attack consumes large amounts of network resources that exhaust the DNS infrastructure until it goes offline, taking the victim's Internet access (www) down with it. Another possible way of taking advantage of DNS flood is through attackers spoofing a victim's DNS infrastructure and through the use of Open Recursive DNS servers and extensions to the DNS protocol. Very small DNS requests can result in very large and a high-volume of DNS responses (i.e. Amplification Factor). Since attackers spoof a victim's DNS infrastructure, all of the reflected/amplified responses flood a victim’s DNS server, which usually takes them offline. Since the DNS requests and responses look 100% normal, this attack is rarely detectable by deep packet inspection technologies.
NTP Flood	In a NTP Flood, attackers use NTP as a variant of a UDP flood. Attackers send valid but spoofed NTP request packets at a very high packet rate and from a very large group of source IP addresses. Since these appear as valid requests, the victim's NTP servers proceeds to respond to all requests. The NTP server can be overwhelmed by the

NTP Amplified (Reflective)

vast number of requests. This attack consumes large amounts of network resources that exhaust the NTP infrastructure until it goes offline.

Another possible way of taking advantage of NTP flood is when attackers spoof a victim's NTP infrastructure and use Open NTP servers, which send (MON_GETLIST) very small requests resulting in a very high-volume of NTP responses (Amplification Factor). Since attackers spoof a victim's NTP infrastructure, all of the reflected/amplified responses flood a victim’s NTP server, which take them offline or flood the network and take it offline as well. This attack is rarely detectable by deep packet inspection technologies because the NTP requests and responses seem to be 100% normal.

Other Amplified Attacks (Reflective)

According to US CERT, certain UDP protocols have been identified as potential attack vectors using Amplified (Reflective) Attacks. Most attacks using these protocols would be performed similarly to the DNS and NTP attacks described above.

- DNS
- NTP
- SNMPv2
- NetBIOS
- SSDP
- CharGEN
- QOTD
- BitTorrent
- Kad
- Quake Network Protocol
- Steam Protocol

Slow Session Attack

In a Slow Session Attack, attackers send valid TCP-SYN packets and perform TCP three-way handshakes with the victim to establish valid sessions between the attacker and victim. The attacker first establishes a large number of valid sessions then slowly responds with an ACK packet and incomplete requests to keep the sessions open for long periods of time. Normally, the attacker will set the attack to send an ACK packet with an incompleted request typically before the session time-out is triggered by the server. The "held-open" sessions can eventually exhaust the victim server's resources used to compute this irregularity. Low-and-slow tools have also been designed to consume all 65,536 available "sockets" (source ports) resulting in a server's inability to establish any new sessions. Slow Session Attacks are

	always non-spoofed in order to hold sessions open for long periods of time.
Slow Read Attack	In a Slow Read DDoS Attack, attackers send valid TCP-SYN packets and perform TCP three-way handshakes with the victim to establish valid sessions between the attacker and victim. The attacker first establishes a large number of valid sessions and begins to request to download a document or large object from each attacking machine. Once the download begins the attacking machines begin to slow down the acknowledgement of received packets. The attackers will continue to slow down the receipt of packets, which consumes excess resources on the delivering server since all the associated processes appear to be in a very slow receiving network. Slow Read Attacks are always non-spoofed in order to hold sessions open for long periods of time.
HTTP Fragmentation	In an HTTP Fragmentation Attack, a non-spoofed attacker establishes a valid HTTP connection with a web server. The attacker proceeds to fragment legitimate HTTP packets into the smallest fragments possible and sends each fragment as slow as the server time-out will allow, which eventually holds the HTTP connection open for a long period of time without raising any alarms. By opening multiple extended sessions per attacker, the attacker can silently force a web application offline with just a handful of attacking machines.
Excessive Verb (HTTP GET Flood)	In a GET Flood, attackers send large numbers of valid HTTP requests to a victim's web server. The HTTP request is most often a GET request and is directed to the most CPU intensive process on the victim web server. Each attacker can generate large numbers of valid GET requests so the attacker can use a relatively small number of attacking machines to take a system offline. HTTP GET Floods are non-spoofed and the source IP address is the actual public IP of the attacker machine (or NAT Firewall). The most common variant of this attack uses GET requests but an attacker can also use HEAD, POST, PUT, OPTIONS or any other HTTP method to cause an outage. This attack is viewed as a low-and-slow Application-Layer attack and normally consumes little bandwidth but eventually renders the victim's servers unresponsive.
Excessive Verb - Single Session	In an Excessive Verb Attack, attackers take advantage of a feature of HTTP 1.1 that allows multiple client requests within a single HTTP session. In this case, attackers can lower the session request rate of an HTTP attack in order to come under the radar of request rate-limiting features found on some attack defense systems deployed today. This attack is viewed as a low-and-slow Application-Layer attack and normally consumes little bandwidth but eventually renders the victim's servers unresponsive.
Multiple Verb - Single Request	In a Multiple Verb Attack, attackers use a variation of the Excessive Verb attack vector. The attacking machines create multiple HTTP requests, not by creating them one after another for example during a single HTTP session attack, but instead by creating a single packet filled with multiple requests. It's a variant of the Excessive VERB attack

	whereby the attacker can maintain high CPU processing loads on the victim server with very low attack packet rates. The low packet rates make the attacker nearly invisible to NetFlow attack detection techniques. Also if the attacker selects the HTTP VERB carefully these attacks will also bypass deep packet inspection technologies as well. This attack is viewed as a low-and-slow Application-Layer attack and normally consumes little bandwidth but eventually renders the victim's servers unresponsive.
Recursive GET	A Recursive GET Attack is a variant of the Excessive Verb attack. In this case, an attacker identifies multiple pages and/or images and generates HTTP GET requests that appear to “scroll” through these pages or images trying to replicate a normal user. This attack can be combined with any of the VERB attack methods to make this attack vector very difficult to detect because the requests appear to be very legitimate.
Random Recursive GET	In a Random Recursive GET Attack, attackers use a modified version of a Recursive GET. This attack is designed primarily for forum sites or news sites whereby web pages are indexed numerically, usually in a sequential manner. The attacking GET statements will insert a random number within a valid range of page reference numbers making each GET statement different than the previous one.
Specially Crafted Packet	In a Specially Crafted Packet Attack, attackers take advantage of websites with poor designs, vulnerable web applications and/or have improper integration with backend databases. For example, attackers can exploit vulnerabilities in HTTP, SQL, SIP, DNS etc., and generate specially crafted packets to take advantage of these protocol “stack” vulnerabilities to ultimately take the servers offline. They can also generate requests that will lock up database queries. These attacks are highly specific and effective because they consume huge amounts of server resources and often are launched from a single attacker. An example of a Specially Crafted Denial of Service attack is MS13-039.