RIOREY
NextGen DDoS Experts

Taxonomy of DDoS Attacks

	Attack Types		Attack Matrix Dimensions									
			Nature of IP	Handshake	Source IP Range	Packet Rate	Packet Size	Packet Content	Fragmenting	Session Rate	Session Duration	VERB Rate
TCP BASED	1	SYN Flood	Spoofed	None	Large	High	Small	---	---	---	---	---
	2	SYN-ACK Flood	Spoofed	None	Large	High	---	---	---	---	---	---
	3	ACK & PUSH ACK Flood	Spoofed	None	Large	High	---	---	---	---	---	---
	4	Fragmented ACK	Spoofed	None	Large	Moderate	Large	---	High	---	---	---
	5	RST or FIN Flood	Spoofed	None	Large	High	---	---	---	---	---	---
	6	Synonymous IP	Spoofed	None	Single IP	High	---	---	---	---	---	---
	7	Fake Session	Spoofed	None	Large	Low	---	---	---	---	---	---
	8	Session Attack	Non-Spoofed	Yes	Small	Low	---	---	---	Low	Long	---
	9	Misused Application	Non-Spoofed	Yes	Small	Variable	---	---	---	High	Short	---
TCP HTTP BASED	10	HTTP Fragmentation	Non-Spoofed	Yes	Small	Very Low	Small	Valid	High	Very Low	Very Long	Very Low
	11	Excessive VERB	Non-Spoofed	Yes	Small	High	---	Valid	---	High	Short	High
	12	Excessive VERB Single Session	Non-Spoofed	Yes	Small	Low	---	Valid	---	Low	Moderate	High
	13	Multiple VERB Single Request	Non-Spoofed	Yes	Small	Very Low	Large	Valid	---	Low	Long	High
	14	Recursive GET	Non-Spoofed	Yes	Small	Low	---	Valid	---	Low	Short	Low
	15	Random Recursive GET	Non-Spoofed	Yes	Small	Low	---	Valid	---	Low	Short	Low
	16	Faulty Application	Non-Spoofed	Yes	Small	Low	---	Valid	---	Low	Short	Low
UDP BASED	17	UDP Flood	Spoofed	---	Very Large	Very High	Small	Not Valid	---	---	---	---
	18	Fragmentation	Spoofed	---	Moderate	Very High	Large	Not Valid	High	---	---	---
	19	DNS Flood	Spoofed	---	Very Large	Very High	Small	Valid	---	---	---	---
	20	VoIP Flood	Spoofed	---	Very Large	Very High	Small	Valid	---	---	---	---
	21	Media Data Flood	Spoofed	---	Very Large	Very High	Moderate	Valid	---	---	---	---
	22	Non-Spoofed UDP Flood	Non-Spoofed	---	Small	Very High	---	Valid	---	---	---	---
ICMP BASED	23	ICMP Flood	Spoofed	---	Very Large	Very High	Variable	Not Valid	---	---	---	---
	24	Fragmentation	Spoofed	---	Moderate	Very High	Large	Not Valid	High	---	---	---
	25	Ping Flood	Spoofed	---	Very Large	Very High	Small	Valid	---	---	---	---

sales@riorey.com | www.riorey.com

RioRey Taxonomy of DDoS Attacks: Definitions

1. SYN Flood. Clients generate a SYN packet (64 bytes) to request a new session from a host server. As the TCP three-way communication handshake is created, the host will track and allocate each of the client's sessions until the session is closed. In a SYN flood, a victim server receives spoofed SYN requests at a high packet rate that contain fake source IP addresses. The SYN flood overwhelms the victim server by depleting its system resources (connection table memory) normally used to store and process these incoming packets, resulting in performance degradation or a complete server shutdown. A well-crafted SYN flood often fools deep-packet inspection filtering techniques. SYN-Cookie defense can be used to defend against large-scale SYN floods but this requires all servers to support this capability.

2. SYN-ACK Flood. Host servers generate SYN-ACK packets in response to incoming SYN requests from clients. During a SYN-ACK flood, the victim server receives spoofed SYN-ACK packets at a high packet rate. This flood exhausts a victim's server by depleting its system resources (memory, CPU, etc.) used to compute this irregularity, resulting in performance degradation or a complete server shutdown.

3. ACK & PUSH ACK Flood. After a TCP-SYN session is established between a host and a client, ACK or PUSH ACK packets are used to communicate information back and forth between the two until the session is closed. During an ACK flood, a victim receives spoofed ACK packets at a high packet rate that fail to belong to any session within the server's connection list. The ACK flood exhausts a victim's server by depleting its system resources (memory, CPU, etc.) used to match these incoming packets, resulting in performance degradation or a complete server shutdown.

4. Fragmented ACK. A variation of the ACK & PUSH ACK Flood. This attack uses 1500 byte size packets to consume large amounts of bandwidth, while generating a relatively moderate packet rate. Because routers do not reassemble fragmented packets at the IP level, these packets usually pass through routers, ACL, firewalls, and IDS/IPS unimpeded. The packet content is usually randomized, irrelevant data. The attacker's goal is to consume all bandwidth of the victim's network. A Fragmented ACK attack will affect performance of all servers in the victim's network.

5. RST or FIN Flood. In order to close a TCP-SYN session between a client and a host, the servers exchange RST or FIN packets to close the session using a three-way or four-way TCP communication handshake. During a RST or FIN flood, a victim server receives spoofed RST or FIN packets at a high rate that do not belong to any session within the server's databases. The RST or FIN flood exhausts a victim's server by depleting its system resources (memory, CPU, etc.) used to match these incoming packets, resulting in performance degradation or a complete server shutdown.

6. Synonymous IP. A victim receives spoofed TCP-SYN packets at a high rate that have the victim's information specified as both the Source IP and the Destination IP. This attack exhausts a victim's server by depleting its system resources (memory, CPU, etc.) used to compute this irregularity, resulting in performance degradation or a complete server shutdown. Although the packet's Source and Destination IP are identically defined within a Synonymous IP attack, the content is irrelevant because the attacker is simply depleting the victim's system resources.

7. Fake Session. This Attack generates a forged SYN, multiple ACK and then one or more FIN/RST packets. These packets together appear to look like a valid TCP session from one direction. Most networks implement asymmetric routing techniques, in which incoming packets and outgoing packets travel on different links to optimize cost and performance. In turn, modern network defense tools are designed to monitor single directional traffic and do not rely on the return traffic from the server. This attack fakes a complete TCP communication and is designed to fool new defense tools that only monitor incoming traffic to the network. There are two variations of this attack: the first variation generates multiple forged SYNs, then multiple ACKs, followed by one or more FIN/RST packets, and the second variation skips the initial SYN, and starts by generating multiple ACKs, followed by one or more FIN/RST packets. The low TCP-SYN rate makes the attack harder to detect than a typical SYN flood while achieving the same result: the depletion of the victim's system resources.

8. Session Attack. A valid TCP-SYN session is generated between a BOT and a victim. Once the session is established, the attacker delays responding with an ACK packet to keep the session open until a Session Time Out is triggered. The empty session exhausts the victim's server by depleting its system resources (memory, CPU, etc.) used to compute this irregularity, resulting in performance degradation or a complete server shutdown. Session Attacks are non-spoofed: the source IP is the actual public IP of the attacker BOT, and the source IP range is equal to the number of BOTs used in the attack.

9. Misused Application Attack. The attacker does not use BOTs to consume the system resources of a victim's server. Rather, an attacker redirects valid clients belonging to a high traffic application, such as peer-to-peer services, to a victim server. The target victim is then overwhelmed with traffic from a group of misdirected computers trying to form a legitimate connection with its server. Once the traffic is misdirected towards the victim server, the attacker computer becomes untraceable by dropping from the network. The overwhelming connection requests received by the victim's server depletes its system resources, resulting in performance degradation or a complete server shutdown.

10. HTTP Fragmentation. In this attack, the BOT (non-spoofed) establishes a valid HTTP connection with a web server. The BOT proceeds to fragment legitimate HTTP packets into tiny fragments, sending each fragment as slow as the server time out allows, holding up the HTTP connection for a long time without raising any alarms. For Apache and many other web servers designed with improper time-out mechanisms, this HTTP session time can be extended to a very long time period. By opening multiple extended sessions per BOT, the attacker can silently stop a web service with just a handful of BOTs.

11. Excessive VERB. The attacking BOT generates a large number of valid HTTP requests to a victim web server. The HTTP request is generally a GET request of a common web page or image, often a large one. Each BOT can generate a large number of valid requests (usually over 10 requests a second) so the attacker can use a relatively small number of BOTs to achieve a successful attack. VERB Attacks are non-spoofed: the source IP is the actual public IP of the attacker BOT and the source IP range is equal to the number of BOTs used in the attack. The most common form of VERB attack uses GET requests but the attacker can also use POST or other HTTP actions to cause the same impact on the victim. An Excessive VERB Attack does not generate significant bandwidth increase on the network but can render the victim unresponsive by consuming server resources.

12. Excessive VERB Single Session. A variation of the Excessive VERB Attack. This attack uses the feature of HTTP 1.1 to allow multiple requests within a single HTTP session. Thus, the attacker can limit the session rate of an HTTP attack and bypass session rate limitation defenses of many security systems. Excessive VERB Single Session Attack and Excessive VERB Attack have the same effect on a victim web server.

13. Multiple VERB Single Request. This Attack is also a variation of the Excessive Verb Attack strategy. The attacking BOT creates multiple HTTP requests, not by issuing them one after another during a single HTTP session, but by forming a single packet embedded with multiple requests. It is a refinement of the Excessive VERB attack, where the attacker can maintain high loads on the victim server with a low attack packet rate. This low rate makes the attacker nearly invisible to netflow anomaly detection techniques. Also, if the attacker selects the HTTP VERB carefully these attacks will bypass deep packet inspection techniques.

14. Recursive GET. Another refinement to the VERB attack is a Recursive GET attack. The attacker collects several pages or images and generates GET requests that "walk" through these pages or images. This method can be combined with any of the VERB attack methods to make this attack very difficult to detect because the requests appear to be legitimate.

15. Random Recursive GET. This attack is a modified version of a Recursive GET but designed for forum sites or news sites where 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 a previous one.

16. Faulty Application. DDoS attackers take advantage of websites with poor designs or improper integration with databases. Using SQL-like injections, an attacker can generate requests that will lock up database queries. These attacks are highly specific and effective because they consume server resources (memory, CPU, etc.).

17. UDP Flood. During a UDP flood, a victim server receives spoofed UDP packets at a very high packet rate and with a large source IP range. The victim server is overwhelmed by the large number of incoming UDP packets. The attack consumes network resources and available bandwidth, exhausting the network until it shuts down. A full communication handshake is not used in the UDP software to exchange data, making UDP attacks difficult to detect and extremely effective in flooding the network bandwidth. UDP floods can overwhelm a network with packets containing randomized or fixed Source IP addresses and can be designed to target a specific server by using the victim's information as the Destination port and IP within the packets.

18. UDP Fragmentation. A variation of the UDP flood. The attacker uses large packets (1500 bytes) to consume more bandwidth with fewer packets. Since these fragmented packets are forged and have no real relationship for reassembly, the victim server receiving these packets will spend CPU resources to "reassemble" useless packets. This often causes the processors to overload and sometimes reboot the entire system. This attack is harder to identify because it resembles good traffic.

19. DNS Flood. An application-specific variation of the UDP flood. During a DNS flood, a victim DNS server receives valid but spoofed DNS request packets at a very high packet rate and from a very large pool of source IP. The victim server cannot determine which packet is from a real server and therefore proceeds to respond to all requests. The server is overwhelmed by the requests. This attack consumes network resources and available bandwidth that exhausts the network until it shuts down. Spoofed DNS attacks are well-crafted flood attacks – the content of spoofed DNS packets are designed to mimic actual DNS requests. Since they are 100% normal looking packets, this attack is not detectable by deep packet inspection. With a wide range of available attacking IP, the attacker can easily evade most traffic anomaly detection techniques.

20. VoIP Flood. A variation of an application specific UDP flood. A victim VoIP server receives spoofed VoIP packets at a very high packet rate and with a very large source IP range. The victim server has to sort out the proper VoIP connections from the forged ones, consuming a detrimental amount of resources. VoIP floods can overwhelm a network with packets containing randomized or fixed Source IP addresses. A fixed Source IP VoIP attack mimics traffic from large VoIP servers, and can be very difficult to identify because it resembles good traffic.

21. Media Data Flood. In addition to VoIP, UDP floods can take the form of any media data, causing a Media Data flood (Video, Audio, etc.). During an attack, a victim server receives spoofed Media Data packets at a very high packet rate and with a very large source IP range. The victim server is overwhelmed by the large number of incoming Media Data packets, consuming network resources and available bandwidth until the network shuts down. Similar to VoIP floods, Media Data floods can overwhelm a network with packets containing randomized or fixed Source IP addresses, making the attack difficult to identify because it resembles good traffic. Both modes of Media Data floods can easily exhaust network bandwidth as well as CPU resources.

22. Non-Spoofed UDP Flood. During this attack, a victim server receives non-spoofed UDP packets at a very high packet rate and is overwhelmed by the large amount of incoming UDP packets. The attack consumes network resources and available bandwidth, exhausting the network until it shuts down. In Non-Spoofed UDP Flood packets, the source IP is the actual public IP of the attacker BOT, and the source IP range is equal to the number of BOTs used in the attack. This type of attack is harder to identify because it resembles good traffic.

23. ICMP Flood. A victim server receives spoofed ICMP packets at a very high packet rate and with a very large source IP range. The victim server is overwhelmed by the large number of incoming ICMP packets. The attack consumes network resources and available bandwidth, exhausting the network until it shuts down. A full communication handshake is not used in the ICMP software stack to exchange data, making ICMP-based attacks difficult to detect. ICMP floods can overwhelm a network with packets containing randomized or fixed Source IP addresses. ICMP floods can target a specific server by using the victim's information as the Destination port and IP within the packets.

24. ICMP Fragmentation. A victim server receives spoofed, large fragmented ICMP packets (1500 byte) at a high incoming packet rate and these packets cannot be reassembled. The large packet size expands the bandwidth of an ICMP attack. In addition, it causes the victim CPU to waste resources when it attempts to reassemble useless packets. This attack will often cause victim servers to overload and reboot.

25. Ping Flood. An application specific adaptation of ICMP flood. During a Ping flood, a victim server receives spoofed ping (IMCP echo requests) at a very high packet rate and from a very large source IP range. The victim server is overwhelmed by the large number of incoming Ping packets. The attack consumes network resources and available bandwidth, exhausting the network until it shuts down. The spoofed Source IP can be random or set as the address of the victim. Since the PING requests are usually well formed and from a large number of source IP addresses, the PING flood cannot be easily detected by either deep packet inspection or anomaly detection techniques.