MAZEBOLT

2018

The Most Common DDoS Attacks

Table of Contents

MazeBolt Introduction

MazeBolt is an Israeli Cyber Security threat assessment company that strengthens enterprises' resistance to cyber-attacks. MazeBolt's pioneering DDoS Testing & Phishing Simulation & Awareness solutions are used by Fortune 1000 & NASDAQ-listed companies in more than 50 countries operating in 20 languages.

MazeBolt's BaseLine DDoS Testing Methodology – the ***de-facto*** industry standard – was developed on the basis of years of in-depth experience and understanding of how DDoS attacks and DDoS mitigation work.

Executive Summary

While the individual DDoS attack code varies by dark web vendor, developer, and attacker, the attacks themselves are based on a finite number of underlying principles. The DDoS attacks in this report were chosen on the basis of public sources and MazeBolt's rich testing experience and constitute the main attacks companies should be validating their mitigation against.

As with most cyberattacks, DDoS attacks are a "when," not an "if." DDoS attacks generally target all three levels of your website infrastructure:

- **Layer 3 (Volumetric IP level),** which generate massive amounts of traffic, clogging the bandwidth, slowing the web or service performance and ultimately preventing website access or the ability to access services
- **Layer 4 (Volumetric IP level and Protocol Transport level)**, which use up all the processing capacity by saturating an end server's CPU or connection table using a connection-oriented attack
- **Layer 7 (Lower volume, higher connections, low and slow, application attacks)** exploit weaknesses in the application layer, overwhelming the database or server powering the application directly

"…it's clear that companies are buying [DDoS Mitigation] solutions that aren't working""

Barrett Lyon, Head Research and Development of Security Solutions @ Neustar

Layer 3 Attacks

ICMP (Internet Control Message Protocol Type 8) Flood

These consume computing power, bring down perimeter devices, and saturate bandwidth, where the packets overload the pipe and servers until the system fails. They are generally spoofed attacks and come at a very high rate. These are effectively echo requests, which may illicit echo responses (ICMP Type 0). If they are not dropped by the DDoS mitigation devices on the perimeter, they may overwhelm the internal network architecture; this flood may also generate outgoing traffic due to answers for the echo request.

IP Fragmented Flood

IP Fragmented Floods are aimed at consuming computing power and saturating bandwidth; they may also crash devices in rare cases because of buggy packet parsing. Fragmented IP Floods are generally spoofed attacks and normally come at a very high rate. They generally have no identifiable Layer4 protocol, just garbage, and the packets have to be reassembled by various devices along the way. Generally, this flood is used as a basic but effective flood to bring down perimeter devices or saturate bandwidth.

Malformed IP Flood

Malformed IP Floods are aimed at consuming computing power and saturating bandwidth. They may also crash devices in rare cases because of buggy packet parsing. Malformed IP Floods are generally spoofed attacks and normally come at a very high rate. They have no identifiable Layer 4 protocol, just garbage. Generally, this flood is used as a basic but effective flood to bring down perimeter devices or saturate bandwidth. Many ISP's today stop this type of attack from occurring since the routers at ISP's will not forward such packets.

Layer 4 Attacks

SYN Flood

A SYN flood, generally caused by botnets, is another attack targeting server resources via the firewall or perimeter defenses. They are aimed at consuming connection resources on the backend servers themselves and on stateful elements, like firewalls and load balancers by sending numerous TCP-SYN requests toward targeted services while spoofing the attack packets source IP. This leaves the TCP backlog saturated and the server and/or daemon attacked will not be able to receive any new connections.

It begins with the attacker sending a message to the targeted server, which responds with an "SYN ACK" (synchronize acknowledgement) message signaling receipt and awaiting the connection to be closed by the requesting machine (the attacker). Instead, the connection says open until

it times out, ultimately exhausting resources and causing the server to go offline.

UDP Fragmentation or UDP Garbage Flood

In a UDP garbage flood, attackers try to saturate bandwidth to bring about a DDoS state to the network. The attack generally occurs by sending a rapid succession of UDP datagrams with spoofed IPs to a server within the network via various different ports, forcing the server to respond with ICMP traffic. This is normally done by sending a rapid succession of UDP datagrams with spoofed IPs to a server within the network via various different ports, forcing the server to respond with ICMP traffic. The saturation of bandwidth happens both on the ingress and the egress direction. This flood also has some garbage in the data section of the datagram.

Large, forged packets of more than 1,500 bytes are sent, requiring fragmentation to "fit" through the pipes, saturating bandwidth to shut down the network to outside, legitimate requests. Because these packets are not legitimate, they cannot be reassembled. While the network firewall is busy trying to put them back together, the network itself can be unprotected for hours. While an "official" DDoS attack, it gives coverage for more nefarious activities to occur in other parts of the network.

See here for a full technical explanation of the UDP Flood/UDP Garbage Flood.

Reflection Attack

A reflection attack passes the threat around to many computers, which then sends them back to the targeted computer, using spoofed sources. The initial (attacking) computers receive the packets (all with the same spoofed source IP – the victims IP – and respond to the spoofed address that routes to the target (the victim). This attack is only possible with connectionless protocols In rare cases, out-of-state TCP packets may also be used if the attacking nodes support the response to out-of- state packets, e.g. UDP.

ACK Flood

An ACK flood is designed to disrupt network activity by saturating bandwidth and resources on stateful devices in its path. By continuously sending ACK packets towards a target, stateful defenses can go down (in some cases into a fail-open mode). This flood could be used as a smoke screen for more advanced attacks. This is true for other out-of-state floods too.

See here for a full technical explanation of an ACK flood.

Empty Connection Flood

Empty connection floods are designed to saturate the targeted open port's sockets. The idea is that as connections increase, you are saturating the TCP stack to finally bring about a situation whereby the particular daemon/service is unable to accept any new connections. An Empty Connection Flood may also saturate other stateful devices in its path such as firewalls or IPS systems. An Empty connection flood generally won't have a high Mbps throughput.

FIN Flood

A FIN Flood is designed to disrupt network activity by saturating bandwidth and resources on stateful devices in its path. By continuously sending FIN packets toward a target, stateful defenses can go down (in some cases - into a fail open mode). This flood could be used as a smoke screen for more advanced attacks. This is true for other out-of-state floods, too.

FIN+ACK Flag Flood

FIN+ACK Floods are aimed at consuming computing power and saturating bandwidth. FIN+ACK Floods are generally spoofed attacks and normally come at a very high rate. FIN+ACK floods, if not dropped by stateful devices on the perimeter, may overwhelm the internal network architecture. Generally, this flood is used as a basic but effective flood to bring down perimeter devices or saturate bandwidth.

See here for a full technical explanation of the FIN+ACK Flood.

URG Flag Flood

URG Floods are aimed at consuming computing power and saturating bandwidth. URG Floods are generally spoofed attacks and normally come at a very high rate. URG Floods, if not dropped by stateful devices on the perimeter, may overwhelm the internal network architecture. Generally, this flood is used as a basic but effective flood to bring down perimeter devices or saturate bandwidth.

ALL TCP Flags Flood

ALL TCP Flags Floods are aimed at consuming computing power and saturating bandwidth. ALL TCP Flags Floods are generally spoofed attacks and normally come at a very high rate. ALL TCP Flags Floods, if not dropped by stateful devices on the perimeter, may overwhelm the internal network architecture. Generally, this flood is used as a basic but effective flood to bring down perimeter devices or saturate bandwidth. These packets should also be rejected on the basis that they are non-RFC compliant, which means they do not follow standard TCP protocols.

PSH+ACK Flag Flood

PSH+ACK Floods are aimed at consuming computing power and saturating bandwidth. PSH+ACK Floods are generally spoofed attacks and normally come at a very high rate. PSH+ACK floods, if not dropped by stateful devices on the perimeter, may overwhelm the internal network architecture. Generally, this flood is used as a basic but effective flood to bring down perimeter devices or saturate bandwidth.

RST Flood

RST Floods are aimed at consuming computing power and saturating bandwidth. RST Floods are generally spoofed attacks and normally come at a very high rate. RST Floods, if not dropped by stateful devices on the perimeter, may overwhelm the internal network architecture. Generally, this flood is used as a basic but effective flood to bring down perimeter devices or saturate bandwidth.

Layer 7 Attacks

Brobot Flood

Brobot is similar to an HTTP flood and is designed to overwhelm web servers' resources by continuously requesting single or multiple URLs from many source attacking machines. Brobot dynamically changes its user agent and can change HTTP method type (GET/POST). Brobot can also add a suffix to the end of URLs, which will enable the request to bypass many CDN systems. When the servers' limits of concurrent connections are reached, the server can no longer respond to legitimate requests from other users.

SlowLoris

A "low-and-slow" attack vector, it has the goal of saturating the entire TCP stack for the HTTP/S daemon. These attacks are harder to detect because they do not need the volume of resources required for other types of attack. They enable a single attacker to take down a web server without affecting other ports or services on the targeted network. SlowLoris sends HTTP headers at certain intervals combined with partial requests, which opens connections to the target machine and keeps them open, eventually overflowing the maximum concurrent connection volume, preventing legitimate clients from accessing the server.

Image of a Slow Loris in the wild. It's a primate originating in South East Asia with a rare toxic bite after which the DDoS Attack is named.

DNS Request Flood/DNS (Domain Name System) Flood

Like many other types of flood attacks, the attackers send spoofed requests at a high packet rate from a wide range of IP addresses; the difference is that the targets are the DNS servers and cache mechanisms. The DNS Request Floods send DNS request packets to a DNS server in an attempt to overwhelm the server's ability to respond to legitimate DNS requests. If the DNS is unavailable to legitimate users, this can completely cripple most modern networks since fully qualified domain names or absolute domain names (a domain name that specifies its exact location within the DNS hierarchy) are used to provide most services. The Amplified DNS flood sends small requests with spoofed IP addresses across the Internet to open DNS resolvers. They reply with responses larger than request, which flood the victim's DNS (Or other) servers, taking them offline.

HTTP/s Flood with Browser Enumeration

HTTP Floods with Browser Enumeration are designed to overwhelm web servers' resources by continuously requesting single or multiple URLs from many source attacking machines, unlike with a normal HTTP Flood (without browser enumeration). When you have browser enumeration, JavaScript can be interpreted, where simple JavaScript challenges are bypassed. When the servers' limits of concurrent connections are reached, the server can no longer respond to legitimate requests from other users.

HTTP GET Flood/HTTP Flooders

Attacks are based on seemingly legitimate HTTP GET or POST requests, forcing the server or applications to respond to every request. These are designed to overwhelm web servers' resources by continuously requesting single or multiple URLs from many source attacking machines. A GET request is used to download a page or image from the server, while a POST request is used to pass data to the server, like a form, uploading a file, etc. It uses less bandwidth but because it requires a more complex response, it still maxes out the server capabilities. HTTP Floods are referred to as application or connection-oriented floods. The number or source IPs and the total amount of connections will be a deciding factor affecting service outage.

HTTPS Flood

Similar to an HTTP Flood, HTTPS Floods are designed to overwhelm web servers' resources by continuously requesting single or multiple URLs from many source attacking machines. When the servers' limits of concurrent connections are reached, the server can no longer respond to legitimate requests from other users. However, an HTTPS flood can also saturate an SSL daemon due to the high amount of computing resources required to perform the asymmetric encryption for a single user.

Dynamic HTTP Flood

Similar to regular HTTP Floods, a Dynamic HTTP Flood continuously changes the suffix of the HTTP request; this forces services like CDNs to request from the originating webserver. Dynamic HTTP Floods are designed to overwhelm web servers' resources by continuously requesting single or multiple URLs from many source attacking machines. When the servers' limits of concurrent connections are reached, the server can no longer respond to legitimate requests from other users.

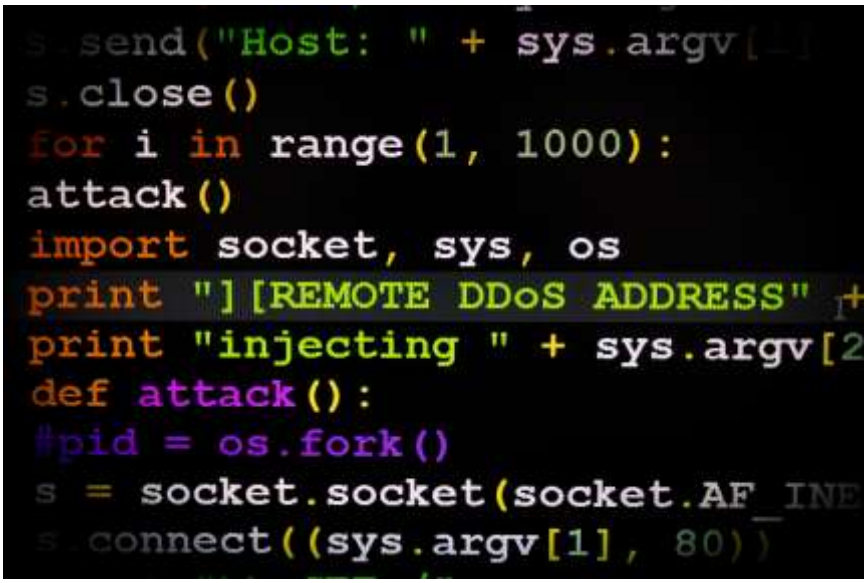
SSL Negotiation Flood

SSL Negotiation Floods attempt to establish many new SSL handshakes with the targeted server. Each handshake in this attack is a new TCP connection and affects the target server. Opening and closing many such connections, SSL/TLS handshakes are up to fifteen times more CPU intensive on the server than on the client. While the server may not be completely down under such an attack, it may be unable to establish any new SSL connections, effectively leaving that SSL service unavailable.

See here for a full technical explanation of an SSL Negotiation Flood.

THC-SSL Flood

This attack uses a single TCP connection to continuously renegotiate new encryption keys. The important thing with this attack is that in one single connection the server "allows" the client to request a new SSL handshake within the same TCP connection. This attack will work effectively on the server, which allows its clients to initiate a new handshake at the time of their choosing, leaving such behavior in the server increases its vulnerability to DDoS attacks.



Conclusion

No matter the size of your network, one of these attacks is highly likely to get through. Malicious actors will do whatever they can to bring your system down. You need to take proactive measures to ensure that your DDoS protection system is as robust and hardened as possible. Speak with your network vendors, your MSSSPs, your cloud providers, and any other entity, such as MazeBolt, that can have an impact on ensuring your DDoS protection is up to the highest possible standards.

References

1. https://en.wikipedia.org/wiki/Denial-of-service_attack#Attack_techniques
2. https://securitytoday.com/articles/2018/02/26/top-ddos-attack-types-exposed.aspx
3. https://www.itbusinessedge.com/slideshows/5-types-of-ddos-attacks-to-defend-against-in-2016-07.html
4. http://www.eweek.com/security/recognizing-the-most-common-ddos-attack-vectors-in-an-it-system
5. https://www.esecurityplanet.com/network-security/types-of-ddos-attacks.html
6. https://www.linkedin.com/pulse/top-10-most-common-ddos-attacks-muhammad-shahbaz-khan/
7. https://www.rivalhost.com/12-types-of-ddos-attacks-used-by-hackers
8. http://blog.fortinet.com/post/security-101-top-10-most-common-ddos-attacks
9. https://www.incapsula.com/ddos/ddos-attacks/