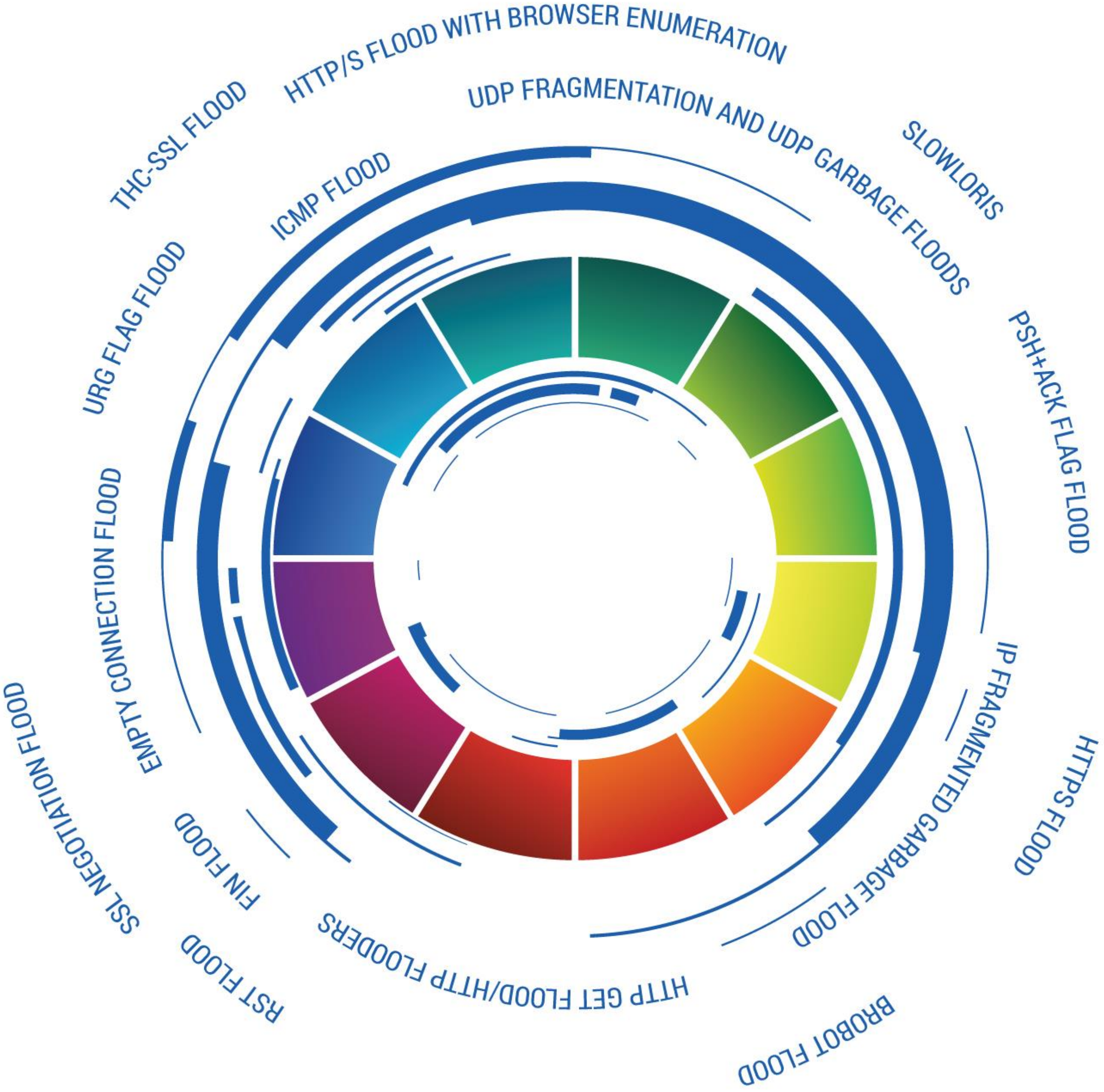
MAZEBOLT

2018

The Critical 18

The DDoS attack test palette



www.mazebolt.com

Contents

MazeBolt Introduction

MazeBolt is an Israeli cybersecurity 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

DDoS attacks come in thousands of flavors, but the underlying infrastructure is based on a finite number of principles. MazeBolt's systematic DDoS testing methodology – in use since 2013 – focuses on the analysis of typical attacks organizations may face. The 18 types listed in this report constitute the main attacks companies should validating their mitigation against.

These 18 DDoS attacks fall into three categories:

- **Layer 3 (Volumetric IP level)** attacks generate massive amounts of traffic, clog bandwidth, slow the web or service performance and ultimately prevent website access or the ability for your customers or employees to access services
- **Layer 4 (Volumetric IP level and Protocol Transport level)** attacks saturate an end server's CPU or connection table using a connection-oriented attack, which uses up all the processing capacity
- **Layer 7 (Lower volume, higher connections, low and slow, application attacks)** overwhelm the database or server powering the application by directly exploiting weaknesses in the application layer

> ***"Distributed denial-of-service (DDoS) attacks are nothing new, yet these attacks remain one of the most common causes of high-profile outages and interruptions of client-facing services."***
>
> John Whetstone, NSS Labs

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.

See here for a full technical explanation of the ICMP Ping Flood.

IP Fragmented Garbage Flood

IP Fragmented Garbage 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.

See here for a full technical explanation of the IP Fragmented Garbage Flood.

Layer 4 Attacks

UDP and UDP Garbage Floods

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.

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 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.

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

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.

See here for a full technical explanation of the FIN 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.

See here for a full technical explanation of the URG Flood.

PSH+ACK Flag Flood

PSH+ACK Floods are generally spoofed attacks and normally come at a very high rate, consuming computing power and saturating bandwidth. 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.

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

RST Flood

RST Floods consume computing power and saturate 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.

See here for a full technical explanation of the RST Flood.

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.

See here for a full technical explanation of Brobot attack.

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.

See here for a full technical explanation of SlowLoris attack.

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.

See here for a full technical explanation of HTTP/s Flood with browser enumeration attack.

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.

See here for a full technical explanation of HTTP GET Flood.

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.

See here for a full technical explanation of HTTPs Flood.

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.

See here for a full technical explanation of THC-SSL attack.

Conclusion

Most of the DDoS mitigation vendors can protect you against these attacks. Mitigation systems are tuned to work across many different environments, but they are generally not fine-tuned enough to ensure they protect your individual environment, and on average your first test will show around a 45% DDoS Gap.

Threat actors' goals are to bring your IT infrastructure or site down for political, commercial, or financial motivations. The best protection for your organization is to ensure your system is hardened and robust. To achieve this, push the system to the limits before a real attack happens so you can quickly find the points of weakness.

Once areas of weaknesses have been identified in your mitigation strategy, work closely with your cloud providers, vendors, and your MSSPs to ensure that your mitigation system is robust and hardened for *your* environment.

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/
10. https://www.nsslabs.com/blog/analyst-insights/enterprises-are-not-properly-protected-against-ddos-attacks/