The Many Faces of DDoS: Variations on a Theme or Two

Lori MacVittie, 2010-16-12

Many denial of service attacks boil down to the exploitation of how protocols work and are, in fact, very similar under the hood. Recognizing these themes is paramount to choosing the right solution to mitigate the attack.

When you look across the "class" of attacks used to perpetrate a denial of service attack you start seeing patterns. These patterns are important in determining what resources are being targeted because it provides the means to implement solutions that mitigate the consumption of those resources while under an attack. Once you recognize the underlying cause of a service outage due to an attack you can enact policies and solutions that mitigate that root cause, which better serves to protect against the entire class of attacks rather than employing individual solutions that focus on specific attack types. This is because attacks are constantly evolving, and the attacks solutions protect against today will certainly morph into a variation on that theme, and solutions that protect against specific attacks rather than addressing the root cause will not necessarily be capable of defending against those evolutions.

In general, there are two types of denial of service attacks: those that target the network layers and those that target the application layer. And of course as we've seen this past week or so, attackers are leveraging both types simultaneously to exhaust resources and affect outages across the globe.

NETWORK DoS ATTACKS

Network-focused DoS attacks often take advantage of the way network protocols work innately. There's nothing wrong with the protocols, no security vulnerabilities, nada. It's just the way they behave and the inherent trust placed in the communication that takes place using these protocols. Still others simply attempt to overwhelm a single host with so much traffic that it falls over. Sometimes successful, other times it turns out the infrastructure falls over before the individual host and results in more a disruption of service than a complete denial, but with similar impact to the organization and customers.

SYN FLOOD

A SYN flood is an attack against a system for the purpose of exhausting that system's resources. An attacker launching a SYN flood against a target system attempts to occupy all available resources used to establish TCP connections by sending multiple SYN segments containing incorrect IP addresses. Note that the term SYN refers to a type of connection state that occurs during establishment of a TCP/IP connection. More specifically, a SYN flood is designed to fill up a SYN queue. A SYN queue is a set of connections stored in the connection table in the SYN-RECEIVED state, as part of the standard three-way TCP handshake. A SYN queue can hold a specified maximum number of connections in the SYN-RECEIVED state. Connections in the SYN-RECEIVED state are considered to be half-open and waiting for an acknowledgement from the client. When a SYN flood causes the maximum number of allowed connections in the SYN-RECEIVED state to be reached, the SYN queue is said to be full, thus preventing the target system from establishing other legitimate connections. A full SYN queue therefore results in partially-open TCP connections to IP addresses that either do not exist or are unreachable. In these cases, the connections must reach their timeout before the server can continue fulfilling other requests.

ICMP FLOOD (Smurf)

The ICMP flood, sometimes referred to as a Smurf attack, is an attack based on a method of making a remote network send ICMP Echo replies to a
single host. In this attack, a single packet from the attacker goes to an unprotected network's broadcast address. Typically, this causes every
machine on that network to answer with a packet sent to the target.

UDP FLOOD

The UDP flood attack is most commonly a distributed denial-of-service attack (DDoS), where multiple remote systems are sending a large flood of UDP packets to the target.

UDP FRAGMENT

The UDP fragment attack is based on forcing the system to reassemble huge amounts of UDP data sent as fragmented packets. The goal of this attack is to consume system resources to the point where the system fails.

PING of DEATH

The Ping of Death attack is an attack with ICMP echo packets that are larger than 65535 bytes. Since this is the maximum allowed ICMP packet size, this can crash systems that attempt to reassemble the packet.

NETWORK ATTACK THEME: FLOOD

The theme with network-based attacks is “flooding”. A target is flooded with some kind of traffic, forcing the victim to expend all its resources on processing that traffic and, ultimately, becoming completely unresponsive. This is the traditional denial of service attack that has grown into distributed denial of service attacks primarily because of the steady evolution of web sites and applications to handle higher and higher volumes of traffic. These are also the types of attacks with which most network and application components have had long years of experience with and are thus well-versed in mitigating.

APPLICATION DoS ATTACKS

Application DoS attacks are becoming the norm primarily because we’ve had years of experience with network-based DoS attacks and infrastructure has come a long way in being able to repel such attacks. That and Moore’s Law, anyway. Application DoS attacks are likely more insidious simply because like their network-based counterparts they take advantage of application protocol behaviors but unlike their network-based counterparts it requires far fewer clients to overwhelm a host. This is part of the reason application-based DoS attacks are so hard to detect – because there are fewer clients necessary (owing to the large chunks of resources consumed by a single client) they don’t fit the “blast” pattern that is so typical of a network-based DoS. It can take literally millions of ICMP requests to saturate a host and its network, but it requires only tens of thousands of requests to consume the resources of an application host such that it becomes unreliable and unavailable. And given the ubiquitous nature of HTTP – over which most of these attacks are perpetrated – and the relative ease with which it is possible to hijack unsuspecting browsers and force their participation in such an attack – an attack can be in progress and look like nothing more than a “flash crowd” – a perfectly acceptable and in many industries desirable event.

A common method of attack involves saturating the target (victim) machine with external communications requests, so that the target system cannot respond to legitimate traffic, or responds so slowly as to be rendered effectively unavailable. In general terms, DoS attacks are implemented by forcing the targeted computer to reset, or by consuming its resources so that it can no longer provide its intended service, or by obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately.

HTTP GET FLOOD

An HTTP GET flood is as exactly as it sounds: it’s a massive influx of **legitimate** HTTP GET requests that come from large numbers of users, usually connection-oriented bots. These requests mimic legitimate users and are nearly impossible for applications and even harder for traditional security components to detect. This result of this attack is similar to the effect: server errors, increasingly degraded performance, and resource exhaustion. This attack is particularly dangerous to applications deployed in cloud-based environments (public or private) that are enabled with auto-scaling policies, as the system will respond to the attack by launching more and more instances of the application. Limits must be imposed on auto-scaling policies to ensure the financial impact of an HTTP GET flood does not become overwhelming.

SLOW LORIS

Slowloris consumes resources by “holding” connections open by sending partial HTTP requests. It subsequently sends headers at regular intervals to keep the connections from timing out or being closed due to lack of activity. This causes resources on the web /application servers to remain dedicated to the clients attacking and keeps them unavailable for fulfilling legitimate requests.

SLOW HTTP POST

A slow HTTP Post is a twist on Slow Loris in which the client sends POST headers with a legitimate content-length. After the headers are sent the message body is transmitted at slow speed, thus tying up the connection (server resources) for long periods of time. A relatively small number of clients performing this attack can effectively consume all resources on the web / application server and render it useless to legitimate users.

APPLICATION ATTACK THEME: SLOW

Notice a theme, here? That’s because clients can purposefully (and sometimes inadvertently) affect a DoS on a service simply by filling its send/receive queues slowly. The reason this works is similar to the theory behind SYN flood attacks, where all available queues are filled and thus render the server incapable of accepting/responding until the queues have been emptied. Slow pulls or pushes of content keep data in the web/application server queue and thus “tie up” the resources (RAM) associated with that queue. A web/application server has only so much RAM available to commit to queues, and thus a DoS can be affected simply by using a small number of v e r y slow clients that do little other than tie up resources with what are otherwise legitimate interactions.

While the HTTP GET flood (page flood) is still common (and works well) the “slow” variations are becoming more popular because they require fewer clients to be successful. Fewer clients makes it harder for infrastructure to determine an attack is in progress because historically flooding using high volumes of traffic is more typical of an attack and solutions are designed to recognize such events. They are not, however, generally designed to recognize what appears to be a somewhat higher volume of very slow clients as an attack.

THEMES HELP POINT to a SOLUTION

Recognizing the common themes underlying modern attacks are helpful in detecting the attack and subsequently determining what type of solution is necessary to mitigate such an attack. In the case of flooding, high-performance security infrastructure and policies regarding transaction rates coupled with rate shaping based on protocols can mitigate attacks. In the case of slow consumption of resources, it is generally necessary to leverage a high-capacity intermediary that essentially shields the web/application servers from the impact of such requests, coupled with emerging technology that enables a context-aware solution better detect such attacks and then act upon that knowledge to reject them.

When faced with a new attack type, it is useful to try to determine the technique behind the attack – regardless of implementation – as it can provide the clues necessary to implement a solution and address the attack before it can impact the availability and performance of web applications. It is important to recognize that solutions only **mitigate** denial of service attacks. They cannot **prevent** them from occurring.

- What We Learned from Anonymous: DDoS is now 3DoS
- There Is No Such Thing as Cloud Security
- Jedi Mind Tricks: HTTP Request Smuggling
- When Is More Important Than Where in Web Application Security
- Defeating Attacks Easier Than Detecting Them
- The Application Delivery Spell Book: Contingency
- What is a Strategic Point of Control Anyway?
- Layer 4 vs Layer 7 DoS Attack
- Putting a Price on Uptime
- Why Single-Stack Infrastructure Sucks

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com