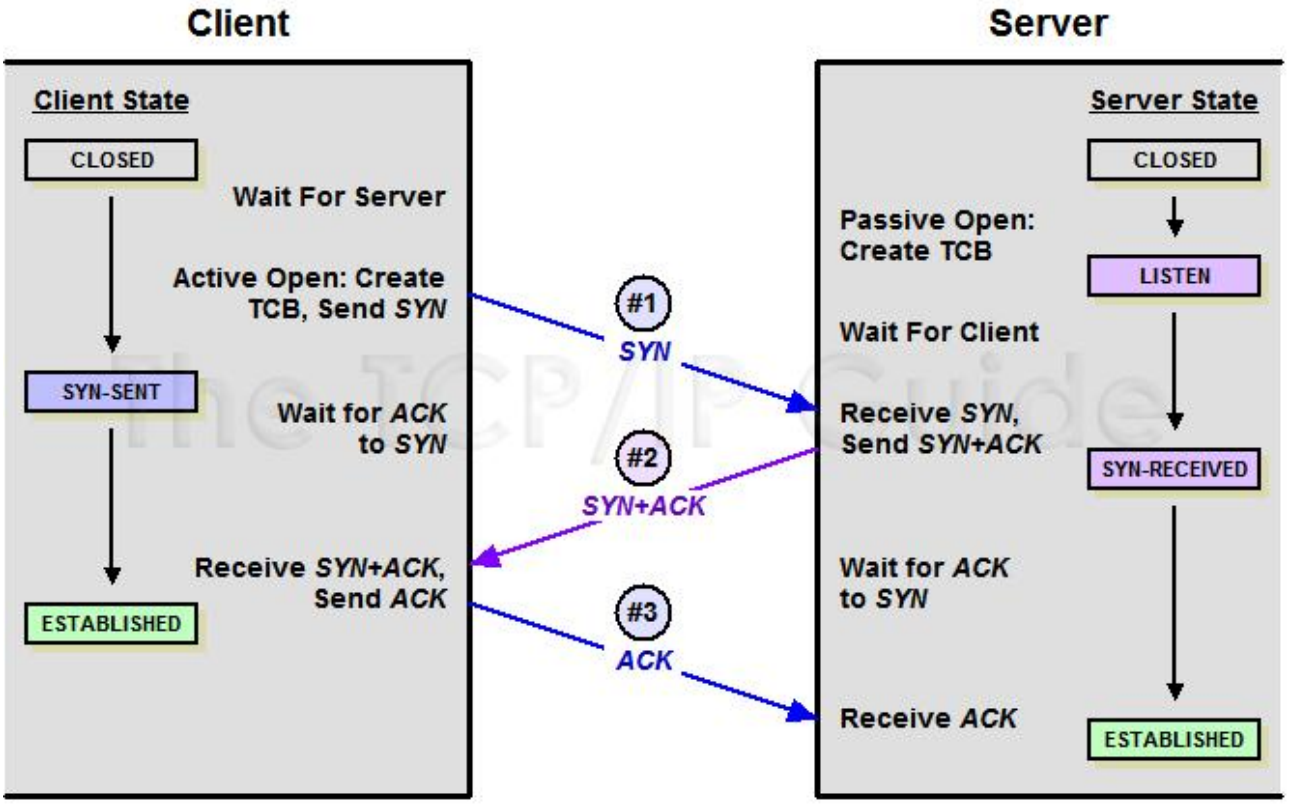
DDoS Attacks on SSL: Something Old, Something New

By ASERT team

Published: 2012-04-24 · Archived: 2026-04-06 02:09:06 UTC

SSL (or TLS) secures web services such as banking, online purchases, email and remote access. Popular services such as Twitter, Hotmail and Facebook are increasingly migrating to SSL to improve security and address privacy concerns. As more transactions and services are protected by SSL, DDoS attacks on SSL secured services are on the rise and are justifiably getting more attention. Some of these attacks are actually standard flood and TCP connection based attacks that have been used for years to disrupt both secured and clear text services. We also see attacks targeting SSL itself. Let's take a look at how attackers are using old and new methods to disrupt SSL protected services.

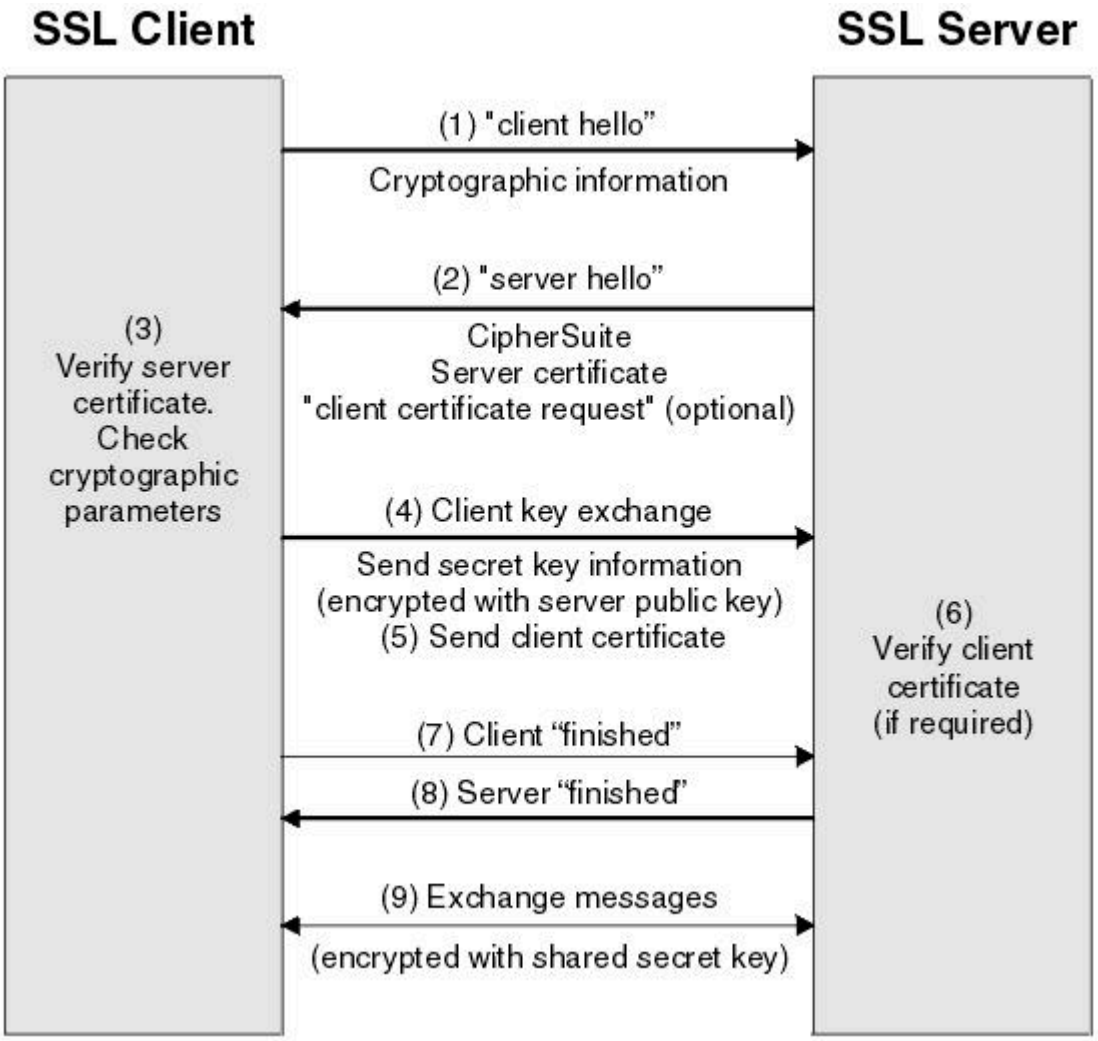
Communication between a client out on the internet and a data center server begins (in most cases) with the traditional TCP handshake. This is true for both SSL secured communications and non-secured.



The TCP layer is a very common target of DDoS. There are various flavors of these attacks but they share one aspect in common – the attack target is the capacity of the infrastructure to support concurrent TCP connections. Regardless of how many servers or how robust the infrastructure (firewalls, load balancers) there is a finite capacity to maintain TCP connections. One of the most common of this type of attack is the well-known Syn Flood, where attackers initiate enough connection open requests ("SYNs") without completing the handshake to exhaust that capacity. A variant of this method of attack is for botnetted hosts to open large numbers of TCP

connections simultaneously and actually complete the TCP handshake, thereby bypassing standard Syn Flood protections. Another means of bypassing traditional SYN-flood protections is Slowloris and its variants, which also complete the TCP handshake but then send a request to the server very slowly, one byte at a time, never actually completing the request.

Once the TCP handshake is completed there is a network layer session available for the SSL handshake to take place. The purpose of this exchange is to validate the authenticity of the parties and to establish the encryption key and options that will secure the subsequent communications. The SSL handshake is shown below.



There are numerous known and potential attacks which exploit the SSL handshake to exhaust server resources. The Pushdo botnet accomplishes this quite easily by sending garbage data to a target SSL server. The SSL protocol is computationally expensive and it generates extra workload on the server to process garbage data as a legitimate handshake. Firewalls don't help in this case because the clients have completed the TCP handshake and are sending traffic to an allowed service.

Another SSL-based attack tool is the THC-SSL-DOS tool, which works by completing a normal SSL handshake but then immediately requests a renegotiation of the encryption method. As soon as the renegotiation completes, it requests another renegotiation, and so on. If the server has SSL renegotiation disabled (a standard security best practice), then the tool simply closes the SSL connection as soon as the negotiation completes and opens a new

connection to start the negotiation process all over again. This is extremely computationally expensive and is effective at making services unavailable to legitimate users due to resource exhaustion. There are numerous other potential attacks that target various aspects of the SSL negotiation process to cause server overload and denial of service. The diagram below is a simplified view of the infrastructure data centers use to provide ecommerce, email or other services protected by SSL.

What is the DDoS attack surface in this infrastructure? First off, the entire data center can be cut off from the outside world through very high volume traffic floods that saturate the incoming links from the internet. Assuming the data center has a provider capable of detecting and screening those types of attacks, what comes next? The firewall is the next target, prone to TCP state exhaustion attacks. Similarly the load balancer/SSL Offload devices are vulnerable. Both maintain tables that track ongoing TCP sessions. In the face of TCP based attacks these devices may become overwhelmed, causing them to stop accepting new connections, remove existing connections or even crash. These actions effectively accomplish the purpose of the attack. Further up the stack, devices supporting SSL and actual application services are attack targets in themselves and have additional application-layer vulnerabilities such as the SSL attacks discussed above. Most firewalls, ADCs, and WAFs include some DDoS protections yet many high end data centers with the most up to date infrastructure have fallen victim to DDoS.

Why do DDoS attacks continue to succeed?

- Detection is reactive – if attacks are detected based on session tables filling up, server response times rising, etc.
- DDoS attacks (by definition) are distributed. What is normal and acceptable behavior from a single session becomes an attack when repeated by thousands of sources. Firewalls, ADCs view traffic on a session by session basis. •
- Blended attacks are effective because each element in the infrastructure is dedicated to performing a particular function.
- There is a lot of NAT out there. DDoS protections built into firewalls and ADCs are heavily based on behavioral attributes of the requesting hosts – e.g. how many sessions from a given source IP. With more and more NAT'd and proxied sources (inside enterprise networks, behind carrier grade NAT, Content Delivery Services) behavioral methods have a hard time teasing out the bad from the good.

What is Arbor's approach?

- Put DDoS protection at the data center edge – in front of the DDoS attack surface.
- Be as invisible as possible – not part of the attack surface.
- Multiple levels of detection. Use individual host behavior, aggregate behavior of multiple hosts, known signatures and attributes of botnet traffic, IP location, reputation, etc.
- Multiple levels of mitigation. Packet based, header based, behavioral, challenge response techniques that indentify infected hosts and spoofed addresses, white and black lists.
- Automate as much as possible, provide manual controls, and report on what is going on (where traffic is coming from, going where, what is requested, rates, what was blocked, what was passed). In short, stop attacks before they reach the attack surface and enable the data center to do what it was designed for.
