TCP HALF OPEN CONNECTION

By Naveen PS

MAKERDEMY
LEARN EXPLORE MAKE

TCP HALF OPEN CONNECTION

What are Half Closed Connections? It's Not a Bug--it's a Feature!

Every TCP connection consists of two half-connection which are closed independently of each other. So, if one end sends a FIN, then the other end is free to just ACK that FIN (instead of FIN+ACK-ing it), which signals the FIN-sending end that it still has data to send. So, both ends end up in a stable data transfer state other than ESTABLISHED--namely FIN_WAIT_2 (for the receiving end) and CLOSE_WAIT (for the sending end). Such a connection is said to be half closed and TCP is actually designed to support those scenarios, so half closed connections is a TCP feature.

TCP has a vulnerability in that the final FIN packet sent to a client can be potentially dropped by routers/networks resulting in a connection that is half-open when the actual intention was to fully close the connection. This and similar approaches have been popular types of Denial of Service attacks as they do not require a lot of bandwidth, yet potentially eat-up valuable handles, sockets, and threads depending on the server implementation, but they can also happen in the real world with increasing frequency thanks to our shoddy wireless carriers.

Operating systems have made attempts to fight back against half-open DDoS attacks by limiting the number of half-open/closed connections that can be present in the operating system at a given time and by introducing maximum lengths of time that connections can remain in a half-open/closed state.

This condition is further aggravated by the optional nature of TCP keep-alive, which if fully-implemented were intended as a protocol-level (as opposed to application level) solution to detecting dead/zombie connections. But, when TCP was designed, bandwidth was considerably more precious than it is now, and there were concerns that mandatory keep-alive timers for TCP would be too "chatty". Therefore keep-alive are optional, not generally used, and not guaranteed to be transmitted by routers according to RFC1122. So, even if you enable keep-alive at the TCP layer in an attempt to detect/handle the scenario,

you may find that as your traffic travels around the world, some routers are dropping the keep-alive packets, creating potentially ANOTHER rare scenario to test.

When TCP establishes a connection, it is considered guaranteed since there is a handshake that takes place:

- The initiating computer sends the Connection request, sending a SYN
- The responding computer grants the request, replying with a SYN-ACK
- The initiating computer sends an acknowledgment, replying with an ACK

At that point the connection is established, and data begins to flow. In contrast, a UDP packet is not guaranteed, and is just sent in the hopes it gets there.

Officially, according to the RFC's, a half-open TCP connection is when one side of the established connection has crashed, and did not send notification that the connection was ending. This is not the common usage today.

THE END