SSL: On the Failure of False Start

David Holmes, 2012-17-04

In Adam Langley’s blog this week (False Start’s Failure), he announced that Google will end support for the False Start modification to the SSL protocol, except for servers that support the Next Protocol Negotiation (NPN) extension. You may recall that when False Start was first announced, many people, including myself, were initially skeptical. Since then, Google and F5 haven’t exactly seen eye-to-eye on SSL and there’s been some public back and forth.

Before I go any further, let me make one thing clear. I admire Adam Langley, and Google, for trying to make SSL (and by extension, the Internet), faster and better. They had a bold idea and a pretty good plan for execution. Ultimately, it didn’t pan out but it was an idea worth trying.

While there were several issues with False Start, Adam cites issues with other SSL providers as the major reason for its failure. I checked with him to see if F5 was included in that category and he said he hadn’t heard of any problems with F5 hardware with regard to False Start or otherwise. The problems with other vendors appeared to be how they processed incoming, and outgoing, SSL records with different threads, leading to contextual bindings that weren’t compatible with False Start. I’m not sure that anyone saw that coming, but I’m not really surprised about it. Different vendors will process SSL in different ways. F5 doesn’t use threading in our high-performance microkernel which is probably why our stack appeared to work well with False Start.

Incompatibly issues weren’t what made me uneasy about False Start though; it was the practice of tinkering with the ordering of messages for performance sake that gave me pause. The Finished message in the SSL handshake is a big deal, I mean, it really is. The fact that SSLv2 doesn’t have that message is what makes that protocol insecure, so tinkering with its usage is significant. Adam doesn’t cite any security problems seen in the wild with the way that False Start clients handled the Finished message, but I heard someone say at a recent security conference that False Start would have made any TLS1.1 clients vulnerable to the BEAST attack. I haven’t done an analysis to see if that’s true or not (I doubt it is) and the end of False Start probably makes this a moot point.

Whether we're talking about compatibility or security, here's a take-away for you: SSL really is hard to do right, and there's a reason that so many of the fortune 1000 trust F5 to terminate their SSL. We do it well, and we've been doing it for over 12 years.

Regarding the future of the SSL protocol, there are a lot of interesting improvements coming in the form of new extensions. Server Name Indication (SNI) is a good one, Next Protocol Negotiation (NPN) shows definite promise, and DNS-based Authentication of Name Entities (DANE) is gaining some traction. F5 will continue to embrace the changes that we see as safe, and look closely at the ones that give us pause.