Electronic Design

By DAVE HUGHES, CEO, HCC Embedded,
https://www.hcc-embedded.com

11 Myths About TLS

Transport Layer Security is the most widely used approach for security on the Internet—the majority of secure transactions depend on it. But myths about TLS still abound.

Security issues are persistently front and center when it comes to the internet, and Transport Layer Security (TLS) often is the go-to solution. Nonetheless, myths surround the technology. HCC Embedded CEO Dave Hughes looks to dispel some of these misconceptions.

1. TLS is broken and can't provide adequate protection against hackers.

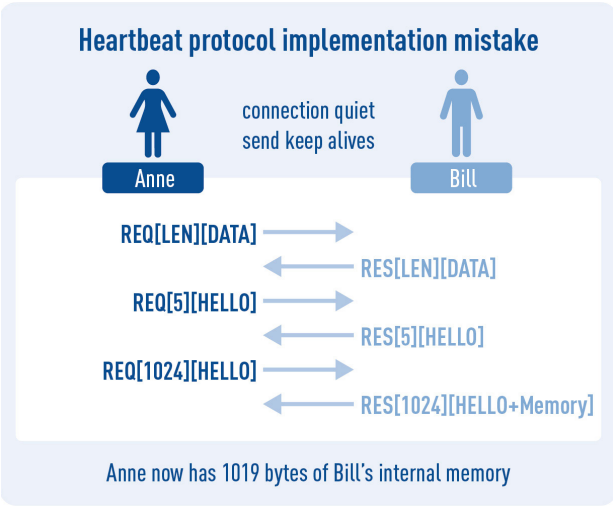
Hearing about widely publicized security breaches, you would think that those designing security are incompetent. This is simply not the case. The truth is, there are no known hacks of TLS 1. Rather, these hackers were successful not due to faulty TLS, but because of a lack of software-quality processes.

For example, a well-designed static-analysis tool would have detected Apple's 2017 TLS vulnerability before it was released. And the Heartbleed Bug, which resulted from an implementation defect in some OpenSSL versions, was caused by software that did not check the scope of a protocol variable and then processed it blindly.

Software-quality processes that include unit testing and boundary case analysis/testing would have instantly alerted developers to the issue, and the detection would have been reinforced by other requirements of the lifecycle process.

2. TLS 1.2 is perfect and will always protect you.

This is never going to be true, but the main criticism facing TLS (and all attempts at security) is that it can be difficult to use safely in real-world environments. This has been demonstrated by the stream of security failures. However, as stated in Myth 1, these protocols can only be effective if they're implemented properly, using proven software-quality processes.

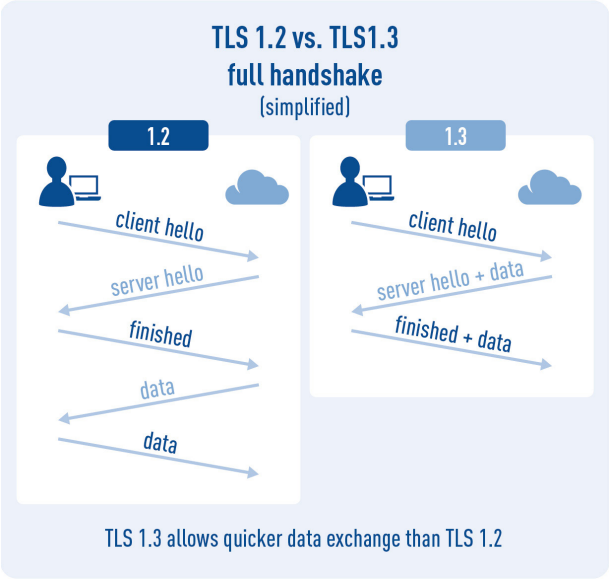


3. TLS 1.3 will fix problems with TLS 1.2.

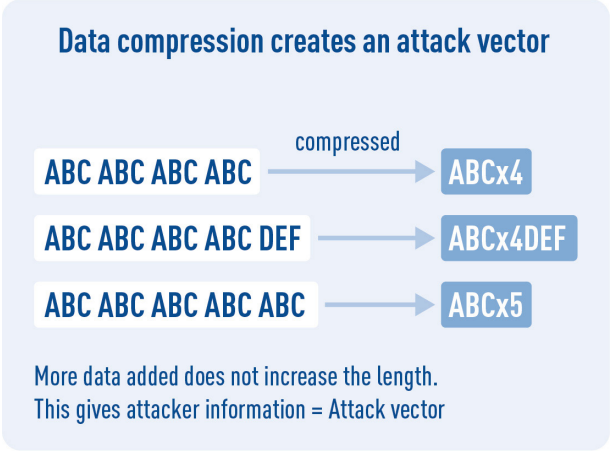
TLS 1.3 is really an enhancement of 1.2, reducing information leakage and cleaning things up rather than fixing TLS 1.2. While it may not fix all problems, the main things that TLS 1.3 enhances are security and connection speed. Security is improved both through the algorithms used as well as the TLS protocol exchanges, which give fewer attack vectors. The speed of TLS has also been improved by simplifying certain key protocol exchanges. Nothing was actually broken in TLS 1.2—it's just a continuous struggle to keep TLS ahead of the bad guys and to improve the user experience.

4. TLS and network security are all about cryptography.

Most recent network security failures have been caused by either the leak of key information by humans, badly written code, or poor integration of the security layer. While some cryptographic algorithms have been weakened in lab conditions, practical attacks rarely exploit these weaknesses.



TLS 1.3 allows quicker data exchange than TLS 1.2



5. Cryptographic algorithms can't be broken.

When using currently available computing power, this is probably true. However, computing power, particularly that available to nation-states, is increasing rapidly and consequently must be kept under observation. Embedded developers first need to assess their risks. If you're really concerned about being hacked by a nation-state, then you need to take different measures than if you're trying to protect your data from a company trying to extract competitor information. Many would argue that if a nation-sate wants to hack you, you have bigger problems to address.

6. Adding compression makes encryption more secure.

Surprisingly, the opposite is the case. The reason is that compression reduces the size by keeping dictionaries of common strings. Therefore, by injecting strings and checking the change in size, you can find out how much that data is used. That's not an easy thing to process, but it's a very useful attack vector for the determined hacker.

7. Properly implemented TLS has no information leakage.

This isn't true, and although desirable, "zero information leakage" was not a goal of TLS. The types of leakage that occur can include the size of messages being conveyed and the fact that a conversation between two people has occurred. Developers need to decide if these are real risks for their system and take additional measures as appropriate. The use of VPNs, for example, can provide additional levels of security.

8. Web browsers mostly now use state-of-the-art security.

State-of-the-art algorithms are available for most browsers, but they often negotiate down to older versions where required. Web browsers all fight for market share, so they want backwards compatibility with as many servers as possible. On the other hand, they all want to be very secure, so they have a difficult balance to strike. But as a matter of course, developers should all use the latest versions of TLS (1.2 and/or 1.3) and not allow this to be negotiated down. All modern browsers support at least TLS 1.2.

9. Side-channel attacks are commonplace.

Side-channel attacks have been used to break crypto-algorithms. To execute them, though, you normally need to be physically local to your target and have the ability to execute millions of test cases without anyone noticing. At best, it's useful for targeting a single specified target.

10. TLS will never be broken.

This seems unlikely. But since both computing power and the imagination of hackers continues to rise, it's possible that a point will be reached where trying to break TLS isn't worth it. This seems unlikely. Rather, it seems inevitable that breaches of TLS will occur and improvements will be required.

However, as the protocol and algorithms are improved, there may come a point where hacking the protocol is too difficult for most to attempt—and that other methods of attacking systems will be preferred. The most obvious example of this is social engineering, where an attacker works in other ways to find system attack vectors. An example of this might involve befriending someone and getting likely password possibilities from him or her. The point is that the users and system managers will have become the weakest point in the system.

11. Security is a perfected art.

The expectation that there's a foolproof solution to security is naive. Most high-profile security breaches have come from three main sources: insiders divulging secrets, poor system management, and badly or inappropriately written software.

The first two sources can only be dealt with by the organizations responsible for the security of the information. Clearly, there are no easy solutions where humans are involved.

But the third source—software quality—is often neglected, especially with rapidly increasing networking requirements and new situations that create new risks. This is where developers should focus their efforts: Sourcing and creating high-quality software developed through the use of proven lifecycle processes.

Dave Hughes founded HCC Embedded, a leading developer of reusable embedded software components, in 2000. In that time, the company has grown substantially, supplying many of the industry's major technology providers. Dave is a "hands-on" embedded specialist, who still actively contributes to the strategy and direction of HCC's core technologies. His extensive experience has made him one of the industry's leading authorities on fail-safe embedded systems, flash memory, and process-driven software methodologies.

Dave is a graduate of the University of Sussex in England.

Reference:

Verifiable Embedded TLS/DTLS