Hochschule Offenburg
offenburg.university

Campus Gengenbach
Klosterstraße 14, 77723

Campus Offenburg
Badstraße 24, 77652

15.12.2016

ivESK researchers find critical security flaws in embedded TLS implementation MatrixSSL

TLS (or SSL, as its predecessor has been called) is a security protocol which set out to prevent attackers from eavesdropping or tampering with sensitive internet communication and is the cornerstone of online banking, shopping, etc. Unfortunately, TLS is a rather complex protocol and, hence, is prone to issues in its implementations. The history of TLS is peppered with a good number of implementation bugs undermining the protocol's security guarantees. The Heartbleed bug in OpenSSL or Apple's goto fail bug are certainly those with the most shady glory in that respect.

In the course of the implementation of the Internet of Things, TLS is increasingly deployed in the embedded world as well. This is where somewhat less popular implementations like MatrixSSL, mbedTLS, WolfSSL or the author's emb::TLS tender their service. All those strive for a small-sized and efficient implementation of TLS for constrained devices.

Given the challenge to implement TLS without security vulnerabilities and the potentially hazardous consequences of failing to do so, we started working on techniques to identify bugs in TLS implementations. We are going to publish a paper on our approach and our findings soon but, without going into too much detail here, we would like to share one particularly alarming discovery in advance.

In a rather recent version (3.8.4) of MatrixSSL's TLS server implementation we found a number of issues related to the processing of a client's initial TLS message, the so called ClientHello message. Though each of these issues alone might be hard to exploit, it is their combination that provides an attacker with a fairly dangerous capability. TLS features a dynamic extension mechanism which allows clients and servers to indicate and negotiate their support for optional protocol features. TLS ensures that this process is protected against unauthorized manipulations. However, given the unfortunate combination of bugs in MatrixSSL, a man-in-the-middle attacker could be enabled to inject additional extensions into the TLS handshake process, which the MatrixSSL server processed as if they were coming from the unaware client. Neither side necessarily found out that they were operating on a different set of extensions.

MatrixSSL's developer team has been informed about this issue and published a fix that is available since version 3.8.6 of MatrixSSL.