Ivan Miller. Reading Summary “Using Frankencerts for Automated Adversarial Testing of Certificate Validation in SSL/TLS Implementations”. 11.01.2022

1. Describe the problem the paper is trying to address.

Internet security relies on correct validation of X.509 certificates presented by servers during the SSL/TLS handshake protocol, which is used by Web, mobile, enterprise, and embedded software to provide end-to-end confidentiality, integrity, and authentication for communication over insecure networks. Implementation of SSL/TLS from scratch is extremely complex, so multiple open-source implementations are available for developers who need to incorporate it into their software.
However, a unified framework for large-scale adversarial testing of certificate validation logic in SSL/TLS implementations does not exist.
Testing correctness of the certificate validation logic is challenging because: 1) it is difficult to generate test inputs: it isn’t feasible to manually create high-quality samples while existing automated techniques could only produce a limited number of inputs 2) it is difficult to interpret the results of testing, since only the fact whether the certificate was accepted or rejected is being recorded and the correctness of the implementation of SSL/TLS is not being questioned. The paper proposed a framework for discovering security vulnerabilities across different SSL/TLS implementations and particularly focused on server authentication which protects from man-in-the-middle and other server impersonation attacks.

2. Describe some of the main ideas behind how the adversarial examples are generated.

The paper proposed an automated testing framework which allowed the generation of over 8M test certificates. The authors scanned the Internet to collect a corpus of real certificates, they broke them into parts and then combined those parts into random combinations of syntactically correct test certificates (“frankencerts”).
The frankencerts were then being used to perform differential testing on multiple open-source SSL/TLS libraries and Web browsers in order to identify cases when one implementation of SSL/TLS accepted a certificate while another rejected the same certificate (meaning that their implementations of the X.509 standard had to be semantically different). Each discrepancy triggered a manual analysis of the source code of the disagreeing implementations. The fact that the authors managed to create frankencerts that did not satisfy the protocol specification, allowed to significantly improve the chances of uncovering subtle implementation flaws.

3. Are there any findings here that you find surprising? Why or why not?

I was surprised to learn that automated adversarial testing is not mandatory performed on SSL/TLS implementations that is crucial to Internet security

4. Do you think there are likely to be fruitful extensions of this approach to software testing to other domains? Discuss.

The authors noted that they received positive feedback from the developers of SSL/TLS implementations about the issues they discovered and many of them had either fixed the issues or were working on implementing a fix. However, I am skeptical about the prospects of scaling this or any other approach to testing the implementation OF SSL/TLS without a comprehensive industry-wide standard of automated testing requirements to be applied to the SSL/TLS software stack.