SymCerts: Practical Symbolic Execution For Exposing Noncompliance in X.509 Certificate Validation Implementations

Sze Yiu Chau Omar Chowdhury Endadul Hoque Huangyi Ge Aniket Kate Cristina Nita-Rotaru Ninghui Li
Purdue University The University of Iowa The University of Kansas Northeastern University

(1) The need for secure communications

- SSL/TLS is now the de facto standard for achieving secure communication

![Percentage of Web Pages Loaded by Firefox Using HTTPS](chart)

(2) Why do we care about X.509 certificates?

- X.509 is used in SSL/TLS for Authentication and Key Distribution
- The security guarantees of SSL/TLS hinge on a correct implementation of the X.509 PKI

(3) How does X.509 work?

- Root CAs
- Intermediate CAs
- End Entities (e.g. Servers)

(4) Implications of bugs in X.509 implementations

- Overly Permissive
- Overly Restrictive
- Violating specifications can lead to 2 contrasting pitfalls

(5) Small Footprint SSL/TLS libraries for IoT

- SymCert
- SSL
- Symbolic
- Execution
- Solver
- SSL

(6) Research Problem

- Goal: Find RFC Violations in X.509 implementations made for IoT.
  - Related Work
    - SSL/TLS protocol state machine and bug finder
    - Cryptographic proofs and reworked state machine
    - Detect incorrect SSL/TLS API usage in applications

(7) Our approach

- X.509 Certificate Chain Input Universe
- SymCert
- Sets of Logical Formulas
- RFC 5280

(8) Making Symbolic Execution practical

- Focus our analysis on small-footprint, small code-base libraries
- Adding domain specific optimizations
  - Does not check cryptographic correctness
  - Concrete Length values in encoded SymCerts
  - Simplify strings (e.g. in name matching)

(10) Notable findings and their implications

- Misintrept UTCTime (MatrixSSL 3.7.2, axTLS 1.4 and 1.5.3, tropicSSL) → e.g. in MatrixSSL 3.7.2 expiration date can shift by 100 years
- Misinterpret OID of ExtKeyUsage (wolfSSL 3.6.6, MatrixSSL 3.7.2) → Overly Permissive (and compatibility issues with custom OID)
- Incorrect Extension Parsing (CyaSSL 2.7.0) → Crash
- Rejects GeneralizedTime (tropicSSL, axTLS 1.4.3) → Overly Restrictive
- Incomplete Extension Handling (various libraries) → Overly Permissive

Takeaway

- X.509 handling in IoT SSL/TLS libraries all deviate from specification
- If there is a vulnerability in the library, it's hopeless for Applications
- We provide automated approach and toolchain for finding violations
- Our experiments turn out to be quite prolific → many problems are fixed
- New versions of SSL/TLS libraries are generally better → Patch often!

* We thank the Purdue Research Foundation and the National Science Foundation for funding this project.

(9) Summary of Experiments and Findings

- We tested 9 implementations from 4 families of SSL/TLS libraries.
- Findings have been reported and well-received by library developers.
  → Many of the problems are fixed in new releases following our reports.