Analysis of control flow traces of network protocol states

M2 Research Project

Keywords : Active automata learning, Binary analysis

Supervisors : Yohan Pipereau (Télécom SudParis, Palaiseau), Olivier LEVILLAIN (Télécom SudParis, Palaiseau)

Context

In the context of networking protocol security, one of the challenge is to automatically verify formal properties of an implementation. Automatic verification is motivated by the diversity of implementations for a same a standard, and by frequent implementation changes which can introduce violations of logical properties. Automatic protocol verification could improve continuous integration pipelines in various project to detect logical problems.

Automatic protocol verification first extracts a model (as an automaton) out of an implementation (e.g. TLS [6]), then it proves correctness properties on the extracted state machine (e.g. Mealy Verifier [7]). State-machine extraction relies on *active automata learning* (AAL) algorithms such as L* [1] to extract the logical behavior of a protocol as an **I/O automaton** (or Mealy Machine). These algorithms have been successfully used to infer various implementations for various protocols (TLS [6], WPA [4], OPC-UA [7], ...).

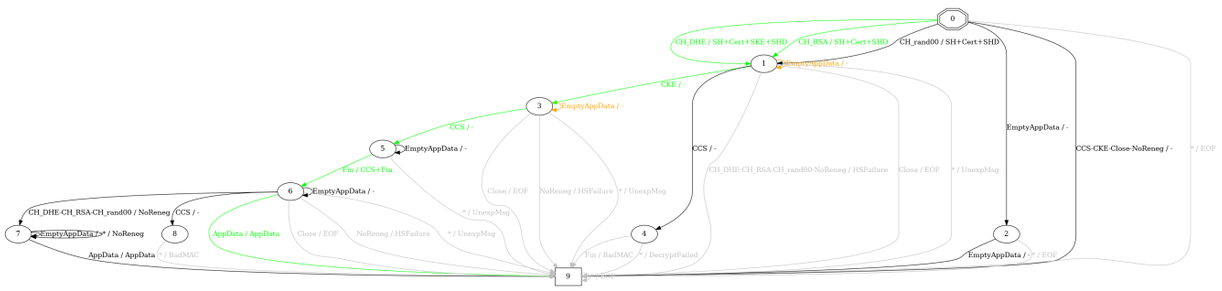

Figure 1: Mealy machine of openssl 1.0.1g for TLS 1.2 extracted with pylstar-tls [6]

Problem

One of the problem of AAL is that it is generally not exhaustive. In particular, it is not possible to know how many states are missing from the set of all logical states of an implementation. There are two main reasons why AAL may lack some logical states:

First, some states could be missed because the input and output vocabulary are incomplete. For example, in the context of TLS, there are many extensions which can be used and may lead to discover new logical states in the protocol.

Second, some states could be missed because the AAL has been configured with an upper-bound on the message sequence used to discover new states and this upper-bound is too low. Some inferred Mealy machines contain states which are very similar and can only be separated by crafting very long message sequence.

Short read list

The project includes reading one of the following paper:

- Learning regular sets from queries and counterexamples [2]
- REPT: Reverse Debugging of Failures in Deployed Software [3]
- The Closer You Look, The More You Learn: A Grey-box Approach to Protocol State Machine Learning [5]

Project proposal

We propose to use Intel Processor Trace (Intel PT), a hardware tracing mechanism which supports tracing of the program control flow. The idea is to identify additional logical states which may be missing from the AAL inference by building control flow subgraph from the traces and compare them with the program full

CFG to identify missing code paths. The approach could help to build a state coverage metric to compare AAL protocol implementations or to compare Stateful Fuzzers.

References

[1] Dana Angluin. Learning regular sets from queries and counterexamples. *Inf. Comput.*, 75(2):87–106, nov 1987.

[2] Dana Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2):87–106, 1987.

[3] Weidong Cui, Xinyang Ge, Baris Kasikci, Ben Niu, Upamanyu Sharma, Ruoyu Wang, and Insu Yun. REPT: Reverse debugging of failures in deployed software. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pages 17–32, Carlsbad, CA, October 2018. USENIX Association.

[4] Chris McMahon Stone, Tom Chothia, and Joeri de Ruiter. Extending automated protocol state learning for the 802.11 4-way handshake. In *Computer Security: 23rd European Symposium on Research in Computer Security, ESORICS 2018, Barcelona, Spain, September 3-7, 2018, Proceedings, Part I*, page 325–345, Berlin, Heidelberg, 2018. Springer-Verlag.

[5] Chris McMahon Stone, Sam L. Thomas, Mathy Vanhoef, James Henderson, Nicolas Bailluet, and Tom Chothia. The closer you look, the more you learn: A grey-box approach to protocol state machine learning. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 2265–2278, New York, NY, USA, 2022. Association for Computing Machinery.

[6] Aina Toky Rasoamanana, Olivier Levillain, and Hervé Debar. Towards a systematic and automatic use of state machine inference to uncover security flaws and fingerprint TLS stacks. In *27th European Symposium on Research in Computer Security (ESORICS)*, volume 13556 of *Lecture Notes in Computer Science*, pages 637–657, Copenhague, France, September 2022. Springer Nature Switzerland.

[7] Arthur Tran Van, Olivier Levillain, and Hervé Debar. Mealy verifier: An automated, exhaustive, and explainable methodology for analyzing state machines in protocol implementations. In *Proceedings of the 19th International Conference on Availability, Reliability and Security, ARES 2024, Vienna, Austria, 30 July 2024 - 2 August 2024*, pages 16:1–16:10. ACM, 2024.