Stateful Fuzzing of OPC UA

SECITC 2024

International Conference on Security for Information Technology and Communications

20th of November - “Ferdinand I” Military Technical Academy

Cristian Daniele, Mark Fijneman and Erik Poll

Radboud University, Netherlands

What's fuzzing?

Pretty old testing technique:

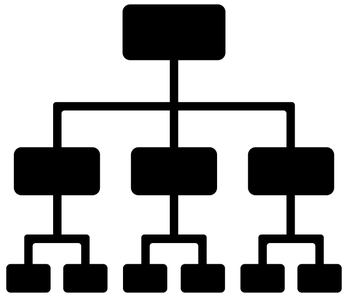
1988 - first fuzzer

2013 - AFL

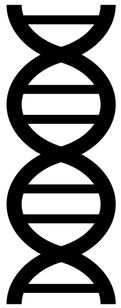
2015 - OSS-Fuzz

… and very effective!

Different approaches

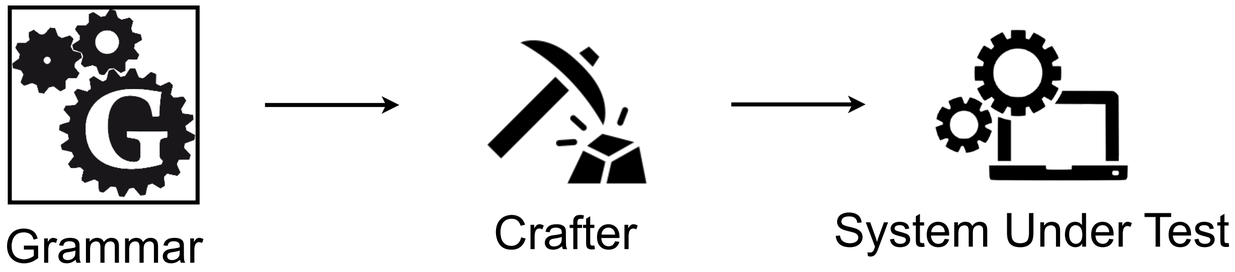


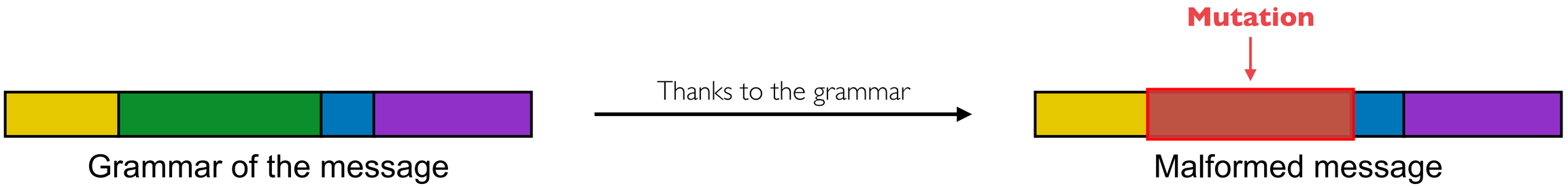
Grammar-based



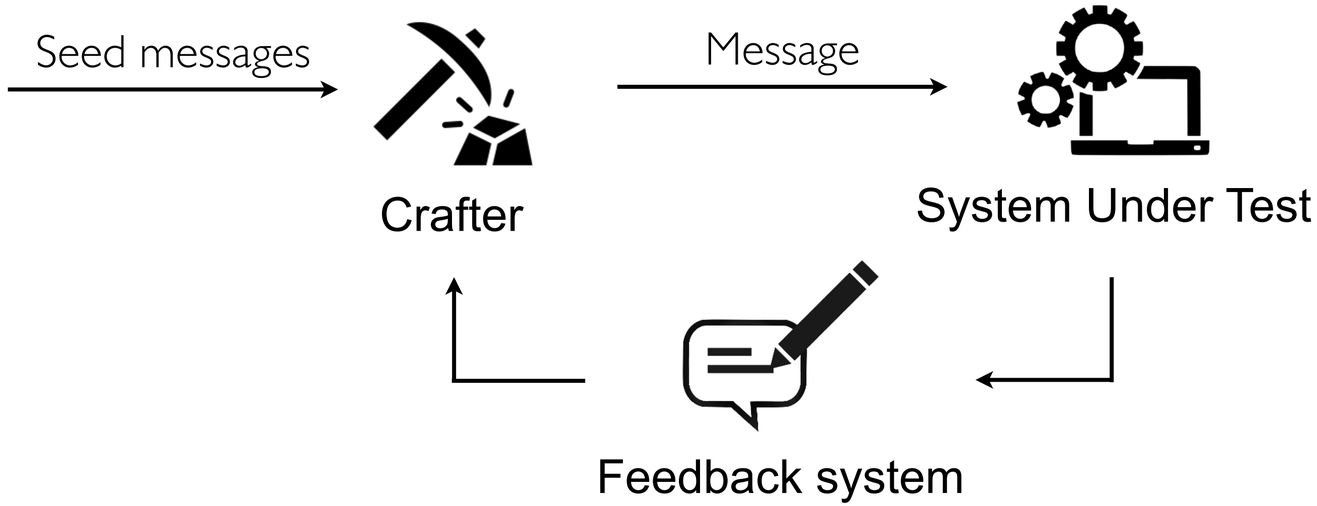
Grey box

How do grammar-based fuzzers work?



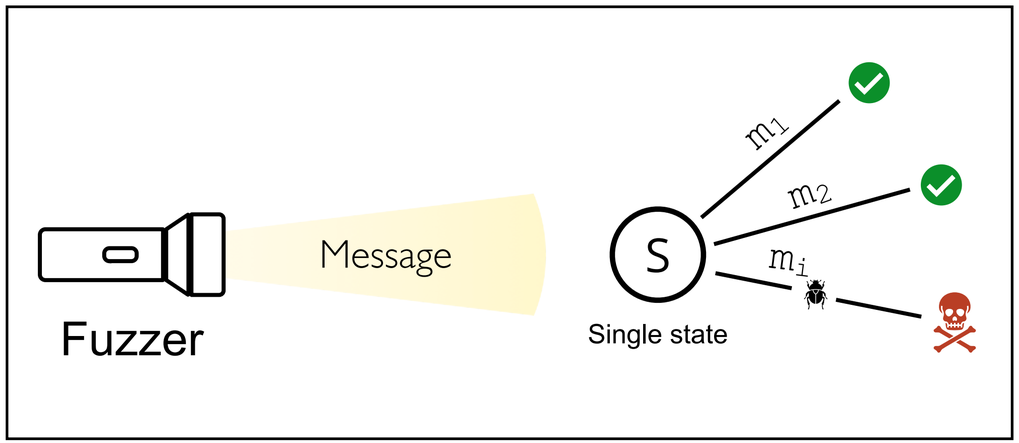


How do grey box fuzzers work?

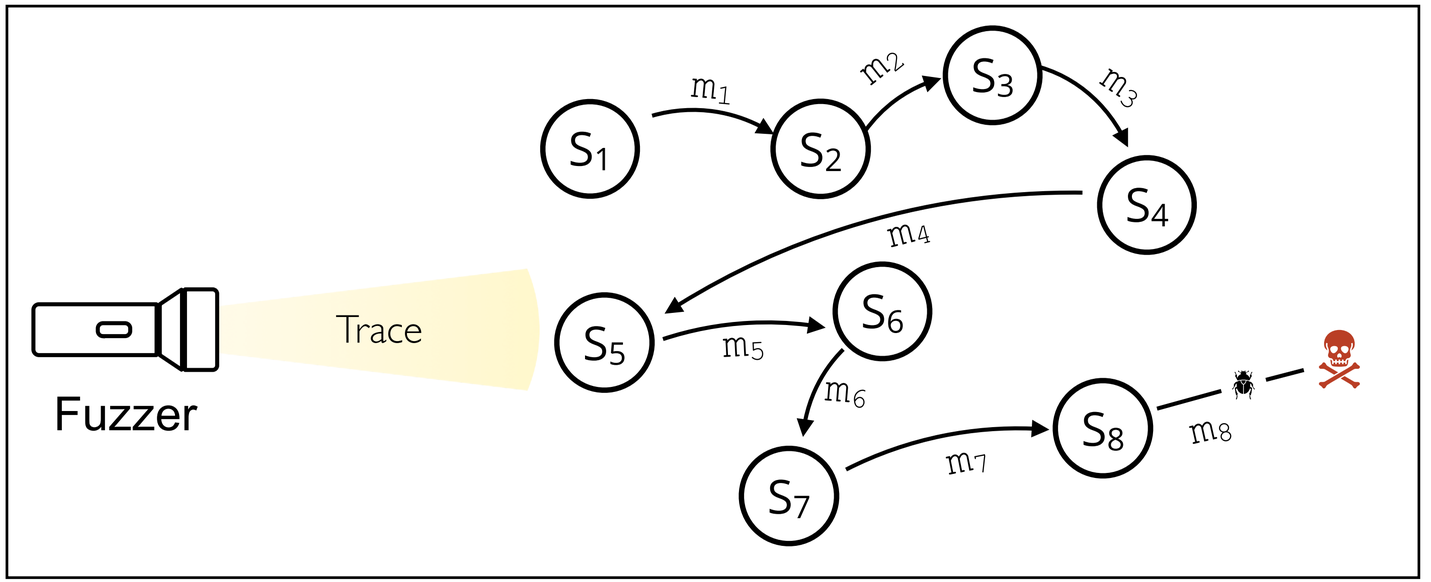


Different targets

Stateless fuzzing



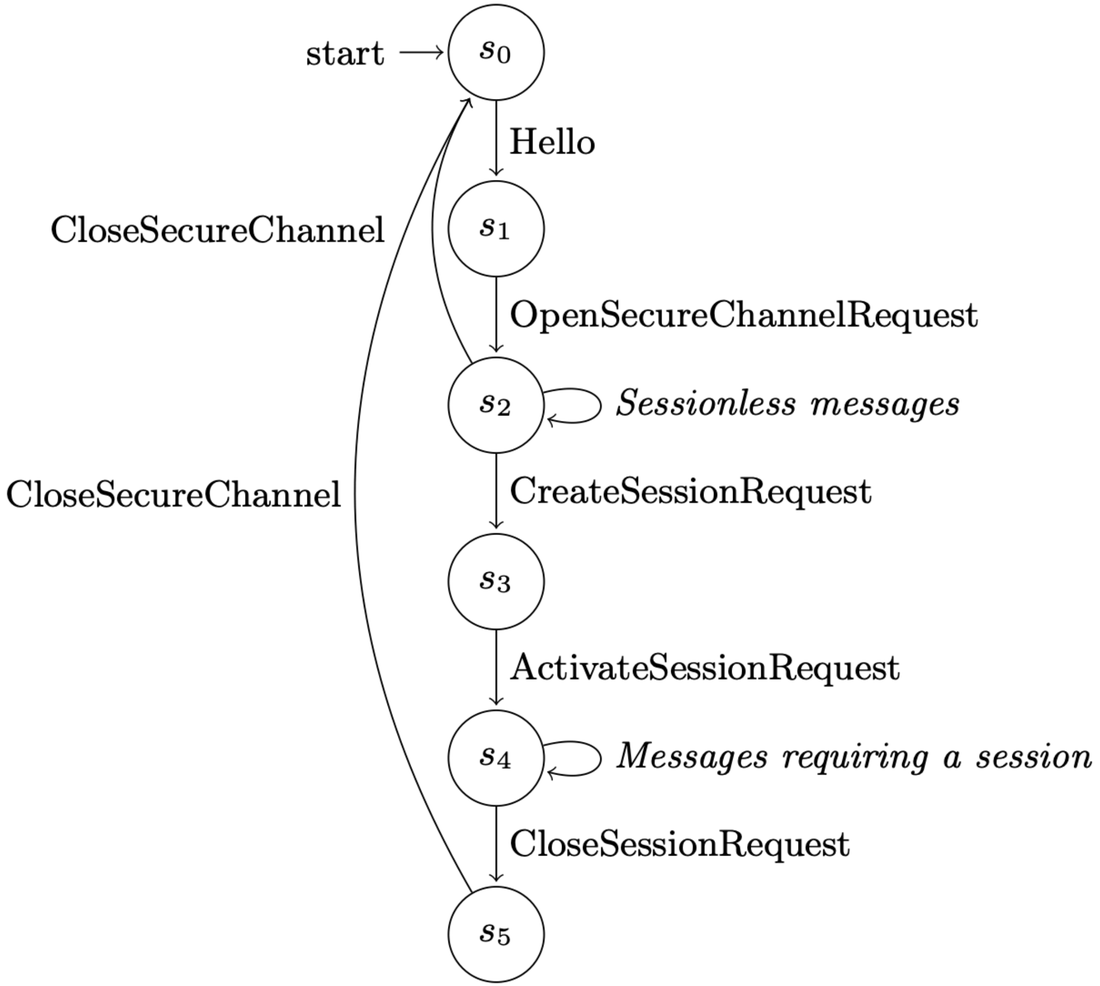
Stateful fuzzing



Nature of OPC UA protocol

OPC UA is a stateful protocol used for the communication between Industrial Control Systems (ICS).

It is widely used in the industry, and several implementations are available open source



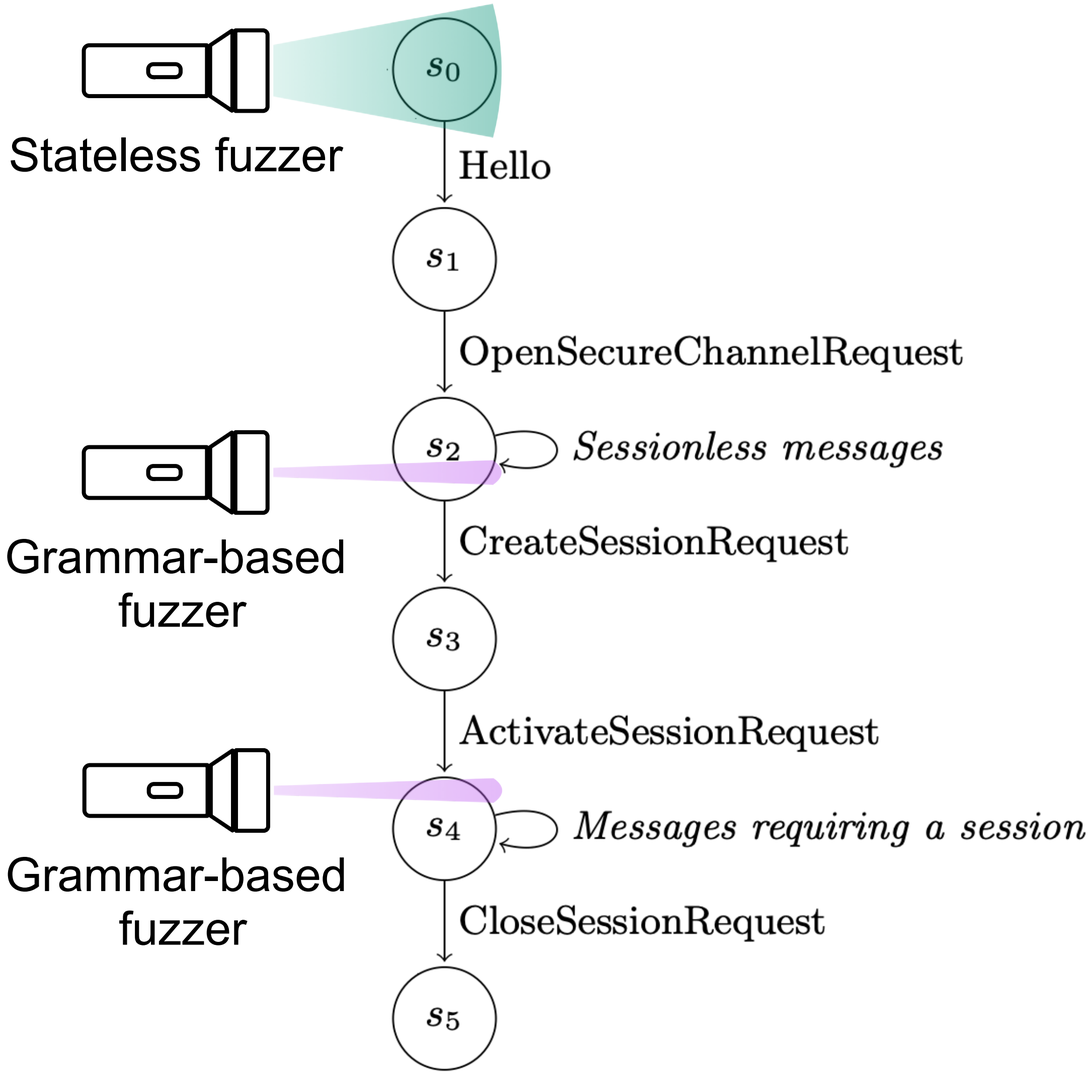
Approaches used in the past

(1)Stateless fuzzers:

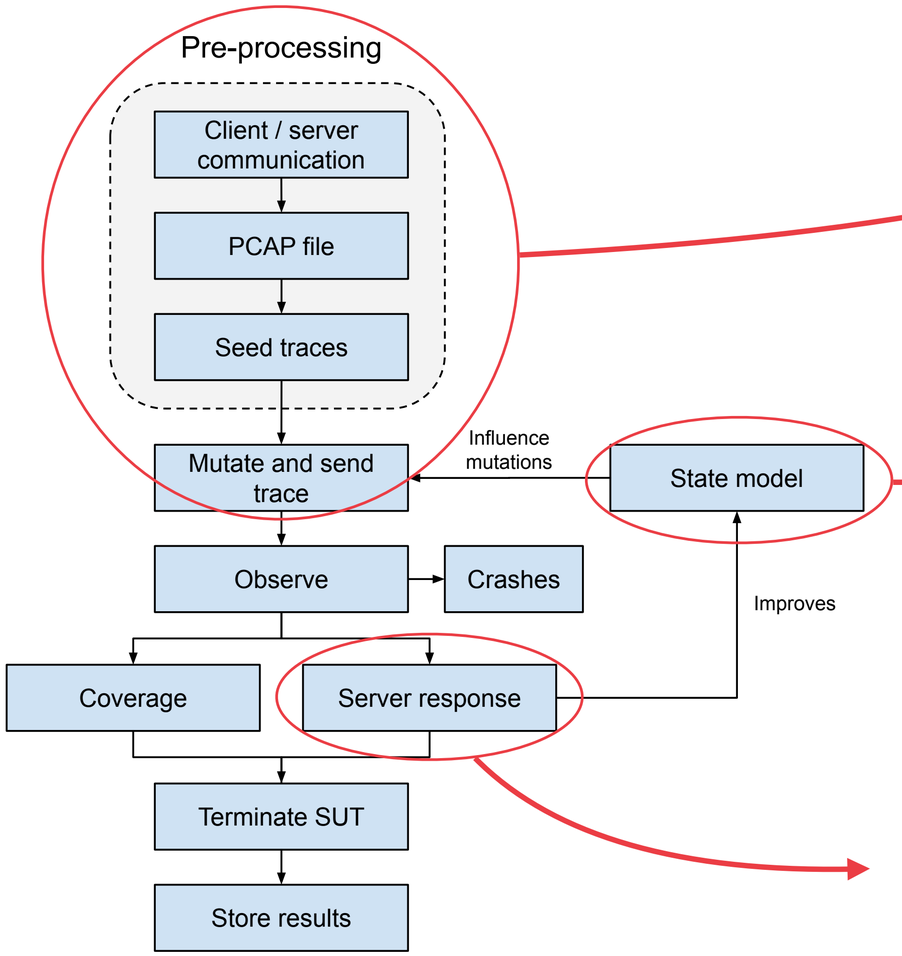
- Not able to explore the state model
- Only fuzz the first state

(2) Grammar-based fuzzers with a limited grammar in input:

- Not able to test the SUT in its whole



Fuzzing OPC UA with the stateful fuzzer AFLNet

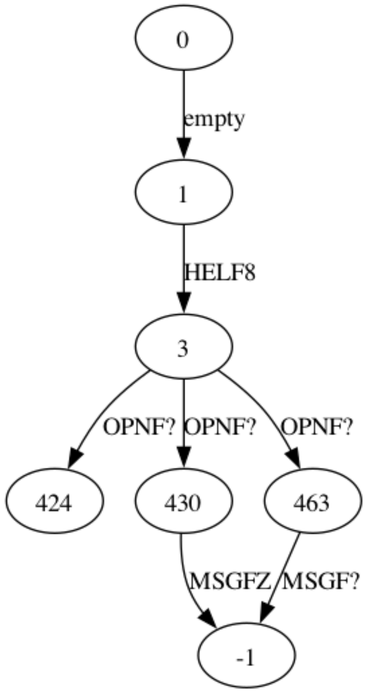


Needs a PCAP file to be able to mutate messages and traces.

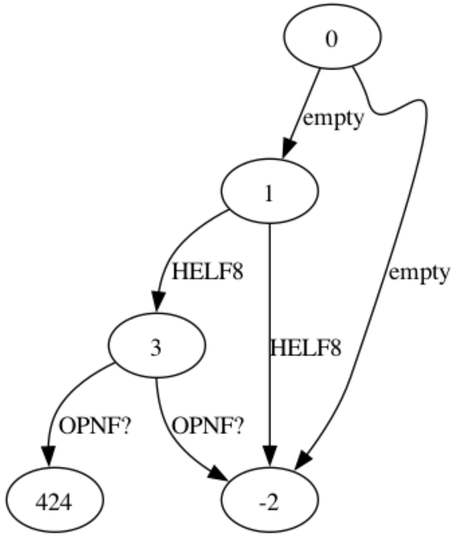
Infers the state model of the system

Uses the server responses to improve its knowledge about the state model

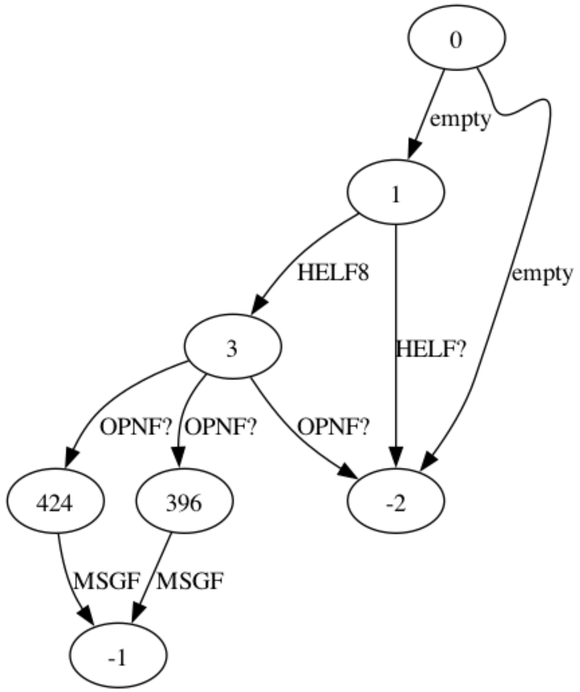
State models inferred by AFLNet



FreeOPCUA implementation



AnsiC implementation



Open62541 implementation

AFLNet Results

These bugs are new and were overlooked by the previous analysis

Implementations	AFLNwe		BooFuzz		AFLNet	
	Unique Crashes	States Covered	Unique Crashes	States Covered	Unique Crashes	States Covered
Open62541	0	1/5	0	4/5	0	5/5
ANSI-C	0	1/5	0	4/5	**1**	5/5
Free_OPCUA	**1**	1/5	0	4/5	**2**	5/5

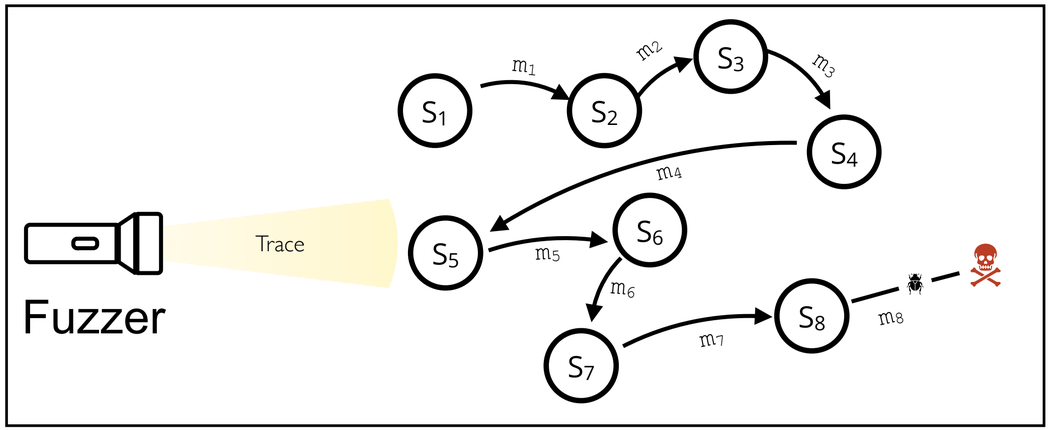
OPC UA implementation fuzzing campaign with the AFLNwe (stateless) BooFuzz (grammar-based) and AFLNet (stateful) fuzzers.

Take away slide

i. Stateful fuzzing is much more challenging than stateless fuzzing

ii. Stateless fuzzers achieve poor performance in fuzzing stateful systems

iii. Grammar-based fuzzers require a comprehensive notion of grammar in input



My LinkedIn