4th UCAAT User Conference on Advanced Automated Testing

ETSI World Class Standards

FUZZ TESTING ITS

Presented by Jürgen Großmann and Dorian Knoblauch

Fraunhofer FOKUS

4th UCAAT User Conference on Advanced Automated Testing

ETSI World Class Standards

OVERVIEW AND GENERAL CONSIDERATIONS

Why should Fuzz Testing be applied to ITS?

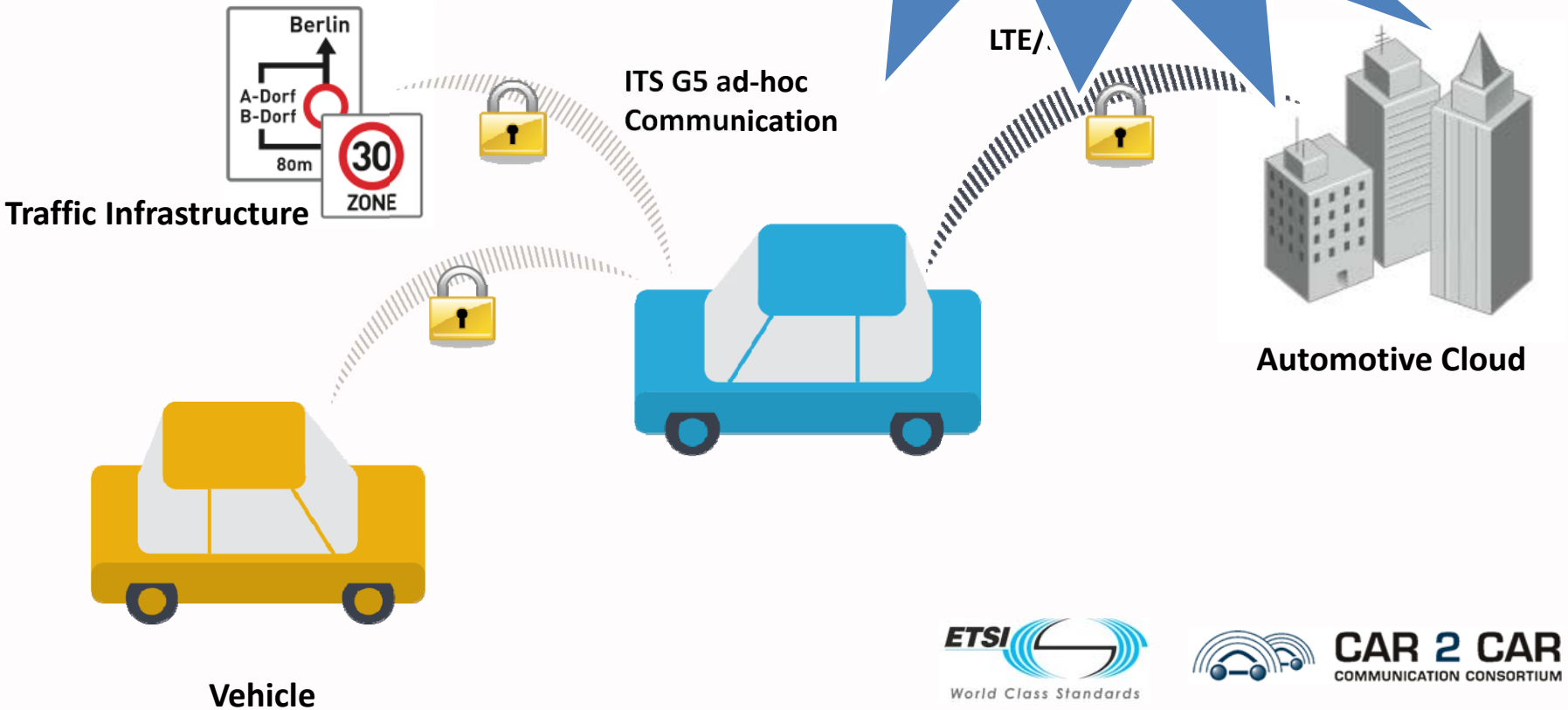
Fraunhofer FOKUS

ETSI World Class Standards

Intelligent Mobility:

… just another application in the IoT

Interoperability & Security



ETSI World Class Standards

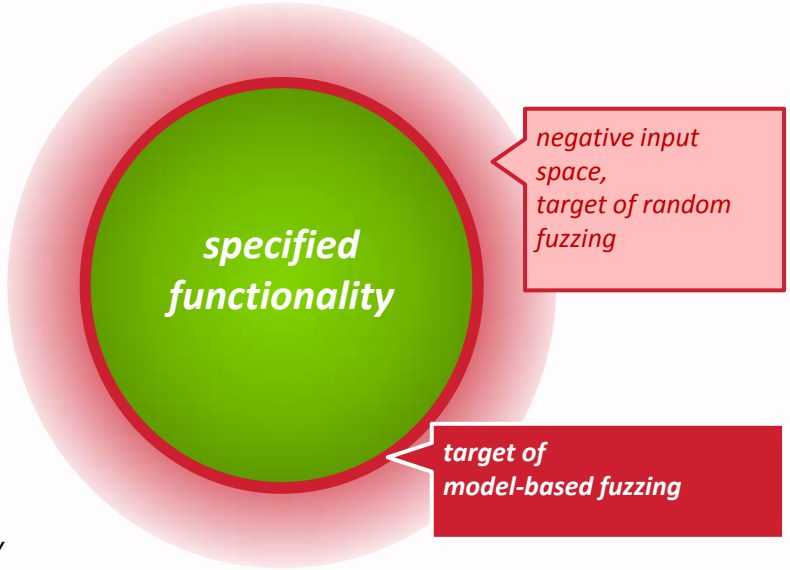
CAR 2 CAR COMMUNICATION CONSORTIUM

Model-Based Fuzzing

Challenge: Finding 0-day vulnerabilities in a highly automated, effective manner (crashes, buffer overflows, SQL injection, cross-site scripting, …)

Solution: Model-based Fuzzing

- aims at fault input validation
- stressing the SUT with semi-valid inputs



see also:
Takanen, Ari; DeMott, Jared D.; Miller, Charles: Fuzzing for Software Security Testing and Quality Assurance, 2008 ; ISBN 978-1-59693-214-2

Fuzzing the ITS Stack

Challenges and constraints

- Complex stack and overall test set up
- Binary encoded data
- Simple interactions (broadcast messages)
- No or only limited feedback from the SUT (Black box approach is required)
- Only a limited set of devices and applications are publicly available

The Tools

TTworkbench, Fuzzing Library Fuzzino, ETSI ITS Conformance Test Suite

FUZZINO: supports generation and mutation based fuzzing

- **platform independent: is implemented in Java**
- **language independent**: provides an **XML**-based interface
- **automated**: automatically selects appropriate fuzzing heuristics
- **communicative**: tells you which fuzzing heuristics are used
- **efficient & scalable**: the user can decide
  - **which fuzzing heuristics** shall be used
  - **amount of fuzz test data**: avoids generating billions of values

https://github.com/fraunhoferfokus/Fuzzino

FUZZINO

4th UCAAT User Conference on Advanced Automated Testing

ETSI World Class Standards

IMPLEMENTATION OF THE FRAMEWORK

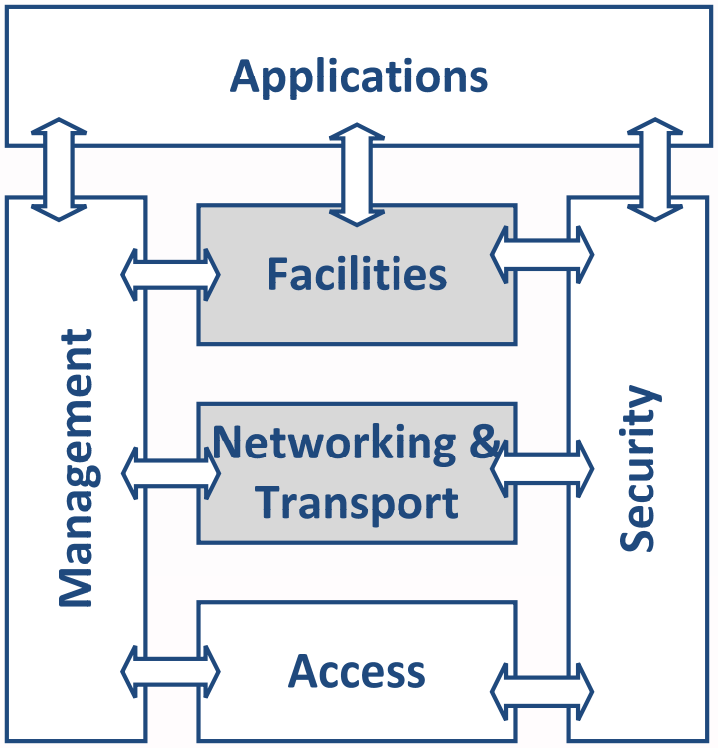
Fuzzing the ITS stack

Fraunhofer FOKUS

Fuzzing the ITS Stack

Underlying principles

- Provide test cases which will likely trigger unexpected behavior
  - Avoid simple random data
  - Supports model-based and Mutation based Fuzzing
- Enables purposive exploitation of
  - Buffer overflows
  - Number overflows
  - Unspecified data
- Targeting ITS protocols
  - GN,BTP,CAM,DENM



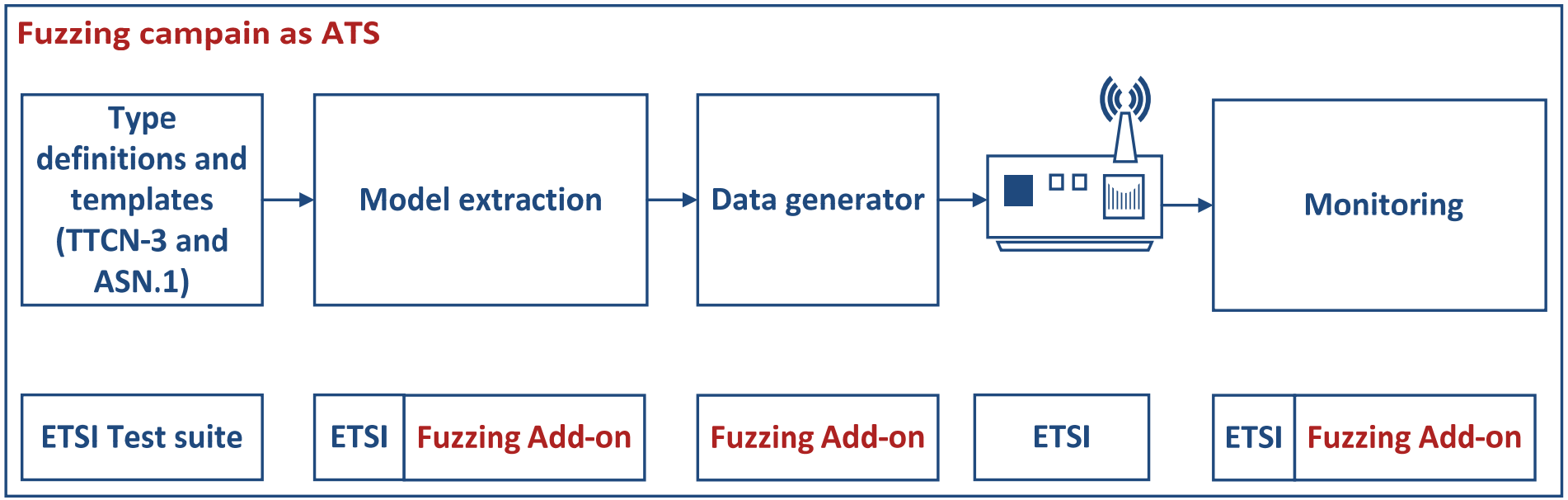
Test selection strategies

finding the optimal test suites

- Pairwise
  - Statistically most errors getting found by a pairwise combination of parameters
  - Fields in the model are treated as parameters that are allocated pairwise with potentiality malicious data provided by Fuzzino
- Fitness-proportionate selection
  - Select fields which allocation is going to cause a crash more likely
  - Uses unique log file notifications as indicator for coverage assuming that different notifications are produced by different part of the program
- Mutation strategies on Binary Level
  - Based on strategies used in American fuzzy lop

Test System Architecture

Integration with ETSI EG 202 798



Integration with TTCN-3

Example snippet

Initialization

Seed selection

Running strategies

```
testcase TC_SIMPLE_PAIR_RANDOM_ND() runs on FuzzEthernet
                                    system FuzzEthernetSystem {

        map(self:ethernetPort, system:ethernetPort);
        var portMap p_map :={p_mode:= eth,p_eth:=ethernetPort};


        var anytype seed := {GeoNetworkingPdu := pdu_gnBTP_A()}

        tcf_simple_pairwise(seed,m_curTimeAdj_list,p_map);
        tcf_simple_mutation(seed,m_curTimeAdj_list,p_map);


        unmap(self:ethernetPort, system:ethernetPort);
        setverdict(pass, "No exception occurred"); }
```

Adjusting timestamp

Wrap up

4th UCAAT User Conference on Advanced Automated Testing

ETSI World Class Standards

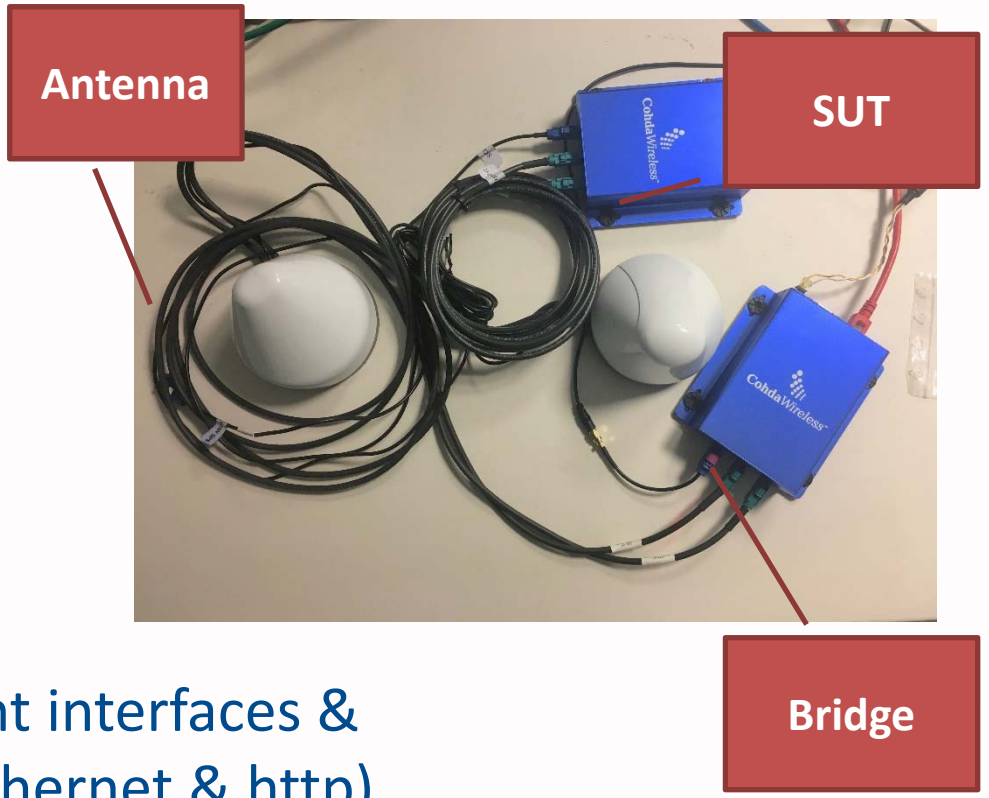
EVALUATION OF THE FRAMEWORK

Results in Fuzzing the ITS stack

Fraunhofer FOKUS

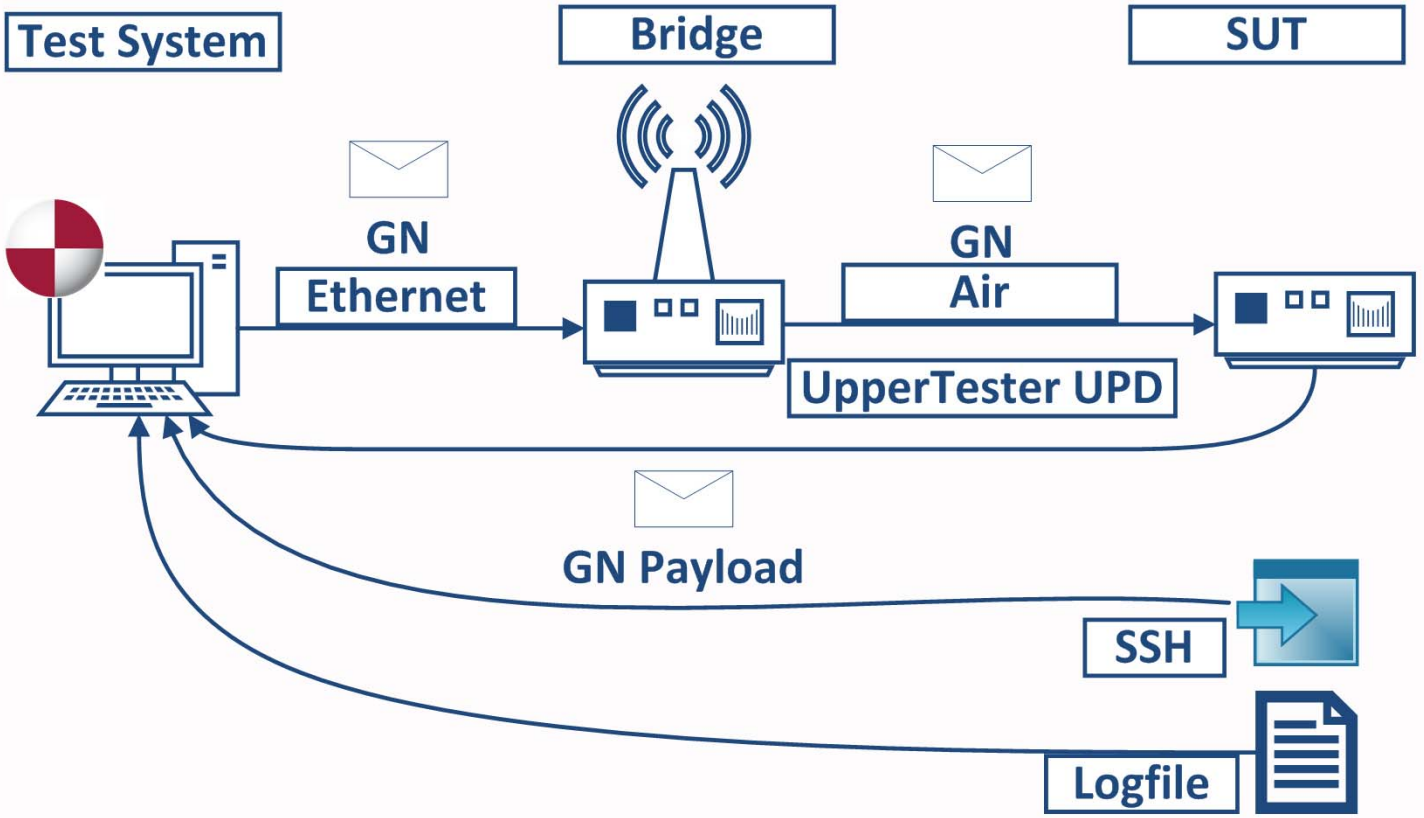
What we have tested

- Cohda stack and C2X-App (device)
- NEC stack (device)
- LDM ++
- i-GAME ITS G5 stack



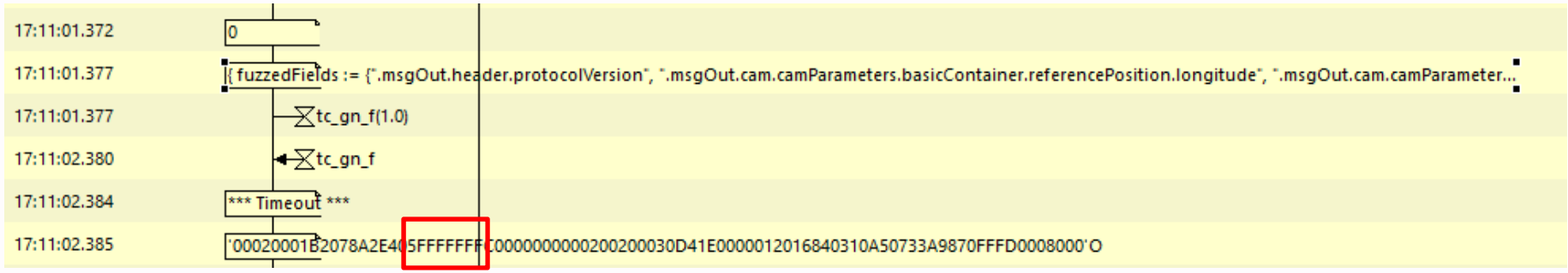
Flexible adaptations to different interfaces & protocols (e.g. air, cable and ethernet & http)

Test Environment Setup

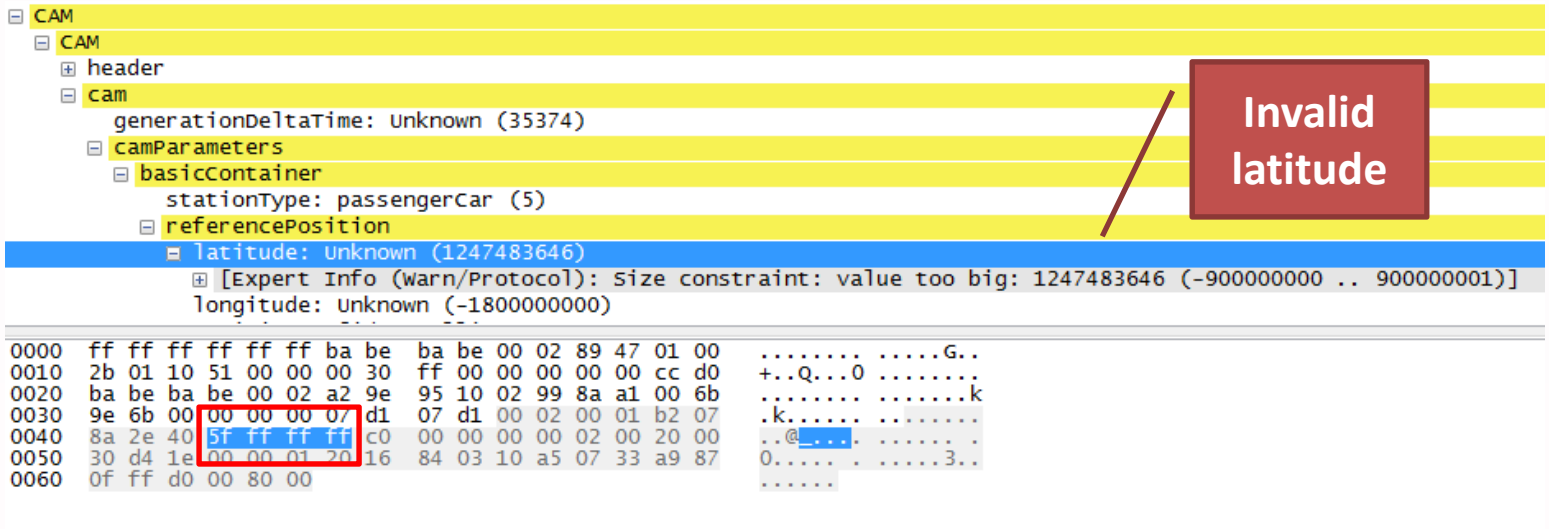


Test Execution

Execution Log



PDU at SUT



Results

- Managed to crash c2x_app on Cohda MK5
  - Segmentation fault
  - *** Error in `./c2x-app_mk5_release': double free or corruption (out): 0x75f17e80
  - Fixed by Cohda Wireless due to professional cooperation
  - Upper Tester

Results

- ITS protocol implementations are quite robust
  - due to simplicity
  - failing to check for specifications
- Managed to pass data beyond specifications, limits
  - on almost every System
  - Causes programs that rely on ITS data to crash
    - Like in the case of the UpperTester
    - Not enough programs out there for further Testing
- Amount/Rate of malicious data is important

Outlook

Fraunhofer Fokus is going to …

- deploy the current approach
  - as an add on to the ETSI conformance tests
  - currently looking for opportunities to present
- extend the current approach
  - addressing SPAT and MAP
  - addressing and integrating ITS security
  - cloud based deployment to allow for remote testing

Contact

Fraunhofer FOKUS

Jürgen Großmann & Dorian Knoblauch
juergen.grossmann@fokus.fraunhofer.de
dorian.knoblauch@fokus.fraunhofer.de

Fraunhofer FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
www.fokus.fraunhofer.de

... and thanks to Spirent for supporting our work
SPIRENT