Model-Based Fuzz Testing for GNSS Receiver

Nina Haag, Daniel Prun, Antoine Blais

Fédération ENAC ISAE-SUPAERO ONERA, Université de Toulouse, France, Collins Aerospace

Abstract

GNSS (Global Navigation Satellite System) receivers are vital for aircraft navigation system reliability and safety. However, traditional test methods recommended by norms have limitations in verifying their expected properties. This article presents a thesis aimed at enhancing the testing process by integrating Model-Based Testing (MBT) and Fuzz Testing approach. Our approach involves leveraging formal models, including behavioral models (e.g., Finite State Machines, Sequence diagrams) and static ones (e.g., OCL, BNF-based grammar descriptions), to generate relevant test cases through a dedicated mutation process. We intend to validate our approach by developing a dedicated framework for GNSS receiver verification, focusing on the critical RAIM (Receiver Autonomous Integrity Monitoring) function.

1 Context

In aviation, ensuring the reliability and safety of aircraft navigation systems, particularly GNSS (Global Navigation Satellite System) receivers, is paramount. Traditional testing methods struggle to verify these critical systems completely, facing resource and time limitations and potentially overlooking blind spots due to their fixed test scenarios.

This article presents the first results of a doctoral thesis started in January 2024 and led by Ms. Nina Haag, highlighting the challenges that underline the need for innovative testing approaches. One promising avenue is the integration of Model-Based Testing (MBT) principles with advanced fuzzing techniques. MBT uses abstract models to systematically design, generate, and execute test cases, improving testing precision and reducing manual effort. Fuzz Testing, or more simply ”Fuzzing” on the other hand, involves feeding unexpected data inputs to uncover bugs. Advanced fuzzing techniques systematically explore input space, revealing edge cases missed by traditional methods.

Combining MBT principles with advanced fuzzing techniques offers a promising approach to address challenges in verifying airborne GNSS receivers. By developing abstract models of GNSS receiver behavior and using them to generate fuzz tests, or subjecting models themselves to fuzz testing, the robustness and resilience can be systematically evaluated, controlled under various conditions.

2 Objective

Our research seeks to explore this integrated approach in enhancing the reliability and safety of airborne GNSS receivers. By leveraging the strengths of both model-based testing and fuzzing, the aim is to develop a comprehensive Model-Based Fuzz Testing (MBFT) framework that can uncover vulnerabilities, like spoofing or jamming attacks and weaknesses in GNSS receivers, ultimately contributing to the advancement of aviation safety and testing methodologies.This approach will systematically explore interface (using black-box fuzzing) and behavior spaces (using white-box fuzzing) of GNSS receivers, revealing vulnerabilities not captured by traditional testing methods. More precisely, the objective of this research is twofold:

1. Explore combining formal models with fuzz testing to systematically generate and execute test cases, utilizing advanced fuzzing methods such as black-, grey-, and white-box techniques. This approach will systematically explore interface and behavior spaces of GNSS receivers, revealing vulnerabilities not captured by traditional testing methods.
2. The research aims to implement an MBFT framework and to evaluate its effectiveness in enhancing GNSS receiver robustness under various scenarios. Integrated into existing GNSS simulation tools, this framework will demonstrate MBFT's efficacy in detecting and mitigating risks and vulnerabilities.

By achieving these objectives, the research aims to contribute to the advancement of system testing methodologies by developing the relationship between formal model and fuzz testing, and aviation safety, by providing insights into innovative approaches for verifying the reliability and resilience of critical airborne navigation systems, including the certification constraints.

3 Current State of the Art

3.1 Model Based Fuzz Testing

MBT and fuzz testing are established methodologies in software testing, particularly in security applications, aiming to detect vulnerabilities through systematic exploration of system behaviors. Recent research focus on modeling the incoming data and protocol of systems under test, opening the way to the black box fuzz testing. For instance, [1], [2] build a grammar-driven description of the input data to guide the fuzz testing process, allowing for the automatic generation of test cases adhering to protocol specifications. [3] use UML sequence diagrams models and mutates them by fuzzing operators to produce invalid data which can then be fed to the system under test. Peng et al. [4] present T-Fuzz, which integrates with existing conformance testing environments and utilizes the Testing and Test Control Notation Version 3 (TTCN-3) testing language for automated model extraction of telecommunication protocols. Recent research propose a novel approach based on machine learning. [5] with the tool MaskFuzzer proposes a Generative Adversarial Network (GAN) model whose objective is to learn protocol syntax in order to generate high-quality test cases that conform to protocol specifications and trigger vulnerabilities. Furthermore, [6] proposes an approach where the automata of the communication protocol of the system under test is automatically inferred, and used for fuzz testing generation. This framework offers automated vulnerability discovery and aids in security analysis of cryptographic protocols.

Other studies have delved into the modeling of behavioral aspects of systems under test, leading to white-box fuzz testing. [7] propose ”A Model-Based Behavioral Fuzzing Approach for Network Service,” utilizing finite state machines to direct the fuzzing process. Wang and Xiao [8] introduce the use of a finite state machine model with the possibility to specify grammar rules with types and default values. [9] uses UML diagrams augmented with Object Constraint Language (OCL) pre and post conditions to formalize the system under test and its behavior.

In summary, MBFT approaches utilize various modeling techniques, including grammars, finite state machines, and formal verification, to guide the fuzz testing process systematically. These approaches demonstrate potential in enhancing security testing practices by automating test case generation and revealing vulnerabilities, like spoofing or jamming risks, in complex systems.

3.2 MBFT for GNSS verification

Despite these advancements, the application of MBFT in the context of GNSS receiver verification within aviation systems remains relatively underexplored. While there are efforts to enhance security testing methods, such as DIAMONDS’ MBFT approach, tailored for security-critical and networked systems, [3], comprehensive studies and practical implementations of this integration specific to GNSS receiver testing are scarce in the current state of research. Therefore, there is a need for further exploration and research in this domain to address the unique challenges posed by airborne GNSS receiver verification. For this reason, the combination of the three subject areas of MBT, fuzzing and GNSS is so unique.

4 Future works and Conclusion

MBFT presents a promising approach to enhancing the robustness and reliability of GNSS receivers in aviation systems. Even if recent research have been undertaken in the domain of formal modeling for fuzz testing, we believe there are a number of areas worth to be explored. We aim particular to develop a GNSS

RAIM model using temporal automata and leverage automata with property expressions like regular and reverse automata to specify its behavior. Additionally, we'll investigate Linear Temporal Logic (LTL) and Signal Temporal Logic (STL) to formalize the receiver's properties, guiding a targeted fuzzing approach based on its requirements and environmental dynamics.

References

[1] Deng-Tao Yang, D. Yang, Yuqing Zhang, *et al.*, "BlendFuzz: A model-based framework for fuzz testing programs with grammatical inputs," *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 1070–1076, Jun. 25, 2012.

[2] F. Pan, Y. Hou, Z. Hong, L. Wu, and H. Lai, "Efficient model-based fuzz testing using higher-order attribute grammarsl," *JOURNAL OF SOFTWARE*, vol. 8, 2013.

[3] M. Schneider, J. Großmann, N. Tcholtchev, I. Schieferdecker, and A. Pietschker, "Behavioral fuzzing operators for uml sequence diagrams," in *System Analysis and Modeling: Theory and Practice*, Ø. Haugen, R. Reed, and R. Gotzhein, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 88–104.

[4] William Johansson, W. Johansson, Martin Svensson, *et al.*, "T-fuzz: Model-based fuzzing for robustness testing of telecommunication protocols," *2014 IEEE Seventh International Conference on Software Testing, Verification and Validation*, pp. 323–332, Mar. 31, 2014.

[5] Weifeng Sun, Weifeng Sun, Bowei Zhang, *et al.*, "MaskFuzzer: A MaskGAN-based industrial control protocol fuzz testing framework," *International Conferences on Smart Internet of Things*, Aug. 1, 2022.

[6] Bin Hu, Xiaojuan Zhang, Ziqing Lin, *et al.*, "A cryptographic protocol vulnerability analysis framework based on fuzz testing and model learning," *2023 3rd International Symposium on Computer Technology and Information Science (ISCTIS)*, Jul. 7, 2023.

[7] J. Bozic and F. Wotawa, "Security testing based on attack patterns," in *2014 IEEE Seventh International Conference on Software Testing, Verification and Validation Workshops*, 2014, pp. 4–11.

[8] J. Wang, T. Guo, P. Zhang, and Q. Xiao, "A model-based behavioral fuzzing approach for network service," *2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control*, pp. 1129–1134, 2013.

[9] J. Botella, J. Capuron, F. Dadeau, E. Fourneret, B. Legeard, and F. Schadle, "Complementary test selection criteria for model-based testing of security components," *International Journal on Software Tools for Technology Transfer*, vol. 21, no. 4, pp. 425–448, 2019.