Poster: Automated Discovery of Sensor Spoofing Attacks on Robotic Vehicles

Kyeongseok Yang*
Korea University
Seoul, Republic of Korea
ks8171235@korea.ac.kr

Sudharssan Mohan*
University of Texas at Dallas
Richardson, Texas, USA
sudharssan.thanammohan@utdallas.edu

Yonghwi Kwon
University of Virginia
Charlottesville, Virginia, USA
yongkwon@virginia.edu

Heejo Lee
Korea University
Seoul, Republic of Korea
heejo@korea.ac.kr

Chung Hwan Kim
University of Texas at Dallas
Richardson, Texas, USA
chungkim@utdallas.edu

ABSTRACT

Robotic vehicles are playing an increasingly important role in our daily life. Unfortunately, attackers have demonstrated various sensor spoofing attacks that interfere with robotic vehicle operations, imposing serious threats. Thus, it is crucial to discover such attacks earlier than attackers so that developers can secure the vehicles. In this paper, we propose a new sensor fuzzing framework SensorFuzz that can systematically discover potential sensor spoofing attacks on robotic vehicles. It generates malicious sensor inputs by formally modeling the existing sensor attacks and leveraging high-fidelity vehicle simulation, and then analyzes the impact of the inputs on the vehicle with a resilience-based feedback mechanism.

CCS CONCEPTS

• **Security and privacy → Software and application security**;
• **Computer systems organization → Embedded and cyber-physical systems**.

KEYWORDS

Robotic vehicle; Sensor spoofing; Fuzzing

ACM Reference Format:
Kyeongseok Yang*, Sudharssan Mohan*, Yonghwi Kwon, Heejo Lee, and Chung Hwan Kim. 2022. Poster: Automated Discovery of Sensor Spoofing Attacks on Robotic Vehicles . In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22), November 7–11, 2022, Los Angeles, CA, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3548606.3563551

1 INTRODUCTION

Robotic vehicles (RVs), such as unmanned aerial and ground vehicles, have been increasingly adopted in commercial and military applications that we rely on, such as delivery, search, and rescue. Meanwhile, researchers have demonstrated various attacks on RVs that can cause critical consequences such as physical accidents. In particular, sensor spoofing attacks[1] allow external adversaries to tamper with RV control systems without having access to the vehicles and cause catastrophic consequences [3–5].

Recent studies such as Son et al. [7] show that one can leverage particular acoustic noises to disturb a gyroscope and destabilize/crash the RV. Trippel et al. [8] and Tu et al. [9] demonstrate that attackers can tamper with inertial sensor inputs to gain precise control over RV movements. However, it remains challenging to discover yet-unknown sensor attacks for an RV and defend against them due to: (1) the *large* and *dynamic* spectrum of spoofed sensor input values that cause control disturbance; and (2) the difficulty of evaluating the RV's *resilience* to those sensor inputs with expensive and time-consuming physical experiments.

In this paper, we develop a feedback-driven fuzzer, SensorFuzz that generates spoofed sensor inputs realistically to explore the large and dynamic input space and discover potential sensor attacks for a target RV. Given the physical properties of the RV and its control system as input, SensorFuzz executes the system on a high-fidelity software-in-the-loop (SITL) simulator [6]. SensorFuzz then generates realistic sensor inputs for various sensors (e.g., a gyroscope and an accelerometer) to reduce false positives by leveraging a *sensor input mutation model*, which has multiple parameters to mutate based on the formal representations of existing sensor attacks [7–11]. During the injection of mutated sensor inputs, SensorFuzz quantitatively measures the RV's resilience to the attack by monitoring the internal states of the control system, and detecting a severe impact that the attack may cause (*e.g.*, a crash or unrecoverable divergence from the mission). This *resilience score* is then used as feedback to generate the next attack case to increase the chance of efficiently discovering new successful attacks.

We summarize our contributions as follows:

- We analyze the unique challenges in discovering sensor spoofing attacks on RVs to define a new methodology to generate realistic sensor attacks and quantify the resilience of RVs to these attacks.
- We design and implement SensorFuzz, a novel sensor fuzzing framework and demonstrate how feedback-driven fuzzing can be applied to the domain of sensor spoofing attacks.
- We validate the capability of SensorFuzz by re-discovering existing sensor attacks and discussing our future evaluation plan.

*These authors contributed equally to this work.

CCS '22, November 7–11, 2022, Los Angeles, CA, USA

ACM ISBN 978-1-4503-9450-5/22/11.
https://doi.org/10.1145/3548606.3563551

[1]We will use *sensor attacks* and *sensor inputs* to indicate sensor spoofing attacks and sensor input values, respectively.

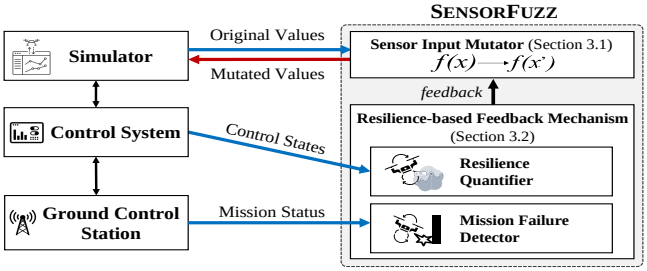


Figure 1: Architecture of SensorFuzz.

2 MOTIVATION AND CHALLENGES

We elaborate on the two major challenges in the automated discovery of sensor attacks via sensor fuzzing that motivates SensorFuzz.

Large Input Space. The fuzzing input space for sensors is extremely large and, worse, sensors generate various input values to the RV continuously. But, it is unnecessary to generate all possible values because there exist a large number of unrealistic inputs that would never occur in the real world. Hence, a well-guided fuzzing that leverages the semantics of sensors and relevant attacks would significantly facilitate the process of attack discovery.

Fuzzing Feedback. Evaluating the impact of sensor attacks on a RV is essential for sensor fuzzing but challenging. Naive metrics such as a drop in altitude, a crash, or failure to reach the destination are too coarse-grained to be useful as feedback to guide the fuzzer, especially unsuitable for making progress to find new attacks.

3 DESIGN

Figure 1 depicts the overall design of SensorFuzz. An RV in a simulated environment consists of three main components:

- A high-fidelity simulator [2, 6] that simulates environmental factors (*e.g.*, wind and magnetic field) and generates sensor inputs for the various sensors accordingly.
- A control system [1] responsible for computing the outputs (*e.g.*, motor outputs) according to given inputs (*e.g.*, sensor readings and mission commands).
- A ground control station (GCS) sending mission commands to the control system, (*e.g.*, take off and fly to coordinates).

SensorFuzz leverages the components as follows. First, in the simulator, we instrument the functions that generate the simulator's sensor inputs to retrieve the original sensor values and inject sensor values mutated by SensorFuzz. Second, the control state logs produced by the control system are used to calculate the resilience of the RV to the mutated values in real-time. The resilience is then used as feedback to aid the generation of the next inputs. Third, the mission status from the GCS is used to detect mission failures.

Our design aims to be non-intrusive, requiring minimal changes to the simulator and avoid affecting the vehicle's operations.

3.1 Sensor Input Mutator

We develop the *sensor input mutation model* by leveraging observations and constraints from the existing sensor attacks and their formulation [7–11]. This model is used to realistically mutate values for multiple sensors (*e.g.*, an accelerometer and a gyroscope) , the sensor to be spoofed in a fuzzing iteration is decided randomly. If the parameters used in an attack on a particular sensor is fully explored according to our feedback mechanism, we restart a new fuzzing iteration with another randomly chosen sensor.

The following equation defines our mutation model to produce the mutated sensor input $\hat{s}(t)$ given an original sensor input $s(t)$ where A is the amplitude, F is the frequency of the digital signal, ϕ is the phase and t is the sampling time.

$$\hat{s}(t) = s(t) + A * sin(2\pi Ft + \phi) \quad (1)$$

We mutate values of A, F, and ϕ to perform different variations of the attacks according to our feedback mechanism. Next, we briefly discuss a few of the attacks we have leveraged to develop this model.

Signal Injection Attack. We use the attacks from [7–9] that distort electrical acceleration signals in a capacitive MEMS accelerometer using signal injection. For signal injection attacks, F serves as the acoustic frequency, played at amplitude A and phase ϕ.

Output Biasing Attack. This attack was derived from Trippel et al. [8]. An output biasing attack can be launched by performing amplitude/phase modulation to manipulate the injected signal, for example, by performing phase modulation to maximize one half of the signal and minimize the other. Specifically, after mutating A, F, and ϕ. the attacker can perform output biasing attack by performing amplitude/phase modulation at the desired timing.

Side-swing & Switching Attacks. We leverage Tu et al. [9] to modify the oscillating pattern of an accelerometer signal, by changing the amplitude A, frequency F and phase ϕ. To perform a side-swing attack, we can increase A during a mission (*i.e.*, when the RV is moving towards a target direction) or decrease A otherwise. For a switching attack, we can mutate F to control the direction of the digital signal.

3.2 Resilience-based Feedback Mechanism

This component aims to evaluate the resilience of the target RV to the generated sensor inputs and use the score as feedback to guide the input mutator toward new successful attack cases.

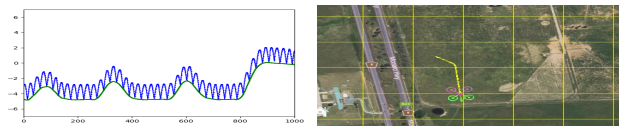
Resilience Quantifier. Our mechanism computes the resilience score ρ using the history of the control errors that the control system tries to minimize to maintain the position, velocity, and acceleration in each of the six degrees of freedom (x, y, z axes and rotation around them). The following equations show how ρ is computed using Integral Absolute Error (IAE).

$$\rho = 1/max(\{IAE(t_1), \dots, IAE(t_n)\}) \quad (2)$$

$$IAE(t) = \int_t^{t+w} |r(s) - x(s)|ds \quad (3)$$

Let $r(s)$ and $x(s)$ the reference and current states, respectively. For each degree of freedom, we collect the errors over time t with a sliding window w, which can be adjusted via configuration. During a mission, we compute IAE multiple times repeatedly for n number of temporal sections that are adjacent, each spanning w duration. The multiplicative inverse of the highest IAE across all sections is taken as the resilience score at the end of the mission.

Mission Failure Detector. Our failure detector monitors the mission status from the GCS and the control states from the control

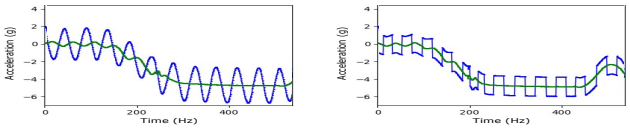


(a) Sensor input graph (b) Simulator view

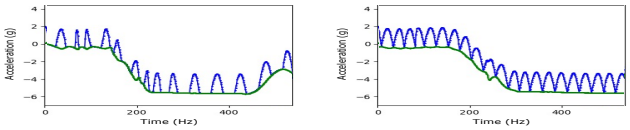
```
start
fuzzing Mode : [5], A[i] : [2], epsilone : [0.4], phi0 : [1]
IAE : [0.0], ISE : [0.0], ITAE : [0.0]
IAE : [0.42236067005433314], ISE : [0.042575252184343995], ITAE : [45.023322468041445]
IAE : [0.6972081439259151], ISE : [0.08024062692278255], ITAE : [75.96878246326621]
IAE : [1.132901101373136], ISE : [0.24279228872808829], ITAE : [128.15745393367251]
IAE : [2.045568266417831], ISE : [0.9308681965127275], ITAE : [240.85224527819082]
IAE : [3.235622972715646], ISE : [2.2145354211178367], ITAE : [393.43008760188241]
IAE : [4.400189160036722], ISE : [3.7599834528853235], ITAE : [549.8107817393195]
IAE : [5.465887804748905], ISE : [5.385134114843183], ITAE : [700.2436222256802]
IAE : [6.328291280143377], ISE : [6.732925560512476], ITAE : [828.8154685939662]
IAE : [6.936207367572933], ISE : [7.600599499841246 5], ITAE : [926.3649530158145]
```

(c) Feedback calculation

Figure 2: Screenshot of a SensorFuzz demonstration.



(a) Signal injection attack (b) Output biasing attack



(c) Side-swing attack (d) Switching attack

Figure 3: Existing sensor attacks [7–11] detected by SensorFuzz.

system (via the resilience quantifier) to determine the end of a mission. Specifically, our mission failure detector repeatedly checks if one of the following events occurs using the mission status information: (1) a vehicle crash, (2) a large delay between two waypoints, and (3) a large and increasing divergence from the mission.

Feedback-driven Fuzzing. When a mission ends successfully, our input mutator refers to the resilience scores of the latest two missions to decide how to mutate the next set of sensor inputs. If the last resilience score decreased compared to the previous one, it maintains the current mutation function for the parameter, as it indicates this function guided the fuzzer toward a greater attack impact. Otherwise, it chooses another mutation function from the list.

4 EVALUATION

Preliminary Results. We present the results and the work done so far. We have implemented the entire fuzzing loop including input generation/mutation and successfully tested it over a set of missions. We have used ArduPilot [1], instrumented the simulator [6] to inject sensor inputs, and modified its logging functions to share data to SensorFuzz is depicted in §3.

Figure 2 shows SensorFuzz in action, performing automated fuzzing to discover sensor spoofing attacks. The graph shows the original accelerometer values in green and mutated values being fed to the sensor in blue. The terminal shows the resilience score being computed. The general information such as drone position and orientation are in the simulator view.

Figure 3 shows the graphs of all the attacks from §3 that have been detected by SensorFuzz successfully.

Evaluation Plan. We plan to evaluate our proof of concept system by the following three methodologies:

- **Input Generation:** We will test the values from the input mutator on a real RV to empirically prove that new attacks SensorFuzz discovers are feasible in practice.
- **Resilience-based Feedback:** We will test the effectiveness of resilience as fuzzing feedback by comparing it to a baseline fuzzer, which generates sensor inputs randomly without any feedback.
- **Attack Discovery and Analysis:** We plan to discover new sensor attacks using SensorFuzz, and analyze their impact and possible defense to demonstrate the effectiveness.

5 CONCLUSION

We present a novel sensor fuzzing framework designed to discover new sensor spoofing attacks on RVs. We present the design and implementation of SensorFuzz that can efficiently explore the large input space for sensors with the formal modeling of sensor attacks and quantification of attack resilience for feedback-driven fuzzing. Our preliminary results with a popular RV control system show that SensorFuzz can detect existing sensor spoofing attacks, demonstrating its promising potential.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their insightful feedback. This work was supported in part by the University of Texas at Dallas Office of Research through the NFRS program, NSF under award number 190821, Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01697 Development of Automated Vulnerability Discovery Technologies for Blockchain Platform Security, No.2022-0-00277 Development of SBOM Technologies for Securing Software Supply Chains, No.2022-0-01198 Convergence Security Core Talent Training Business, and No.IITP-2022-2020-0-01819 ICT Creative Consilience programs), and a gift from Cisco Systems.

REFERENCES

[1] Ardupilot. 2022. https://ardupilot.org/.
[2] Gazebo. 2022. https://gazebosim.org.
[3] The Guardian. 2009. US drones hacked by Iraqi insurgents. https://www.theguardian.com/world/2009/dec/17/skygrabber-american-drones-hacked.
[4] BBC News. 2014. Military and civilian drones have a crucial weakness that means they can be hacked. https://www.bbc.com/future/article/20140206-can-drones-be-hacked.
[5] Fox News. 2015. Drones vulnerable to terrorist hijacking, researchers says. https://www.foxnews.com/tech/exclusive-drones-vulnerable-to-terrorist-hijacking-researchers-say.
[6] SITL Simulator. 2022. https://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html.
[7] Yunmok Son, Hocheol Shin, Dongkwan Kim, Youngseok Park, Juhwan Noh, Kibum Choi, Jungwoo Choi, and Yongdae Kim. 2015. Rocking drones with intentional sound noise on gyroscopic sensors. In *USENIX Security'15*.
[8] Timothy Trippel, Ofir Weisse, Wenyuan Xu, Peter Honeyman, and Kevin Fu. 2017. WALNUT: Waging doubt on the integrity of MEMS accelerometers with acoustic injection attacks. In *EuroS&P'17*. IEEE.
[9] Yazhou Tu, Zhiqiang Lin, Insup Lee, and Xiali Hei. 2018. Injected and delivered: Fabricating implicit control over actuation systems by spoofing inertial sensors. In *USENIX Security'18*.
[10] Zhengbo Wang, Kang Wang, Bo Yang, Shangyuan Li, and Aimin Pan. 2017. Sonic gun to smart devices: Your devices lose control under ultrasound/sound. In *Black Hat USA*.
[11] Chen Yan, Hocheol Shin, Connor Bolton, Wenyuan Xu, Yongdae Kim, and Kevin Fu. 2020. Sok: A minimalist approach to formalizing analog sensor security. In *IEEE S&P'20*. IEEE.