



Computational Intelligence in Security for Information Systems  
17th International conference 9-11 October 2024

# Fuzzing Robotic Software using HPC and LLM

Francisco Borja Garnelo Del Río, Francisco J. Rodríguez Lera

Camino Fernández Llamas, Vicente Matellán Olivera

Universidad de León - Campus de Vegazana s/n, 24071 León (Spain)

[infbgd01@estudiantes.unileon.es](mailto:infbgd01@estudiantes.unileon.es), {fjrodl, cferll, vmato}@unileon.es

---



**DMARCE**

Decision Making in Autonomous Robots: Cybersecurity and Explainability



DMARCE (EDMAR+CASCAR) Project PID2021-126592OB-C21 + PID2021-126592OB-C22 funded  
by MCIN/AEI/10.13039/501100011033 and by ERDF A way of making Europe

# Table of Contents

## Introduction

Challenges & innovation  
Research Objectives  
Goals and Hypotheses

## Materials and Methods

Intro  
Infrastructures  
Software

## Proof-of-Concept Design

Intro  
Generation  
Simulation  
Monitor  
Validation

## Contribution

## Conclusions & Future Work



### 1 Introduction

- Challenges & innovation
- Research Objectives
- Goals & Hypotheses

### 2 Materials and Methods

- Infrastructures
- Software

### 3 Proof-of-Concept Design

- Simulation
- Integration
- Validation

### 4 Contribution

### 5 Conclusions and Future Work



## Introduction

- Challenges & innovation
- Research Objectives
- Goals and Hypotheses

## Materials and Methods

- Intro
- Infrastructures
- Software

## Proof-of-Concept Design

- Intro
- Generation
- Simulation
- Monitor
- Validation

## Contribution

## Conclusions & Future Work

# Section 1

# Introduction

# Introduction - Challenges & innovation

## Introduction

Challenges & innovation

Research Objectives

Goals and Hypotheses

## Materials and Methods

Intro

Infrastructures

Software

## Proof-of-Concept Design

Intro

Generation

Simulation

Monitor

Validation

## Contribution

## Conclusions & Future Work



### Challenges in Robotic Software Testing:

- Traditional software testing methods are insufficient for complex robotic systems.
- Need for extensive computational resources for fuzz testing.

### Innovative Approach:

- Development of a modular, scalable testing framework (HOUSE).
- Utilizing HPC for scalable and robust testing environments.
- AI-driven input generation for fuzz testing based on natural language commands.

# Introduction - Research Objectives

## Introduction

Challenges & innovation  
Research Objectives  
Goals and Hypotheses

## Materials and Methods

Intro  
Infrastructures  
Software

## Proof-of-Concept Design

Intro  
Generation  
Simulation  
Monitor  
Validation

## Contribution

## Conclusions & Future Work



## Main research Objectives:

- **Strengthen Security** Apply fuzzing to enhance the security of robotic software using open-source tools within a scalable framework.
- **Assess AI Benefits** Investigate the benefits of integrating generative AI in the fuzzing process, focusing on message generation and mutation.
- **Leverage HPC Capabilities** Utilize HPC to support and scale the fuzzing process, evaluating its effectiveness based on prior studies.

# Introduction - Goals and Hypotheses

## Introduction

Challenges & innovation  
Research Objectives  
Goals and Hypotheses

## Materials and Methods

Intro  
Infrastructures  
Software

## Proof-of-Concept Design

Intro  
Generation  
Simulation  
Monitor  
Validation

## Contribution

## Conclusions & Future Work



## Research question

How does AI contribute to the improvement of fuzzing processes in ROS2 robotic systems?

This question arises from the following hypothesis and questions:

- **H1:** AI can significantly enhance the fuzzing process for ROS2 robotic systems.
- **Q1:** How does natural language simplify the definition of complex input scenarios in fuzzing?
- **Q2:** How does non-specialized AI hardware impact performance?



## Introduction

- Challenges & innovation
- Research Objectives
- Goals and Hypotheses

## Materials and Methods

- Intro
- Infrastructures
- Software

## Proof-of-Concept Design

- Intro
- Generation
- Simulation
- Monitor
- Validation

## Contribution

## Conclusions & Future Work

# Section 2

# Materials and Methods

# Materials and Method - Framework & key tools

## Introduction

Challenges & innovation  
Research Objectives  
Goals and Hypotheses

## Materials and Methods

Intro  
Infrastructures  
Software

## Proof-of-Concept Design

Intro  
Generation  
Simulation  
Monitor  
Validation

## Contribution

## Conclusions & Future Work



### House framework Components:

- **HPC Support:** Enables large-scale testing.
- **Fuzzing Techniques:** Automated input generation and mutation.
- **AI Integration:** Natural language processing for intelligent input generation.

### Key Tools:

- **RoboFuzz:** Core robotic fuzzing tool.
- **Marcoroni LLM:** AI model for generating test inputs.
- **Singularity:** Container technology for deployment.



# Materials and Methods - Concepts

## Introduction

- Challenges & innovation
- Research Objectives
- Goals and Hypotheses

## Materials and Methods

- Intro
- Infrastructures
- Software

## Proof-of-Concept Design

- Intro
- Generation
- Simulation
- Monitor
- Validation

## Contribution

## Conclusions & Future Work



**Large Language Model (LLM)** is an AI designed to understand and generate human-like text using deep learning on vast data.

**Fuzzing** automates the generation and testing of malformed inputs to find unexpected software behavior.

**ROS2 framework** Provides a flexible architecture for building and deploying complex robotic systems.

**High Performance Computing (HPC)** uses clusters of powerful processors to solve complex problems quickly.

# Materials and Methods - Infrastructures

## Introduction

Challenges & innovation  
Research Objectives  
Goals and Hypotheses

## Materials and Methods

Intro  
Infrastructures  
Software

## Proof-of-Concept Design

Intro  
Generation  
Simulation  
Monitor  
Validation

## Contribution

## Conclusions & Future Work



We used two different infrastructures with **non-specialized hardware for AI**:

- **Standalone SDO**, virtual machine with 8GB ram and 6 x Intel Xeon E3-12xx v2 vcpu (virtual cpu) and linux kernel *5.3.11-100.x86\_64* (*x86\_64*). Local storage on mechanical hard disks.
- **High-Performance Computing HPC**, computing cluster with Haswell nodes in bare-metal with 48GB of ram and 2 x Intel Xeon E5-2630 v3 @ 3.20GHz with a total of 16 cores and a Linux kernel *3.10.0-1062.9.1.x86\_64* (*x86\_64*) . Network storage and cache on solid disks.

*NOTE: When using containers the distro is indifferent.*

# Materials and Methods - Software

## Introduction

Challenges & innovation  
Research Objectives  
Goals and Hypotheses

## Materials and Methods

Intro  
Infrastructures  
Software

## Proof-of-Concept Design

Intro  
Generation  
Simulation  
Monitor  
Validation

## Contribution

## Conclusions & Future Work



The experiments made use of the following software:

- **RoboFuzz**, an autonomous fuzz testing tool for robotic systems.
- **SLURM** (Simple Linux Utility for Resource Management) a widely used open-source workload manager and job scheduler for Linux and Unix-based clusters and supercomputers.
- **Singularity** a container solution created to run complex applications on HPC clusters in a simple, portable, and reproducible way.
- **Docker**, an open container platform for developing, shipping, and running applications.
- **Marcoroni 7B V3-GGUF**, an auto-regressive language model designed for the English language, integrated into ROS2 using the `llama_ros` package .



## Introduction

- Challenges & innovation
- Research Objectives
- Goals and Hypotheses

## Materials and Methods

- Intro
- Infrastructures
- Software

## Proof-of-Concept Design

- Intro
- Generation
- Simulation
- Monitor
- Validation

## Contribution

## Conclusions & Future Work

# Section 3

# Proof-of-Concept Design

# Proof-of-Concept Design - Steps of fuzz testing

## Introduction

Challenges & innovation  
Research Objectives  
Goals and Hypotheses

## Materials and Methods

Intro  
Infrastructures  
Software

## Proof-of-Concept Design

Intro  
Generation  
Simulation  
Monitor  
Validation

## Contribution

## Conclusions & Future Work

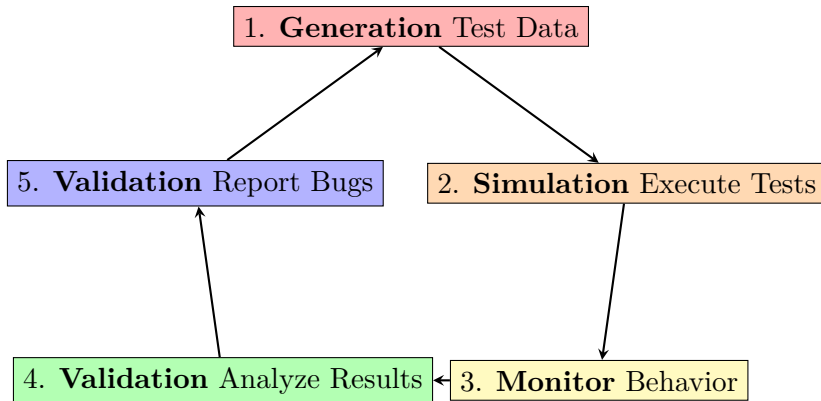


Figure: Fuzzing Testing Cycle Diagram

# Proof-of-Concept Design - Generation

## Introduction

Challenges & innovation  
Research Objectives  
Goals and Hypotheses

## Materials and Methods

Intro  
Infrastructures  
Software

## Proof-of-Concept Design

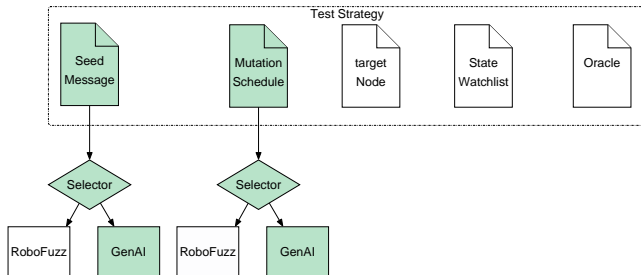
Intro  
Generation  
Simulation  
Monitor  
Validation

## Contribution

## Conclusions & Future Work



**Integration of GenAI:** Expanded RoboFuzz's input management capabilities to utilize LLM or VLM models compatible with llama.cpp .



**Figure:** The input test strategies of RobotFuzz, inclusive of the genai module features.

# Proof-of-Concept Design - Simulation

## Introduction

- Challenges & innovation
- Research Objectives
- Goals and Hypotheses

## Materials and Methods

- Intro
- Infrastructures
- Software

## Proof-of-Concept Design

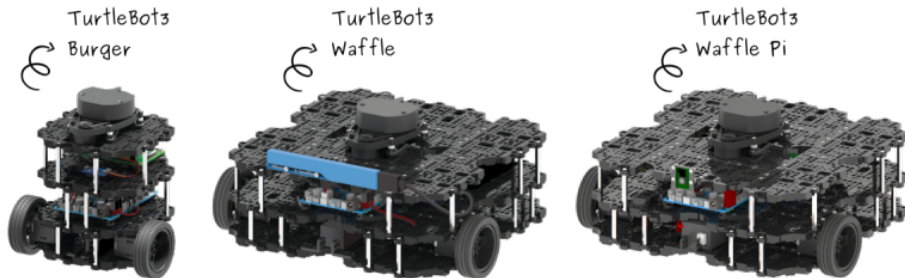
- Intro
- Generation
- Simulation
- Monitor
- Validation

## Contribution

## Conclusions & Future Work



**Simulation with Turtlebot3(tb3) robot:** Conducted fuzzing on a differential wheeled mobile robot equipped with a LiDAR sensor designated topics.



# Proof-of-Concept Design - Monitor

## Introduction

Challenges & innovation  
Research Objectives  
Goals and Hypotheses

## Materials and Methods

Intro  
Infrastructures  
Software

## Proof-of-Concept Design

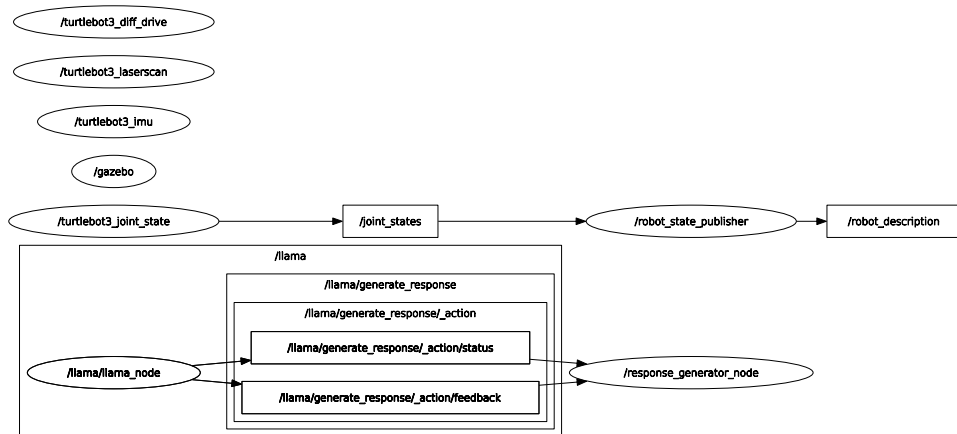
Intro  
Generation  
Simulation  
Monitor  
Validation

## Contribution

## Conclusions & Future Work



`rqt_graph` associated all active nodes and topics during the validation process.





# Proof-of-Concept Design - Monitor

## Introduction

- Challenges & innovation
- Research Objectives
- Goals and Hypotheses

## Materials and Methods

- Intro
- Infrastructures
- Software

## Proof-of-Concept Design

- Intro
- Generation
- Simulation
- Monitor
- Validation

## Contribution

## Conclusions & Future Work

```
root@2f4347e537e8:/# ros2 topic echo /robot_description

INIT: 0x7f254d6b6348 0x7f254d6b925c
data: ' 5y@gB'
---
data: ' 5y#fB'
---
data: ' 5y@fN'
---
```

**Figure:** It shows the monitoring of messages received by the `/robot_description` topic during the validation process.



# Proof-of-Concept Design - Validation

## Introduction

- Challenges & innovation
- Research Objectives
- Goals and Hypotheses

## Materials and Methods

- Intro
- Infrastructures
- Software

## Proof-of-Concept Design

- Intro
- Generation
- Simulation
- Monitor
- Validation

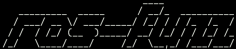
## Contribution

## Conclusions & Future Work



**Validation Process:** Detailed logs and statistics of the fuzzing process, including seed generation and mutation.

```
root@2f4347e537e8:/robofuzz/src# python3 -u ./fuzzer.py --tb3-strd --ros-pkg turtlebot3 --no-cov --watchlist watchlist/tb3_mod.json --genai
INIT: 0x7fe9ba6ec348 0x7fe9ba6ef25c



[+] Initializing shm for coverage tracking
[+] Initializing test case queue
[+] Starting TurtleBot3 STIL + TRD stack & Gazebo simulator
[tb3] started tb3 sitl stack 281514

----- TARGET INFO -----
- TOPIC: /robot_description
- message type: <class 'std_msgs.msg._string.String'>

----- BEGIN FUZZING -----
CYCLE: 0 ROUND: 0 EXEC: 0 1714129301.5873528
NEW CYCLE
Current field: data (<U0>)
STAGE: GEN
[INFO] [1714129301.951389825] [response_generator_node]: [GENAI] Generation BEGIN prompt=[length=6 description=Fill with random characters]
max_token=2M max_chars=6
[INFO] [1714129372.554119552] [response_generator_node]: [GENAI] Generation END
[INFO] [1714129372.556795978] [response_generator_node]: [GENAI] Time to eval: 16.795803785324097 s
[INFO] [1714129372.558986123] [response_generator_node]: [GENAI] Prediction speed: 0.46497239046841543 t/s
[INFO] [1714129372.561027554] [response_generator_node]: [GENAI] Generated AI result=| 5y@fB|
[executor] len(msg_list): 1
repeating 0-th sequence
[+] Starting TurtleBot3 STIL + TRD stack & Gazebo simulator
[tb3] started tb3 sitl stack 281617
[executor] started ros2 bag recording
[executor] start watching
[executor] stop watching
[executor] killing ros2 bag
[+] no error found
```

**Figure:** The operation of the GenAI module in seed generation.

# Proof-of-Concept Design - Validation

## Introduction

Challenges & innovation  
Research Objectives  
Goals and Hypotheses

## Materials and Methods

Intro  
Infrastructures  
Software

## Proof-of-Concept Design

Intro  
Generation  
Simulation  
Monitor  
Validation

## Contribution

## Conclusions & Future Work



```
root@2f4347e537e8:/robofuzz/src# python3 -u ./fuzzer.py --tb3-strd --ros-pkg turtlebot3 --no-cov --watchlist watchlist/tb3_mod.json --genai
INIT: 0x7fe9ba6ec348 0x7fe9ba6ef25c

  /---\
 /---\ /---\ /---\ /---\ /---\
/---\ /---\ /---\ /---\ /---\
\---/ \---/ \---/ \---/ \---/
 \---/ \---/ \---/ \---/ \---/

[*] Initializing shm for coverage tracking
[*] Initializing test case queue
[*] Starting TurtleBot3 STIL + TRD stack & Gazebo simulator
[tb3] started tb3 sitl stack 281514

----- TARGET INFO -----
- TOPIC: /robot_description
- message type: <class 'std_msgs.msg._string.String'>

----- BEGIN FUZZING -----
CYCLE: 0 ROUND: 0 EXEC: 0 1714129301.5873528
NEW CYCLE
Current field: data (<U0>)
STAGE: GEN
[INFO] [1714129301.951389825] [response_generator_node]: [GENAI] Generation BEGIN prompt=[length=6 description=Fill with random characters|
max_token=24 max_chars=6
[INFO] [1714129372.554119552] [response_generator_node]: [GENAI] Generation END
[INFO] [1714129372.556795978] [response_generator_node]: [GENAI] Time to eval: 16.795803785324097 s
[INFO] [1714129372.558986123] [response_generator_node]: [GENAI] Prediction speed: 0.46497239046841543 t/s
[INFO] [1714129372.561027554] [response_generator_node]: [GENAI] Generated AI result=| 5y@fB|
[executor] len(msg_list): 1
repeating 0-th sequence
[*] Starting TurtleBot3 STIL + TRD stack & Gazebo simulator
[tb3] started tb3 sitl stack 281617
[executor] started ros2 bag recording
[executor] start watching
[executor] stop watching
[executor] killing ros2 bag
[+] no error found
```

# Contribution

## Introduction

Challenges & innovation  
Research Objectives  
Goals and Hypotheses

## Materials and Methods

Intro  
Infrastructures  
Software

## Proof-of-Concept Design

Intro  
Generation  
Simulation  
Monitor  
Validation

## Contribution

## Conclusions & Future Work



- **Integration of LLM:** Developed a new module, GenAI, for RoboFuzz to utilize Large Language Models (LLM) in the seed generation and mutation process.
- **AI-Driven Fuzzing:** Replaced the traditional Mutator module with a generative AI model, enhancing the efficiency and effectiveness of fuzz testing.
- **Modular Design:** Designed for modular utilization with Singularity container technology, facilitating large-scale deployment in High-Performance Computing (HPC) environments.
- **Open Source Tool:** Provided a ready-to-use deployable tool with all associated materials, including the proof of concept and code, accessible on GitHub for review and utilization.

# Conclusions

## Introduction

Challenges & innovation  
Research Objectives  
Goals and Hypotheses

## Materials and Methods

Intro  
Infrastructures  
Software

## Proof-of-Concept Design

Intro  
Generation  
Simulation  
Monitor  
Validation

## Contribution

## Conclusions & Future Work



### ■ Generative AI:

- Represents a new paradigm with significant advantages, such as the ability to "program" generation and modification using natural language.
- Not always optimal for use cases that cannot be easily explained in natural language.
- Performance heavily depends on hardware characteristics, impacting efficiency on older or non-optimized systems.

### ■ HPC Integration:

- Enhances the effectiveness and efficiency of fuzz testing techniques for ROS2 robotic systems.
- Contributes to the reliability and security of ROS2 systems by enabling parallelization and acceleration of the fuzz testing process.

# Future Work

## Introduction

Challenges & innovation  
Research Objectives  
Goals and Hypotheses

## Materials and Methods

Intro  
Infrastructures  
Software

## Proof-of-Concept Design

Intro  
Generation  
Simulation  
Monitor  
Validation

## Contribution

## Conclusions & Future Work



- **Specific AI Models:** Explore the use of more specific AI models tailored for different tasks to enhance the quality of fuzz testing. This includes developing models that can handle specific types of data or scenarios encountered in ROS2 systems.
- **AI-Optimized Hardware:** Investigate the performance of AI-optimized hardware to improve the efficiency of generative AI models. This involves testing on newer hardware with better support for AI workloads to reduce latency and increase throughput.
- **Framework Expansion:** Expand the application of the developed framework to other domains and robotic systems. This could involve adapting the framework for use in different types of robots or in other areas of software testing.
- **Integration with Other Tools:** Integrate the fuzzing framework with other security and testing tools to create a more comprehensive testing suite.

## Introduction

- Challenges & innovation
- Research Objectives
- Goals and Hypotheses

## Materials and Methods

- Intro
- Infrastructures
- Software

## Proof-of-Concept Design

- Intro
- Generation
- Simulation
- Monitor
- Validation

## Contribution

## Conclusions & Future Work

Thank you for your attention.  
Do you have any questions?



**DMARCE (EDMAR+CASCAR) Project PID2021-126592OB-C21 + PID2021-126592OB-C22 funded by MCIN/AEI/10.13039/501100011033 and by ERDF A way of making Europe**