GOVERNMENT ENGINEERING COLLEGE
THRISSUR

BINARY FUZZER USING KAITAI STRUCT

SUBMITTED BY

DARSHANA DAS K

Student Id. TCR21CS020

KRISHNAHARI P

Student Id. TCR21CS032

LIBNA KURIAKOSE T

Student Id. TCR21CS033

PARVATHY C M

Student Id. TCR21CS049

PROJECT GUIDE

DR. EZUDHEEN P

27 FEBRUARY 2024

Declaration of Authorship

We declare on our honour that the work presented in this dissertation, entitled "Binary fuzzer using Kaitai Struct," is original and was carried out by Darshana Das K (TCR21CS020), Krishnahari P (TCR21CS032), Libna Kuriakose T (TCR21CS033), and Parvathy C M (TCR21CS049) under the supervision of Professor Dr. Ezudheen P (ezudheen@gmail.com).

27 February 2024

Darshana Das K

Krishnahari P

Libna Kuriakose T

Parvathy C M

Abstract

As software applications and binary file formats become more complex, effective testing methods are crucial for ensuring software reliability and security. Existing fuzzers are mostly format-agnostic which makes them ineffective when a specific format is required. This project introduces a new binary fuzzer crafted with Kaitai Struct, a language for defining binary data formats. The fuzzer stands out by employing Kaitai Struct's specifications to parse and generate structured inputs for various binary file formats.

Keywords: Kaitai Struct, Structure-aware fuzzing, Binary files, Grammars

Contents

List of Figures

1

Introduction

Under the guidance of Dr. Rahul Gopinath from The University of Sydney, this project, the Binary Fuzzer, took shape through collaborative efforts. Our connection with Dr. Rahul Gopinath began with an email expressing our interest in contributing to his research, particularly in fuzzing, where he is an esteemed author of the fuzzingbook. Subsequent to our correspondence, a meeting was scheduled, marking the commencement of the Binary Fuzzer project.

The significance of developing a fuzzer for binary formats is crucial in today's complex software landscape. As software applications become more intricate, ensuring their reliability and security is paramount. The Binary Fuzzer project addresses this need by utilizing the capabilities of Kaitai Struct. Kaitai Struct provides an efficient and versatile platform for parsing binary formats, facilitating the creation of a robust fuzzer. This introduction sets the stage for a comprehensive exploration of the Binary Fuzzer project, covering its development, methodologies, and contributions to the fields of fuzzing and binary security.

2

Problem

2.1 Background

Testing is vital in software development to ensure correctness, meet requirements, and ensure security. Fuzzing, a testing technique, involves providing unexpected inputs to a program to uncover vulnerabilities. Kaitai Struct, a declarative language, is used to describe binary data structures, facilitating the creation of parsers for various file formats or network stream packet formats. The described format in a Kaitai Struct file can be compiled into source files in supported programming languages, generating parsers that provide access to the data structure in a user-friendly API.

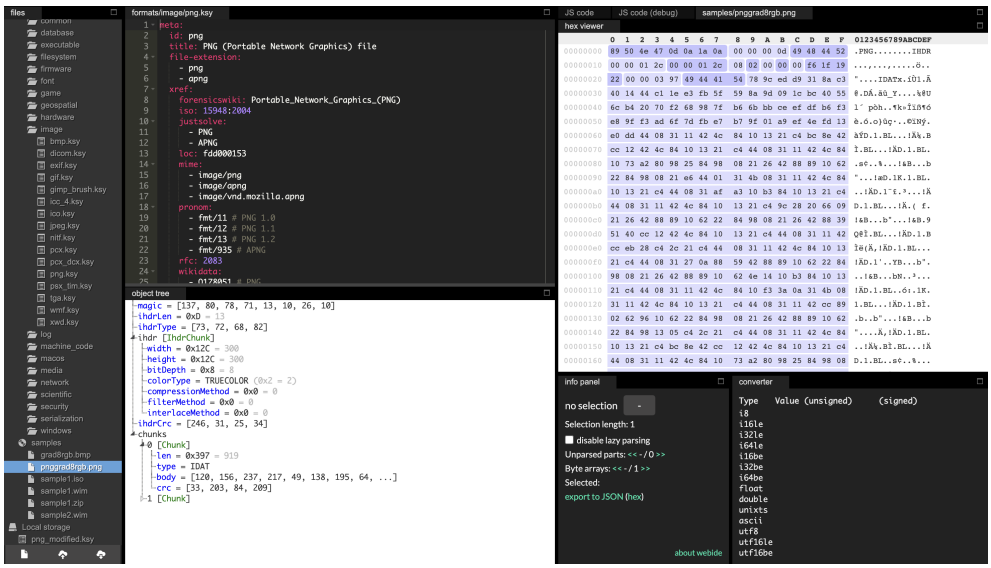

Figure 2.1: *Kaitai Struct Web IDE showing the ksy file of png file format and the input png file that gets parsed.*

2.2 Formal Definition of Problem

The problem involves creating a Binary Fuzzer which serves the critical purpose of automating the creation of test cases adhering to the specified format outlined in Kaitai Struct definitions (.ksy). The Binary Fuzzer incorporates a dynamic approach to test case generation by initially processing the input .ksy file to construct a grammar representation. This grammar, derived from the Kaitai Struct definition, encapsulates the structural rules, data types, and conditional logic specified within the file. Subsequently, the fuzzer utilizes this grammar to intelligently generate diverse and contextually relevant test cases from random streams of bytes. Through its logical approach and systematic generation of test cases, the Binary Fuzzer provides a valuable tool in handling a wide range of binary file formats.

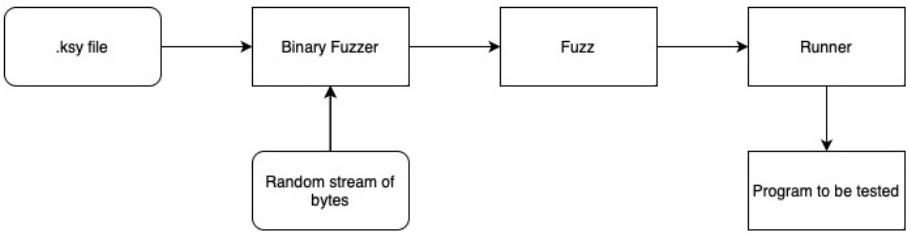

Figure 2.2: *Workflow diagram*

2.3 Illustration

Imagine you have developed a web application that takes user input as a binary file format, say png and processes it to generate a personalized poster. As part of your testing process, you want to ensure that your application handles various inputs correctly and doesn't crash or behave unexpectedly. So we need test cases that would be in the png file format. For this, we use the Binary Fuzzer. The Binary Fuzzer takes a .ksy file of png as input. This file defines the structure of the binary format of png using the Kaitai Struct language. The Binary Fuzzer also takes a random stream of bytes as input. This represents the raw binary data that needs to be parsed based on the rules defined in the .ksy file. Thus it produces the test cases that we require.

2.4 Potential Users

The Binary Fuzzer is useful for software developers, quality assurance teams, and security analysts. They can use it to validate their software requiring binary files as inputs.

2.5 Estimated user base after two years

After two years, Binary Fuzzer could attract a diverse user base. This might include developers from software development companies, QA teams across industries, and security professionals.