### Report on Fuzzer Development Progress

#### 1. Introduction

This report outlines the progress made in developing a SQL fuzzer. Two primary approaches have been explored so far: utilizing web scraping to open URLs and extract request details, and using Selenium WebDriver to count and inspect input elements on web pages.

The following Python script was implemented to open a given URL in the default web browser and extract request details using the requests library:

import webbrowser

import requests

def open\_and\_extract(url):

    # Open the URL in the default web browser

    webbrowser.open(url)

    # Extract request details

    response = requests.get(url)

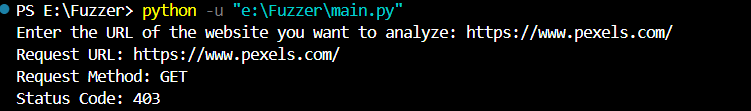
    # Print request details

    print(f"Request URL: {response.url}")

    print(f"Request Method: {response.request.method}")

    print(f"Status Code: {response.status\_code}")

RESULT:



Using Selenium WebDriver to Count and Inspect Input Elements

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.chrome.service import Service as ChromeService

from webdriver\_manager.chrome import ChromeDriverManager

# Initialize the WebDriver using webdriver\_manager

driver = webdriver.Chrome(service=ChromeService(ChromeDriverManager().install()))

# Function to count the number of input elements

def count\_input\_elements(url):

    # Open the URL

    driver.get(url)

    # Find all input elements

    input\_elements = driver.find\_elements(By.TAG\_NAME, 'input')

    # Print the number of input elements found

    print(f"Number of input elements: {len(input\_elements)}")

    # Optionally, you can print the details of each input element

    for i, element in enumerate(input\_elements, start=1):

        print(f"Input {i}: {element.get\_attribute('outerHTML')}")

# URL to be inspected

url = 'https://www.pexels.com/'

# Count the input elements on the given URL

count\_input\_elements(url)

# Close the WebDriver

driver.quit()

RESULTS:

