P R O J E C T D O C U M E N T

Test Plan / Test Cases Design Document

Project Name	코드 난독화 도구 제작

15 조

202002562 조인우
202002508 손지웅
201902686 노형우

지도교수: 조은선 교수님 (서명)

Table of Contents

1. Introduction

1.1. 연구 질문/ 가설

본 연구는 다음과 같은 연구 질문에 답하고자 한다:

- **RQ1.**
  제안하는 ANTLR 또는 LLVM 기반 선택적 난독화 도구는 기존 난독화 도구에 비해 AI 기반 복원 도구(LLM)에 대해 저항성을 얼마나 향상시키는가?
- **RQ2.**
  [제안하는 시스템/도구]는 [사용자 유형/상황]에 따라 [사용성/효과/만족도]에 차이를 보이는가?

본 연구는 다음과 같은 가설을 설정할 수 있다:

- **H1.**
  제안된 소스코드 수준 난독화 기법은 기존 난독화 방식(Tigress) 보다 LLM(GPT-4o) 기반 역난독화 시도에 대해 유의미하게 높은 복원 저항성을 보일 것이다.
- **H2.**
  ANTLR 기반 소스 코드 난독화와 LLVM 기반 IR 난독화는 각각의 적용 대상과 상황에 따라 보안성과 성능 측면에서 상이한 효과를 보일 것이다.

2. Test Plan

1. 배경과 목적

1.1 배경

최근 대형 언어 모델 (LLM, GPT-4o)의 발전으로 기존 코드 난독화 기법의 한계가 드러나고 있다. LLM은 난독화된 코드의 의미를 높은 정확도로 복원할 수 있어, 소프트웨어 보안 위협이 심화되고 있다. 이에 따라 코드의 구조 자체를 변형하는 난독화 기법의 필요성이 대두되고 있다.

본 테스트의 목적인 구조적 변형을 통한 구문 수준 코드 난독화 기법이 실제로 LLM기반 역난독화 저항성, 코드 복잡도, 실행 성능(시간) 등에 미치는 영향을 실험적으로 검증하는 것이다. 또한 제안 기법과 기존 난독화 도구(Tigress)간의 차이 효과 차이를 정량적으로 비교하고자 한다.

2. 테스트 상세

2.1 독립/ 종속 변수 정의

독립 변수

1. 원본 코드 (난독화가 적용되지 않은 원래의 코드)
2. ANLTR 기반 난독화 기법
3. 기존 난독화 도구 (Tigress의 다양한 옵션)
4. ALTLR + Tigress 복합 적용

종속 변수

1. LLM 기반 역난독화 성공 여부
2. 코드의 구조적 복잡도
3. 프로그램의 실행 시간
4. 난독화 후 코드의 가독성 (필요시)

2.2 실험 대상/ 환경

테스트용 miniC 언어 기반, C언어 기반 프로그램 10종 (조건문, 반복문, 함수 호출, assignment 등 다양한 구조를 포함하도록 한다.

하드웨어 : Apple Silcon M4, 16GB RAM, Mas OS 환경

소프트웨어 :

1. ANTLR 기반 구조적 난독화 도구 (java 기반)
2. LLM (GPT-4o)
3. 실행 시간 측정 (터미널 time 명령어)

네트워크 : LLM API 호출을 위한 인터넷 환경
3. 테스트 관리
3.1 실험 절차 요약
1. 테스트 코드 준비 2. 난독화 적용 3. 지표 측정 4. 데이터 정리 및 분석
3.2 측정 지표 및 도구
정량 평가 지표 : 사이클릭 복잡도, 실행 시간, LLM 기반 역난독화 성공 여부 정성 평가 지표 : 가독성 사용 도구 : ANTLR, Java, Tigress, LLM API, 복잡도 분석 도구

Test Cases

1. 테스트 케이스

1.1 테스트 케이스 명세

Id	대상(모델/조건)	실험 조건	테스트 데이터	평가지표	예상 결과
TC-1	원본 코드	난독화하지 않은 원본 코드	C 코드 10종	사이클릭 복잡도, 실행 시간, LLM 복원 여부	가장 낮은 복잡도, 최단 시간, LLM 복원 가능
TC-2	ANTLR 기반 난독화 코드	제안 도구 적용	동일	동일	복잡도 상승, 실행 시간 소폭 증가, LLM 복원률 감소
TC-3	Tigress - Flatten	Flatten 적용	동일	동일	복잡도 상승, 실행 시간 증가, LLM 복원률 일부 감소
TC-4	Tigress - AddOpaque	AddOpaque 적용	동일	동일	복잡도 상승, 실행 시간 증가, LLM 복원률 일부 감소
TC-5	Tigress - RenameIdentifiers	RenameIdentifiers 적용	동일	동일	복잡도 변화 미미, 실행 시간 변화 미미, LLM 복원률 거의 동일
TC-6	ANTLR + Tigress	두 기법을 순차적으로 적용	동일	동일	복잡도 최고, 실행 시간 증가, LLM 복원률 최저 예상

1.2 검증 기준(metirc)	
평가지표	평가 기준
사이클릭 복잡도	난독화 전후의 복잡도 차이로 평가, 높을수록 구조적 난이도 증가로 간주
실행 시간	난독화 전후 평균 실행 시간의 상대적 증가율로 평가
LLM 복원 성공률	LLM에 “코드를 원래대로 복원하라” 프롬포트 입력 후 난독화 구조 제거 여부로 평가.
가독성 (선택)	난독화 코드의 이해 난이도를 5점의 척도로 평가받아 평균값 산출

3. AI 도구 활용 정보

사용 도구	*GPT-4o*
사용 목적	*코드 역난독화 및 복원 저항성 실험, 소스코드 생성*
프롬프트	● *해당 코드를 원래대로 복원해줘* ● *간단한 C언어 코드 작성해줘.*
반영 위치	*1. 난독화 도구 평가* 2. Testcase 작성
수작업	*있음*
수정	*AI가 제공한 Test코드의 실행 검증, 논리 보강 등*