Fuzz Testing을 통한 위성 SW 분석

종합설계1 Week4

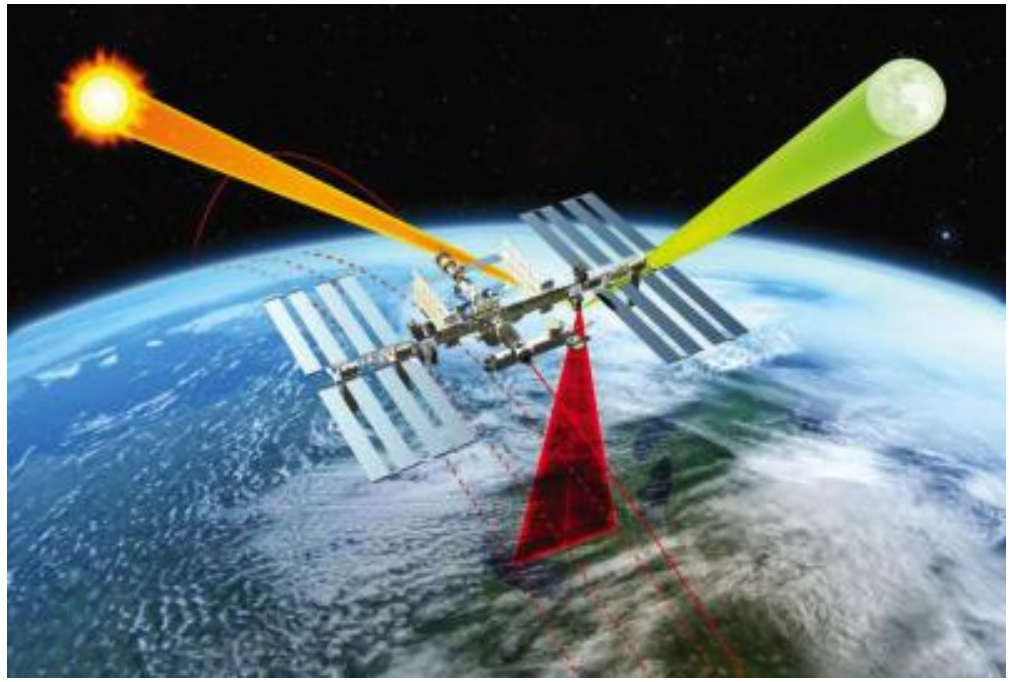
202002473 김승혁
201902733 이정윤
202002699 조민기

목차

연구 개발의 필요성

왜 필요한가 ?

- 위성 소프트웨어는 미션 크리티컬 시스템임.
- 사소한 오류에도 위성 기능을 상실할 가능성이 큼
- 대규모 위성 프로젝트는 이러한 오류 가능성을 모두 고려하고, 가능한 모든 방식을 동원하여 신뢰도를 100%에 가깝게 만들어 발사. (ex: 우주 방사선에 버틸 수 있는지 직접 위성에 방사선을 망가질 때까지 쪼임)
- But, 소형 위성의 경우 비용, 시간의 문제로 완벽한 검증을 마치지 못하고 발사하는 경우가 많음.



연구 개발의 필요성

왜 필요한가 ?

- 우주 산업의 문턱이 낮아짐 -〉 오픈소스 위성 비행 소프트웨어 활용 증가
- 그러
- Fuz
  Stat
  려한
- Fuzzing을 통한 위성 SW 테스트의 신뢰도가 떨어지는 상황임 .

위성 SW에 특화된 효과적인 Fuzzing 기법을 연구 !

연구 개발의 목표 및 내용

- 다양한 기법의 Fuzzer를 fprime에 적용
- fprime의 특성을 고려하여 Fuzzer의 성능을 정량적으로 평가하고 비교 분석
- 분석 결과를 토대로 fprime 환경에서 각 fuzzer의 장단점을 명확히 하고, 어떤 유형의 fuzzer가 fprime에 더 적합한지 연구

=〉 Fprime 기반 위성 소프트웨어의 신뢰성 검증을 위한 최적화된 fuzzing 적용 방안 + fprime 특화 fuzzer 개발에 필요한 요구사항 도출

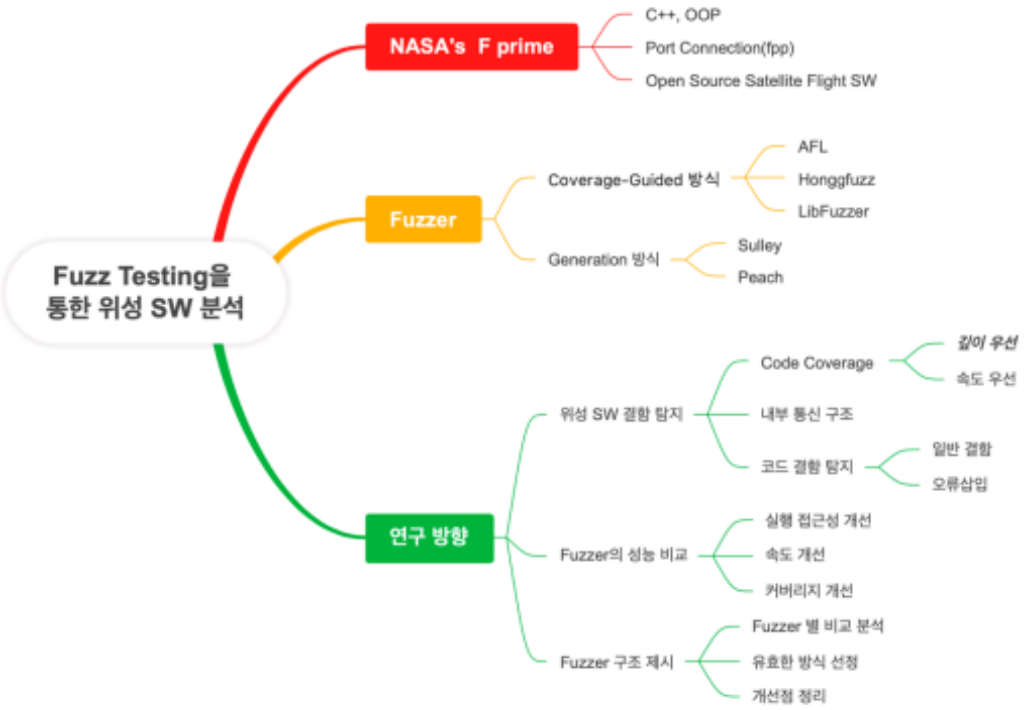


그림 1 브레인스토밍 결과

이해 당사자 인터뷰 / 설문 인사이트

- **이성호 교수님 (Software Analysis & Testing Research Group, 충남대학교)**

인터뷰 내용	● 위성 SW 테스팅: 기능 테스트, 유닛 테스트, 정적 분석 등이 활용되나 미션 크리티컬 SW 맞춤형 기법은 부족함. ● 퍼징 연구 방향: 퍼저 성능 비교(정량 지표 활용: 커버리지, 결함 탐지율 등) 후 실제 결함 탐색 시도 권장. 간단한 대상(fprime)부터 시작하여 경험 축적 후 복잡한 대상(cFS) 고려. ● 기존 퍼저 한계: 위성 SW의 특성(예: 모듈 간 통신)을 고려하지 못해 커버리지 확보 및 모듈 간 상호작용 결함 탐지에 한계가 있을 수 있음.
인사이트	**대상 SW(fprime)를 실제 구동해보고 단계적으로 퍼저를 적용해보는 것이 가장 중요하며, 실제 문제를 직접 부딪혀야 문제 정의 및 해결 방향이 명확해지기 때문에, 정량적 평가 지표 설정을 통해 분석해 봐야한다.**

이해 당사자 인터뷰 / 설문 인사이트

- **김형신 교수님 (Embedded Systems Laboratory, 충남대학교)**

인터뷰 내용	● 위성 SW 오류의 치명성: 온보드 컴퓨터 오작동은 치명적. 특히 자세 제어 실패로 인한 배터리 방전, 잘못된 명령 처리로 인한 연료 낭비 등은 위성 기능 상실 또는 임무 실패로 이어질 수 있음. ● 오류 발생 원인: 방사선, 온도 등 환경적 요인도 있지만, 기록상 가장 많은 원인은 운영자의 잘못된 명령 전송(Operator Error)임. 소프트웨어 자체 결함도 원인이 될 수 있음. ● 위성 시스템 특징: 고신뢰성 설계(하드웨어/소프트웨어 리던던시), 자동 복구 기능(와치독 등) 존재. 하지만 비정상적인 상태(예: 잘못된 컨트롤 로직 수행)는 탐지 및 복구가 어려울 수 있음. 배터리 방전 시 원격 복구 불가능. ● 사전 테스트 중요성: 발사 전 지상에서의 철저한 테스트(정적/동적 테스트, 고장 주입 시험 등)가 매우 중요함. 대학 개발 위성은 검증 부족으로 실패 확률 높음. 퍼징은 다양한 테스트 방법 중 하나로, 특히 예상치 못한 입력에 대한 강건성 검증에 기여할 수 있음.
인사이트	**위성 SW는 일반 SW와 다른 특수성을 가지고 있으며, 테스트에서도 이를 반영하는 효과적인 테스트가 매우 중요하다.**

이해 당사자 인터뷰 / 설문 인사이트

- **Jakob Holst Svenningsen (퍼징 연구 경험자(MMS-Fuzzer), DMC)**

인터뷰 내용	• 연구 경험: 상태 기반 시스템(MMS) 대상 네트워크 프로토콜 퍼저 개발 경험. • 퍼저 유형 비교: Mutation 기반 퍼저는 사용이 간편하고 상태 없는 시스템(Stateless)에 효과적이나, 상태 기반 시스템(Stateful)의 복잡한 구조나 취약점 식별에는 한계가 있음. Generation 기반/Genetic 퍼저가 상태 기반 시스템에 더 적합할 수 있음. • 퍼징 연구 어려움: 기존 라이브러리(Kitty, Boofuzz 등) 문서 부족, 좋은 테스트 케이스 설계의 어려움 (단순 랜덤 입력 이상의 설계 필요). • 평가 지표: 실행 시간 대비 발견된 취약점 수(효율성), 서버 응답 시간 (DoS 취약점 관련) 등. 시스템의 중요도에 따라 결과 해석 필요. • 개발 조언: 대상 프로토콜(fprime의 fpp 등)에 대한 깊은 이해가 중요함. 기존 라이브러리 활용 권장.
인사이트	**대상 SW에 특성(프로토콜 등)을 잘 분석한 뒤, 이에 맞는 Fuzzing 전략을 찾아 적용하는 것이 가장 중요하다.**

기대 효과 및 향후 확장 가능성

- Fprime 신뢰성 검증 효율성 증대
- Fuzzer 성능 비교 데이터 제공
- Fprime 특화 Fuzzer 개발
- 다른 위성 SW 적용 가능성 연구 (ex: cFS)

연구 개발의 추진전략 및 방법

- 4월 초: fprime 실행 환경 설정 완료
- 4월 중: fprime 구조 분석
- 4월 말: 1차 Fuzzer 적용 및 결과 분석
- 5월 초: 2차 Fuzzer 적용 및 분석
- 5월 중: 1차, 2차 비교 분석 및 3차 Fuzzer 적용
- 5월 말: 연구 결과 분석 및 논문 초안 작성

협업은 Github로 진행. 각자 특정 Fuzzer를 사용해 분석한 결과를 공유하고 연구 방향을 정하는 주간 회의를 매주 가질 것. 대면 + 비대면 진행

AI 도구 활용 정보

사용 도구	네이버 클로바노트, Claude 3.7, o3-mini
사용 목적	인터뷰 전사문 생성, 인터뷰 내용 요약
프롬프트	● 인터뷰 내용을 정리해줘 ● 작성한 문단에서 문법에 틀린 표현이 있는지 검사해줘
반영 위치	1. 인터뷰 질문 목록 (p.7) 2. 연구 개발의 필요성 (p.5)
수작업 수정	있음(논리 보강, 인터뷰 내용 정정)

참고문헌(Reference)

1. Incorporating security practices into the development of the Maritime Messaging Service – Jakob Holst Svenningsen (2024)
2. 인공 위성 탑재 소프트웨어의 소프트웨어 고장 주입시험 - 배지훈, 김형신 (2022)

Thank You