ROS2 환경에서 활용된 퍼징 도구에 관한 조사

김은민 [1], 서지원 [2]
[1] 단국대학교 사이버보안학과 학부생
[2] 단국대학교 사이버보안학과 교수
emkim@dankook.ac.kr, jwseo@dankook.ac.kr

A Survey on Fuzzing Tools Deployed in ROS2

Eunmin Kim[1], Jiwon Seo[1]
[1]Dept. of Cyber Security, Dankook University

요 약

본 논문은 ROS2 환경에서 사용 가능한 다양한 퍼징 도구들을 조사함으로써, 이 분야의 연구 동향을 분석하고자 한다. 최근 로보틱스의 활용 범위가 확대됨에 따라 ROS 보안 취약점에 대한 우려 또한 증가하고 있다. 이에 따라, ROS2 환경에 특화된 다양한 퍼징 테스트 도구들의 기능, 효율성 및 적용 사례를 비교 분석하여 ROS2 의 보안 강화에 기여할 수 있는 방안을 모색한다. 또한, 현존하는 퍼징 도구들의 한계를 진단하고, 이를 개선한 새로운 퍼징 도구 개발의 필요성에 대해 논의한다.

1. 서론

로봇 시스템의 활용 범위가 확대됨에 따라, 로봇 개발을 위한 오픈소스 소프트웨어 플랫폼인 Robot Operating System (ROS) 의 쓰임도 늘어나고 있다. 그러나 ROS 는 메세지 암호화, 인증, 권한 설정 등의 별도의 보안 조치 사항이 존재하지 않는다[1]. 이와 같은 보안 요구 사항의 부재로, ROS 를 사용하는 대부분의 로봇 시스템에 보안 구멍이 생길 수 밖에 없는 상황이다.

이러한 문제를 해결하기 위한 방법으로 퍼징 테스트가 주목받아오고 있다. 퍼징 테스트는 소프트웨어에 의도적으로 잘못됐거나 예기치 않은 입력을 넣어 잠재적인 취약점을 찾아내는 테스트 기법이다. 기존 소프트웨어 환경에서 퍼징은 일반적으로 입력 데이터의 유효성을 검증하고, 예외 처리 능력을 평가하는 데 사용된다. 그러나 ROS 환경에서의 퍼징은 더욱 복잡하다. ROS 는 여러 노드가 서로 메시지를 주고받으며 동작하는 분산 시스템이기 때문에, 단순한 데이터 형식의 검증을 넘어서 메시지의 구조와 그 처리 방식 자체가 중요한 퍼징 대상이 된다. 이로 인해, ROS의 독특한 구조와 기능을 이해하고 그에 특화된 퍼징 도구를 개발하는 것이 필수적이다.

이와 같은 이유로, 이미 ROS2 환경에서 사용가능한 다양한 퍼징 도구에 대한 연구가 존재한다[2,3,4,5]. 본 논문에서는 이러한 도구들의 특성을 비교하고 분석하여, 현재의 보안 도전을 해결하고 향후 보다 효과적인 ROS 퍼징 도구 개발을 위한 기초 자료를 제공하고자 한다.

2. 배경지식

2.1. 퍼징 진행 과정

퍼징은 크게 1) 입력값 생성 2) 테스트 적용 3) 입력값 변형 및 피드백으로, 총 세 단계로 나누어 생각해 볼 수 있다.

먼저 입력값 생성에서는 퍼징 대상 시스템에 입력할 값을 생성한다. 무엇을 입력값으로 선정할 것인지가 관건이다. 이후, 테스트 적용에서는 앞서 생성한 입력값을 퍼징 대상 시스템에 전달하여 테스트를 수행한다. 마지막 단계인 입력값 변형 및 피드백에서는 전 단계의 입력값에 대한 피드백을 기반으로 기존의 입력값에서 변형을 하여 새로운 입력값을 생성하는 단계이다. 이 과정을 반복하여 계속 루프를 돌며 테스트를 진행하게 된다.

3. ROS 퍼징 도구 비교

표 1 에서 확인할수 있다시피, 본 논문에서는 2020 년에서부터 2024 년까지의 퍼징 도구, 퍼저를 비교

분석하였다. 초점을 두고 분석한 부분은 퍼징 입력값과 무엇을 중점으로 피드백을 진행하였는지, 어떠한 버그를 타겟팅하였는지 등이다.

4. 퍼징 도구 별 특징

입력값의 'single'과 'multiple'은 퍼징의 입력값을 무엇으로 설정했는지에 따라 달라진다. single 에 해당하는 값은 사용자 커맨드나 센서 정보, 형상 파일 등을 그대로 제공하는 일차원적인 값이다. 그에 반해 multiple 에 해당하는 값은 제공된 입력값에 대해 도달하는 순서를 변경하여 전달하는 등의 값이다.

피드백 방식에서도 차이가 있다. Code coverage 피드백 기반인 ros2_fuzz[2]와 ROZZ[4]는 프로세스의 수가 많아지거나 실행 순서가 뒤섞이면, 피드백을 위한 작업이 늘어나게 된다. RoboFuzz[3]의 경우는 이론과 실제 작업 환경의 불일치가 흔하게 발생할 수 있다는 로봇 시스템의 특성을 고려하여 두 환경 모두에서 실행하여 피드백 하는 semantic feedback 을 채택하였다. 다만, 모든 실행에서 나타날 수 있는 것들을 state 로 정의해야 하는 번거로움이 있다. ROFER[5]의 메세지 기반 피드백은 ROS 의 노드 간 통신 특징에 따라 메세지로 인해 state 가 상시 달라질 수 있다는 점에서 착안한 피드백 방식이다.

2020 년부터 2024 년까지의 네 가지 퍼저들 모두 ROS2 Foxy 버전을 기반으로 제공하고 있다.

<표 1> 최신 ROS2 퍼저들 비교 정리

퍼징 도구	입력값	피드백	ROS 버전	년도
ros2_fuzz[2]	single	code coverage	foxy	2020
RoboFuzz[3]	single	semantic feedback	foxy	2022
ROZZ[4]	multiple	code coverage	foxy	2022
ROFER[5]	multiple	message feature	foxy	2024

5. 결론

오늘날 ROS 는 로봇 개발을 위한 오픈소스 소프트웨어 플랫폼으로 널리 사용되고 있지만, 보안을 고려하지 않고 설계된 탓에 여러 보안 취약점이 존재한다. 이를 해결하기 위해 다양한 퍼징 기법들이 제안되고 있으며, 이는 ROS 의 보안 강화를 위한 중요한 방향성을 제시하고 있다. 대부분 주어진 입력값에 대해 어떤 변환을 적용했는지, 들어간 값에 대한 피드백으로 무엇을 채택했는지에 대한 부분만 상이했다. 기존의 코드 커버리지 기반 피드백에서 메세지 기반 피드백으로 바뀐 부분은 ROS 의 특성을 잘 반영하여 퍼징 효율을 향상시켰다.

아쉬운 점은 2020 년부터 가장 최근인 2024 년도 퍼저[5]도 Foxy 버전에서 제공된다는 것이다. 현재 Foxy 는 최신 운영 체제와의 호환성 제한, 라이브러리 및 미들웨어의 구버전 의존성 등으로 인해 새로운 하드웨어 및 소프트웨어 환경에서의 활용도가 제한적이다. 또한 보안 향상과 관련된 최신 기술의 통합이 더딘 점도 걸림돌이다.

이러한 배경을 바탕으로, 앞으로 Humble 기반의 퍼저 개발을 목표로 할 것이다. Humble 버전은 보다 광범위한 운영 체제 지원과 함께 최신 라이브러리 및 개발 도구와의 호환성을 제공하므로, 더욱 효과적이고 광범위한 퍼징 테스트가 가능할 것이다. 이를 통해, ROS2 환경에서의 안전성과 신뢰성을 향상시킬 수 있을 것으로 기대된다.

사사문구

본 연구는 2024 년 과학기술정통신부 및 정보통신기획평가원의 SW 중심대학사업 지원을 받아 수행되었음(2024-0-00035)

참고문헌

[1] Teixeira, Rafael R., Igor P. Maurell, and Paulo LJ Drews., "Security on ROS: analyzing and exploiting vulnerabilities of ROS-basedsystems", Latin American robotics symposium (LARS) Brazilian symposium on robotics (SBR) and workshop on robotics in education (WRE), Natal, Brazil, 2020, p.1-6.

[2] JnxF, and Gavanderhoorn. ros2_fuzz. GitHub, n.d., https://github.com/rosin-project/ros2_fuzz. Accessed 5 September 2024.

[3] Kim, Seulbae, and Taesoo Kim. "RoboFuzz: Fuzzing robotic systems over robot operating system (ROS) for finding correctness bugs." Proceeding of 30th ACM Joint European Software Engineering Conference and Symposium on the foundations of Software Engineering., Singapore, Singapore, 2022, p.447-458.

[4] Xie, Kai-Tao, et al. "ROZZ: property-based fuzzing for robotic programs in ROS." 2022 International Conference on Robotics and Automation (ICRA). IEEE, Philadelphia, USA, 2022.

[5] Bai, Jia-Ju, Hao-Xuan Song, and Shi-Min Hu. "Multi-Dimensional and Message-Guided Fuzzing for Robotic Programs in Robot Operating System." Proceedings of the

29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2. Rotterdam, Netherlands, 2024, p.763-778.