Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.34, NO.4, Aug. 2024 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2024.34.4.681

ROS 침입 탐지 시스템을 위한 공격 데이터셋 구축*

김 형 훈,[1†] 이 승 민,[1] 허 재 웅,[2] 조 효 진[3‡]
[1,3]연세대학교 (대학원생, 교수), [2]숭실대학교 (연구원)

Attack Datasets for ROS Intrusion Detection Systems*

Hyunghoon Kim,[1†] Seungmin Lee,[1] Jaewoong Heo,[2] Hyo Jin Jo[3‡]
[1,3]Yonsei University (Graduate student, Professor), [2]Soongsil University (Researcher)

요 약

최근 수년 동안 무인지상차량 및 무인항공기와 같은 Robotics 분야에 대한 연구 및 개발이 활발히 진행되고 있다. 이러한 발전에는 서로 다른 애플리케이션 간의 통신 및 데이터 관리를 원활하게 해주는 미들웨어의 사용이 중요한 역할을 하고 있으며, 여러 가지의 산업용 통신 미들웨어 프로토콜들이 출시되고 있다. 그중 ROS (Robot Operating System)가 로봇 시스템 개발을 위한 주요 플랫폼으로 널리 사용되고 있지만, 초기 설계 과정에서 보안 측면을 전혀 고려하지 않았기 때문에 통신을 도청하거나 악의적인 메시지를 주입하는 등의 다양한 공격에 취약한 상태이다. 이에 대응하기 위해, ROS에 대한 보안 솔루션을 제안하는 많은 연구가 진행되고 있으며, 특히 침입 탐지 시스템을 위한 ROS 데이터셋을 제안하는 연구도 진행되었지만, 이와 같은 연구는 매우 부족한 상황이다. 본 논문에서는 ROS 침입 탐지 시스템의 성능 발전에 기여할 수 있도록 ROS 환경에서 발생 가능한 새로운 유형의 공격 시나리오를 제안하고, 실제 로봇 시스템으로부터 수집한 ROS 공격 데이터셋을 구축하며 오픈 데이터셋으로 제공한다.

ABSTRACT

In recent decades, research and development in the field of industrial robotics, such as an unmanned ground vehicle (UGV) and an unmanned aerial vehicle (UAV), has been significant progress. In these advancements, it is important to use middleware, which facilitates communication and data management between different applications, and various industrial communication middleware protocols have been released. The robot operating system (ROS) is the most widely adopted as the main platform for robot system development among the communication middleware protocols. However, the ROS is known to be vulnerable to various cyber attacks, such as eavesdropping on communications and injecting malicious messages, because it was initially designed without security considerations. In response, numerous studies have proposed countermeasures to ROS vulnerabilities. In particular, some work has been proposed on generating ROS datasets for intrusion detection systems (IDS), but there is a lack of research in this area. In this paper, in order to contribute to improving the performance of ROS IDSs, we propose a new type of attack scenario that can occur in the ROS and build ROS attack datasets collected from a real robot system and make it available as an open dataset.

Keywords: Robot Operating System, Attack Dataset, Intrusion Detection System, Cyber Security

I. 서 론

Robotics 기술은 제조, 물류, 유통 등 공장 자동화를 위한 산업용 로봇, 수술 보조 및 원격 상담 서비스를 제공하는 의료용 로봇, 무인이동차량과 같이 인간을 대신하여 우주 및 해양을 조사할 수 있는 탐

Received(06. 07. 2024), Modified(07. 22. 2024), Accepted(07. 23. 2024)
* 본 연구는 2024년 정부(방위산업체)의 재원으로 국방과학연구소의 지원을 받아 수행된 연구임 (UI2200575D)
† 주저자, axolotl0210@gmail.com
‡ 교신저자, hyojin.jo@yonsei.ac.kr(Corresponding author)

사용 로봇 등 다양한 분야에서 사회적으로 많은 이점을 제공하기 위해 연구 및 개발이 활발히 진행되고 있다[1]. 이러한 Robotics 분야의 발전에는 이기종 하드웨어 간의 원활한 통신을 할 수 있도록 지원하는 미들웨어의 사용이 중요한 역할을 하고 있으며, MQTT (Message Queueing Telemetry Transport[2]), ROS (Robot Operating System[3]), ZeroMQ[4] 등 여러 가지의 산업용 통신 미들웨어 프로토콜이 출시되고 있다[5].

출시된 산업용 통신 미들웨어 프로토콜 중, ROS가 Robotics 분야에서 가장 널리 사용되고 있다[6]. ROS는 로봇 시스템 개발을 위한 메타 운영체제로, 프로세스 간의 메시지 전달, 패키지 관리 등 개발 환경에 필요한 각종 기능을 제공하는 미들웨어이다. 또한, 오픈소스로 공개되어 있기에 확장성 및 유연성이 좋고, 비용을 절감할 수 있으며, 다양한 커뮤니티에서 사용함으로써 버그를 신속하게 해결할 수 있는 등의 장점이 있다[1][7]. 하지만, ROS는 초기 설계 단계에서 보안 메커니즘인 암호화, 인증, 접근 제어 등을 고려하지 않았기 때문에, 이러한 취약점을 악용함으로써 통신을 도청하거나 악의적인 메시지를 주입하는 등의 다양한 공격을 수행할 수 있게 된다. 이와 관련하여 실제 ROS 환경에서 공격을 수행한 연구 사례가 다수 진행되었다[8-10].

ROS의 취약한 부분을 보완하기 위해, 보안 메커니즘[11][12]을 적용할 수 있는 방법론을 제안하거나 침입 탐지 시스템[13][14]을 제안하는 등의 여러 연구가 제안되고 있다. 또한, ROS를 사용하는 실제 로봇 시스템 환경에서 공격을 수행한 후, 데이터셋을 구축하여 제공하는 연구도 진행되었다[15]. 이와 같이, 데이터셋을 제공하는 연구는 침입 탐지 시스템의 성능 개선 등 보안 솔루션 개발에 큰 도움을 제공할 수 있으므로 다른 분야에서는 활발히 연구되고 있지만[16][17], ROS 분야에서는 데이터셋을 제공하는 연구가 단 하나에 불과하여 해당 분야의 연구는 아직 부족한 상황이다[15]. 추가적으로, ROS의 성능 및 보안 측면에서 미약한 부분을 보완하고자 DDS (Distributed Data System)를 미들웨어로 채택한 ROS2[18]가 출시되었으며, 최근에 ROS에서 ROS2로 전환하고 있는 추세이다[19]. 하지만 여전히 ROS를 사용하고 있는 곳이 많으며, ROS2가 주요 자동화 작업에 널리 배포되려면 시간이 오래 걸릴 것으로 예상된다[20]. 이를 통해, 안전하고 신뢰할 수 있는 ROS 환경을 제공하기 위해서는 ROS에 대한 보안 연구가 지속적으로 수행되어야 한다는 것을 볼 수 있다.

이에 비추어, 본 논문에서는 ROS에 대한 침입 탐지 시스템 개발 및 연구에 도움을 제공할 수 있도록, ROS 환경에서 발생 가능한 새로운 공격 시나리오를 제안하며, 실제 로봇 시스템 환경에서 제안한 공격 시나리오를 기반으로 공격을 수행하고 데이터셋을 구축한다. 기존에 ROS 공격 데이터셋을 제공하는 연구[15]와 비교하였을 때, 본 논문에서는 3가지의 새로운 공격 시나리오를 더 포함한 데이터셋을 제공한다. 이를 통해, ROS 환경에서의 침입 탐지 시스템 연구를 더욱 발전시킬 수 있을 뿐만 아니라 데이터의 다양성도 확보할 수 있다. 추가적으로, 본 논문에서 구축한 공격 데이터셋을 오픈 데이터셋으로 제공하며, 해당 데이터셋은 https://sites.google.com/view/gsi-cpss-lab/datasets에서 사용할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 ROS에 대한 배경지식을 다루고, 3장에서는 ROS 데이터셋 관련 연구를 소개한다. 4장에서는 본 논문에서 제안하는 새로운 유형의 ROS 공격 시나리오와 구축한 ROS 공격 데이터셋에 대해 설명한다. 마지막으로 5장에서는 결론을 맺는다.

II. 배경지식

2.1 ROS 구성 요소

ROS는 로봇 시스템 개발을 단순화하고, 코드의 재사용성을 촉진하는데 목적을 두고 있다[7]. 이를 달성하기 위해, ROS는 로봇 응용 프로그램을 개발하고 실행하는데 필수적인 다양한 구성 요소로 이루어져 있다. 이번 장에서는 ROS의 핵심 구성 요소 5가지 (i.e., Master, Node, Topic, Service, Action)에 대해 설명한다.

- Master: Node 간의 메시지 통신 및 정보 교환을 관리하는 중개자 역할을 수행함. 만약 ROS 내에 Master가 없을 경우, Node는 서로 메시지를 송·수신하거나 Service를 호출할 수 없음.
- Node: ROS의 기본적인 실행 단위로 하나의 프로세스를 의미함. 여러 개의 Node가 네트워크를 형성하여 로봇 시스템을 구축하며, 각 Node는 독립적으로 실행됨.

- Topic: ROS의 대표 통신 메커니즘으로, Node 간 메시지를 연속적으로 송·수신할 때 사용되는 비동기식 단방향 통신 메커니즘이며, Publish와 Subscribe 형태의 구조를 지님.
- Service: Request와 Response 형태의 동기식 양방향 통신 메커니즘으로, Client가 특정 작업을 요청하면 Server는 요청받은 작업을 처리하고 결과를 Client에게 전달하는 구조를 지님.
- Action: 목표 달성을 위한 비동기식 양방향 통신 메커니즘으로, 요청 후 응답까지 시간이 오래 걸리는 장기 작업을 처리하거나 중간 상태 피드백이 필요한 경우 사용됨.

2.2 ROS 통신 구조

ROS는 Node 간 메시지 교환을 중심으로 하는 통신 메커니즘을 제공하며, 3가지 통신 방법 (e.g., Topic, Service, Action)이 있다. 그중 Topic 통신 메커니즘이 가장 많이 사용되며, 통신 구조는 그림 1과 같다. 참고로, Service와 Action 통신 메커니즘도 Topic과 비슷한 절차로 진행된다.

먼저, 모든 Node 간의 통신을 제어 및 관리하기 위해 ROS Master를 실행한다. 다음으로, Topic을 Publish 하려는 Node는 구동과 함께 Node 명, Topic 명, Port 번호 등 여러 정보를 ROS Master에게 등록한다. Topic을 Subscribe 하려는 Node도 동일하게 ROS Master에게 여러 정보를 등록하며, ROS Master는 Subscriber Node에게 Publisher Node에 대한 정보를 알려준다. Subscriber Node는 ROS Master로부터 받은 정보를 기반으로 Publisher Node에게 접속 요청을 하게 되며, Publisher Node는 TCPROS (Transmission Control Protocol ROS) 통신을 위한 URI (Uniform Resource Identifier)와 Port 정보를 Subscriber Node에게 전송한다. Subscriber Node는 해당 정보를 통해 TCPROS를 이용하여 Publisher Node에게 다시 접속 요청을 함으로써 Socket 연결이 이루어지게 된다. 이로써, Subscriber Node는 Publisher Node로부터 메시지를 수신할 수 있게 된다. 추가로, 메시지는 데이터에 대한 유형과 값을 설명하기 위해 구조화된 형식으로 구성되어 있으며, Topic은 메시지 유형을 식별하는 역할을 수행한다.

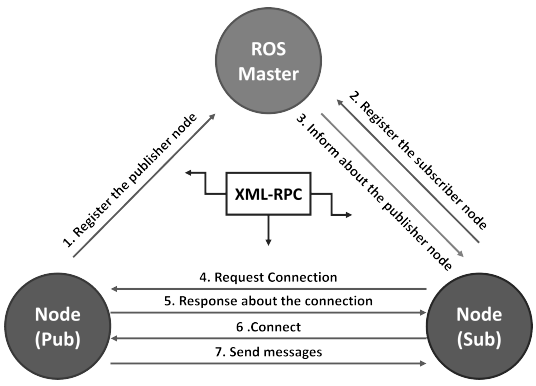

Fig. 1. Topic communication in ROS

III. 관련 연구

E. Değirmenci et al.[15]은 ROS를 사용하는 실제 로봇 시스템 환경에서 공격을 수행하고 데이터셋을 수집한 후, 오픈 데이터셋 (i.e., ROSIDS23)으로 제공하였다. ROSIDS23에는 4가지 공격 시나리오가 포함되어 있다. 첫 번째는 Denial of Service (DoS) 공격으로, 다량의 네트워크 패킷을 발생시키거나 시스템 리소스를 과도하게 소모시킴으로써, 정상 Node가 시스템에 접근하는 것을 방해한다. 두 번째는 Unauthorized Publish 공격으로, 권한이 없는 공격자가 ROS 라이브러리를 사용하여 새로운 Node를 생성한 후, 악성 메시지를 주입함으로써 로봇 시스템의 오작동을 유도한다. 세 번째는 Unauthorized Subscribe 공격으로, 로봇 시스템의 물리적인 동작에 영향을 주는 것은 아니지만, ROS로 통신되고 있는 모든 데이터를 획득하기 위해 공격자는 ROS 라이브러리를 사용하여 Subscribe 하는 새로운 Node를 생성한다. 마지막으로 Subscriber Flood 공격은 DoS 공격의 일종으로, 공격자가 수많은 Node를 생성한 후, 특정 Topic에 대해 Subscribe 하는 공격이다. ROSIDS23는 앞서 설명한 4가지의 공격 시나리오를 기반으로 데이터셋을 구축하였으며, CICFlowMeter[21] 도구를 통해 수집한 데이터셋으로부터 83개의 Feature를 추출하고 라벨링을 진행하였다.

본 논문에서는 ROSIDS23에서 제공한 4가지의 공격 시나리오가 포함된 데이터셋 보다 더 다양한 공격 시나리오가 포함된 데이터셋 제공을 목표로 한다.

IV. ROS 공격 시나리오 및 데이터셋

이번 장에서는 ROS 환경에서 발생할 수 있는 새로운 유형의 공격 시나리오와 이를 기반으로 구축한 ROS 공격 데이터셋에 대해 상세히 설명한다.

4.1 공격 시나리오

제안하는 새로운 유형의 공격 시나리오는 ROS 노드를 생성하여 공격하는 With a Fake Node 공격과 노드를 생성하지 않고 공격하는 Without a Fake Node 공격으로 크게 2가지로 구성된다. 해당 공격 기법에 대한 설명은 아래와 같다.

4.1.1 With a Fake Node 공격

With a Fake Node 공격은 권한이 없는 공격자가 ROS 통신을 하는 새로운 Fake Node를 생성하여 메시지를 주입하는 기법이다. 기존 ROS 공격 연구[9]에서 이미 Fake Node를 생성하여 공격하는 기법 (i.e., Unauthorized Publish)을 제안하였지만, [9]에서는 Fake Node를 생성할 때 Node 명과 Topic 명을 어떻게 설정하였는지 그리고 이에 따라 어떠한 현상이 발생하는지에 대해서 다루고 있지 않다. 본 논문에서는 With a Fake Node 공격의 2가지 방법과 각 공격에 따른 영향 및 결과에 대해서 설명한다.

With a Fake Node 공격의 첫 번째 방법은 SNST (Same Node Same Topic) 공격이다. SNST 공격은 공격하고자 하는 Publisher Node의 Node 명과 Topic 명을 동일하게 설정하여 새로운 Fake Node를 생성하고 메시지를 주입하는 방법이다. 그림 2처럼, Node 명이 "Talker"이고, Topic 명이 "Chatter"인 메시지를 Publish 하는 Node와 해당 메시지를 Subscribe 하는 Node가 존재한다고 가정하였을 때, SNST 공격을 수행할 경우, 기존 Publisher Node는 연결이 끊기게 되어 더 이상 메시지를 전송할 수 없게 된다. 이로 인해, Subscriber Node는 공격자가 새롭게 생성한 Fake Node가 전송하는 메시지만 수신하게 된다. 실제로 실험을 진행하였을 때, 그림 4 (좌)를 보는 바와 같이, Subscriber Node는 Publisher Node가 전송하는 "hello world: N" (N은 0부터 1씩 증가하는 정수를 의미함) 메시지를 정상적으로

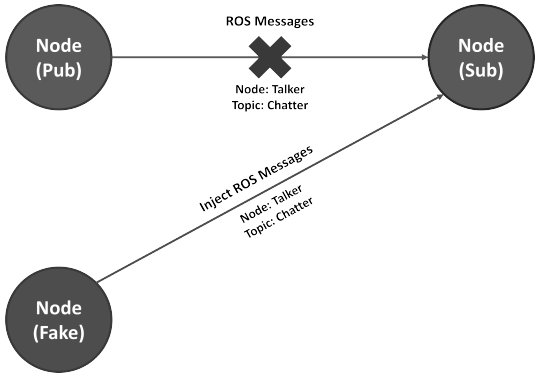

Fig. 2. SNST attack with a fake node

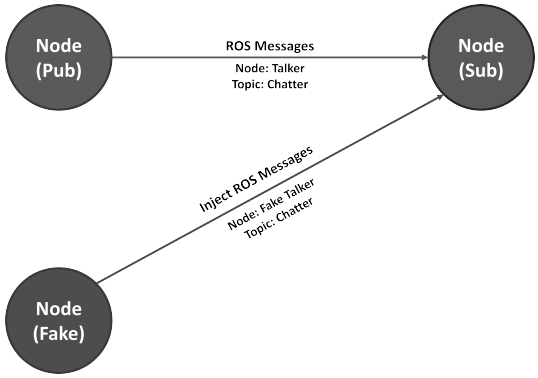

Fig. 3. DNST attack with a fake node

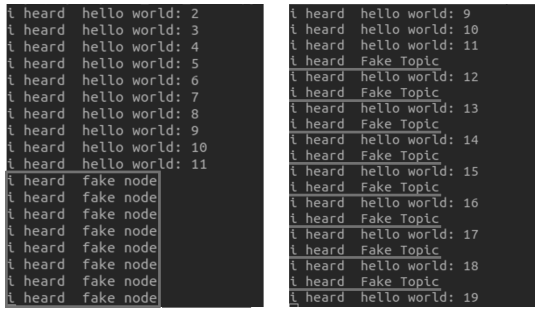

Fig. 4. Results of attacks with a fake node ((left) SNST attack / (right) DNST attack)

수신하는 도중에 특정 시점부터 Fake Node가 전송하는 "fake node" 메시지만을 수신하는 것을 볼 수 있다. 참고로, 이러한 현상은 ROS의 특성으로 인해 발생되는 것이며[22], 기존 Publisher Node를 재시작하게 되면 Fake Node의 연결이 끊기고 Subscriber Node와의 통신이 재개된다.

With a Fake Node 공격의 두 번째 방법은

DNST (Difference Node Same Topic) 공격이다. DNST 공격은 공격하고자 하는 Publisher Node의 Node 명은 다르게 설정하고 Topic 명은 동일하게 설정하여 새로운 Fake Node를 생성한 후 메시지를 주입하는 방법이다. 그림 3과 같이, DNST 공격을 수행할 경우, Subscriber Node는 정상 Publisher Node와 새롭게 생성된 Fake Node가 전송하는 모든 메시지를 수신하게 된다. 실험을 진행한 결과, 그림 4 (우)를 보는 바와 같이, Subscriber Node는 기존 Publisher Node와 Fake Node로부터 송신된 모든 메시지들을 번갈아 가며 수신하는 것을 볼 수 있다. DNST 공격을 이용하면 SNST 공격처럼 기존 Publisher Node의 연결 끊김 현상 없이 데이터 변조 및 서비스 거부 공격 등을 수행할 수 있다.

4.1.2 Without a Fake Node 공격

Without a Fake Node 공격은 권한이 없는 공격자가 ROS 통신을 하는 새로운 Fake Node를 생성하고 메시지를 주입하는 것이 아닌 네트워크 프로토콜을 이용하여 패킷을 전송함으로써 메시지를 주입하는 기법이다. 해당 공격 기법에도 2가지의 방법이 존재하며, 첫 번째 방법은 TCP를 이용한 공격이고, 두 번째는 UDP (User Datagram Protocol)를 이용한 공격이다.

Without a Fake Node 공격의 첫 번째 방법인 TCP Injection 공격은 그림 5와 같이 공격자가 Subscriber Node에게 TCP 패킷을 직접 전송함으로써 ROS 메시지를 수신할 수 있게 하는 방법이다. 해당 공격을 성공적으로 수행하기 위해서는 TCP 패킷의 SEQ (Sequence Number)를 동기화하는 절차가 필요하며, 과정은 그림 6과 같다. 공격자는 먼저 Publisher Node와 Subscriber Node 사이에서 TCPROS로 전송되는 패킷을 모니터링 하며, SEQ, ACK (Acknowledgement Number), Data의 크기 및 구조를 파악한다. SEQ는 이전 패킷의 ACK의 값으로 설정되며, ACK는 SEQ의 값에서 Data 크기만큼 더한 값으로 설정되는 것을 확인할 수 있다. 또한, ROS 메시지는 지정된 필드로 구조화되어 있기 때문에, 공격자가 원하는 결과를 얻기 위해서는 패킷의 페이로드 즉, Data의 어느 부분이 무엇을 의미하는지 정확히 파악하는 과정이 필요하다. 이렇게 헤더와 Data를 설정하고 TCP 패킷

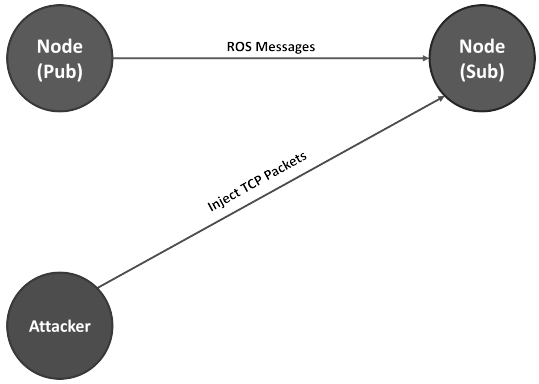

Fig. 5. TCP injection attack without a fake node

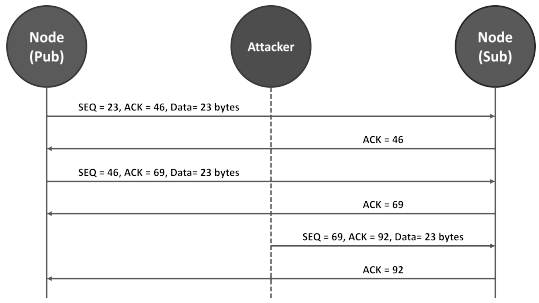

Fig. 6. Synchronization of the sequence number in TCP injection attack

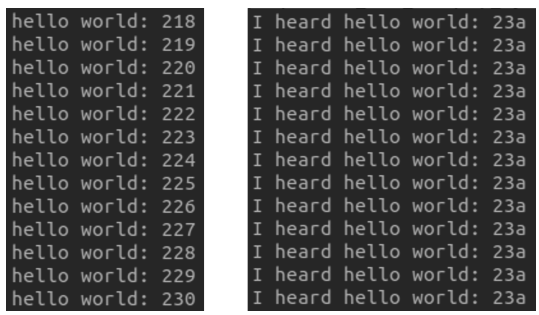

Fig. 7. Results of TCP injection attacks ((left) before the attack / (right) after the attack)

을 생성하여 전송한다면, Subscriber Node는 해당 패킷을 정상적으로 수신하게 된다. 또한, 기존 Publisher Node가 전송하는 패킷은 SEQ가 일치하지 않아 수신하지 못하게 된다. 그림 7을 보는 바와 같이, TCP Injection 공격을 실제로 실험한 결과, Subscriber Node는 TCP Injection 공격이 수행된 시점 이후로 Publisher Node가 전송하는 "hello world: N" 메시지가 아닌 "hello world:

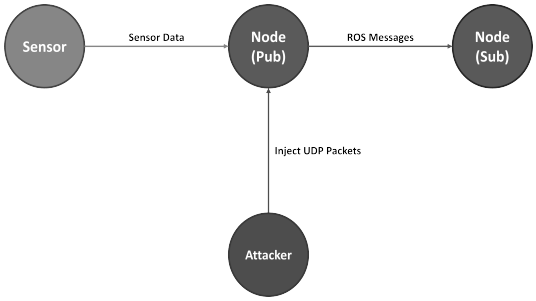

Fig. 8. UDP injection attack without a fake node

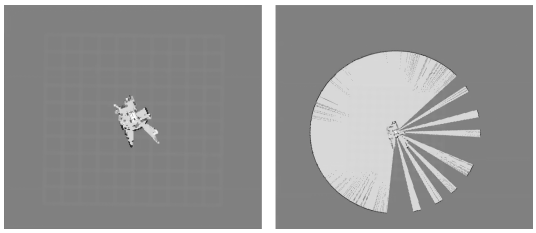

Fig. 9. Results of UDP injection attacks ((left) before the attack / (right) after the attack)

23a" 메시지만 수신하는 것을 확인하였다. 이를 통해, TCP Injection 공격이 성공적으로 수행된 것을 볼 수 있다.

Without a Fake Node 공격의 두 번째 방법인 UDP Injection 공격은 그림 8과 같이 UDP 패킷으로 전송되는 Sensor Data를 공격자가 모방하여 Publisher Node에게 패킷을 전송하는 방법이다. 해당 공격을 성공적으로 수행하기 위해서는 공격 대상인 Sensor Data의 구조를 정확하게 파악해야 한다. 이를 위한 방법으로, Sensor 값이 전송되는 패킷을 모니터링하고 분석함으로써 구조를 파악하거나 Sensor에 대해 자세히 설명되어 있는 문서 또는 자료를 확인함으로써 구조를 파악할 수 있다. 본 연구에서는 LiDAR 패킷을 대상으로 UDP Injection 공격을 수행하였으며, LiDAR 센서에 대한 문서[23]를 확인함으로써 패킷의 구조를 파악하였다. 그림 9를 보는 바와 같이 UDP Injection 공격을 실험한 결과, 주입한 LiDAR 패킷의 Channel Data의 값대로 Visualization 되는 것을 볼 수 있다.

4.2 데이터셋 생성 및 수집 환경

ROS 공격 데이터셋 생성 과정 및 수집 환경은 그림 10과 같다. 실제 로봇 시스템으로부터 데이터

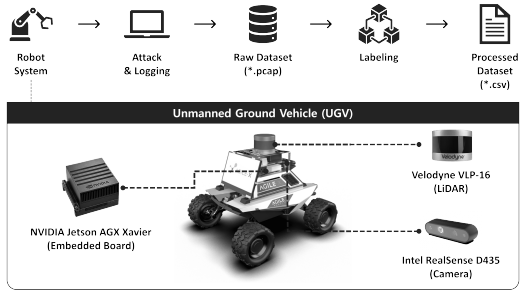

Fig. 10. Overview of dataset generation and experimental setup

를 수집하기 위해, 무인지상차량 (WeGo ST MINI with R&D Kit Pro)을 사용하였으며, 해당 무인지상차량에는 크게 3개의 제품이 탑재되어 있다. 첫 번째는 임베디드 보드 (NVIDIA Jetson AGX Xavier)가 탑재되어 있으며, Ubuntu 18.04 운영체제와 ROS Melodic 버전이 설치되어 있다. 두 번째는 LiDAR 센서 (Velodyne VLP-16)가 탑재되어 있으며, 임베디드 보드와 Ethernet으로 연결되어 있고 UDP 통신을 한다. 마지막으로 Camera 센서 (Intel RealSense D435)가 탑재되어 있으며, 임베디드 보드와 USB로 연결되어 있다.

무인지상차량을 대상으로 4.1절에서 설명한 2가지 공격 시나리오를 이용하여 공격을 수행하고 데이터셋을 생성한다. With a Fake Node 공격의 SNST 공격과 DNST 공격 그리고 Without a Fake Node 공격의 TCP Injection 공격의 경우, 무인지상차량을 움직이게 하는 패키지인 geometry_msgs의 Twist 메시지를 Publish 하고 있는 Node와 해당 메시지를 Subscribe 하는 Node가 존재하는 상황에서 공격을 수행하였다. UDP Injection 공격의 경우, LiDAR 센서를 대상으로 공격을 수행하였다. 해당 공격 과정을 수집하기 위해, 무인지상차량의 임베디드 보드에서 Wireshark 또는 Tcpdump와 같은 네트워크 트래픽 수집 도구를 사용하였다. 참고로, ROS 라이브러리에서 ROS 메시지를 저장할 수 있는 Rosbag 패키지를 제공하고 있지만, 해당 패키지는 Publish 된 Topic에 대한 정보만을 기록한다. 이러한 경우, Without a Fake Node 공격의 TCP Injection 공격과 UDP Injection 공격에 대해서는 데이터를 수집할 수 없기 때문에 적합하지 않아, 본 논문에서는 Wireshark 도구를 사용하여 패킷을 수집함으로써 데이터셋을 구축하였다.

생성된 데이터셋을 활용하기 위해서는 어떤 데이

터가 정상이고 공격인지 라벨을 부여하는 과정이 중요하므로, Wireshark 도구를 통해 수집된 Raw 데이터셋 (*.pcap)에서 With a Fake Node 공격과 Without a Fake Node 공격에 해당하는 패킷을 “Attack”으로 라벨링하는 과정을 진행한다. 최종적으로, 각 패킷의 계층 별로 필드 (e.g., Source IP, Source Port 등)를 추출하고 “Normal”과 “Attack”으로 라벨링한 데이터셋 (*.csv)을 제공하며, 라벨링 방법에 대한 상세한 설명은 아래와 같다.

4.3 데이터셋 라벨링

4.3.1 With a Fake Node 공격에 대한 라벨링

With a Fake Node 공격의 경우, 기존 ROS 환경에서 존재하지 않았던 새로운 Node를 생성하여 악의적인 메시지를 주입하는 공격이므로, 기존에 실행되고 있던 Node와 Topic에 대한 Port 그리고 공격이 실행된 이후의 Node와 Topic에 대한 Port를 서로 비교함으로써 새롭게 생성된 Port로 전송된 패킷을 공격 패킷으로 라벨을 부여한다. Node와 Topic에 대한 Port 정보는 ROS Master에게 요청함으로써 확인할 수 있다.

4.3.2 Without a Fake Node 공격에 대한 라벨링

Without a Fake Node 공격은 TCP 또는 UDP 패킷을 생성할 때 헤더나 페이로드에 Magic Number와 같은 식별할 수 있는 정보를 삽입함으로써 라벨링 과정을 진행한다. TCP Injection 공격의 경우, TCP 헤더의 Reserved 필드 값을 다른 값으로 변경함으로써 라벨링을 진행한다. Reserved 필드는 차후의 사용을 위해 남겨둔 예비 필드로 3bits 크기이며 0으로 채워져 있다. 공격을 수행할 때는 해당 필드 값을 1에서 2^3 사이의 값으로 변경한다. UDP Injection 공격의 경우, UDP 패킷의 페이로드에 특정한 패턴을 삽입함으로써 라벨링을 진행한다. 예를 들어, LiDAR 센서에 대해 공격을 수

Table 1. The details of our proposed datasets

Attack Type	Description	# of Normal	# of Attack	# of Total
Benign	This scenario contains network traffic captured for about 6 minutes without any attacks.	1,009,106	0	1,009,106
Denial of Service (DoS)	This scenario contains network traffic captured for about 6 minutes with SYN flood attacks[24]. The attack was performed 3 times for 30 seconds each.	941,828	11,345,569	12,287,397
SNST of With a Fake Node	This scenario contains network traffic captured for about 6 minutes with SNST attacks. The attack was performed 1 time for 3 minutes.	979,872	23,414	1,003,286
DNST of With a Fake Node	This scenario contains network traffic captured for about 6 minutes with DNST attacks. The attack was performed 1 time for 3 minutes.	980,393	22,732	1,003,125
Unauthorized Subscribe	This scenario contains network traffic captured for about 6 minutes with unauthorized subscribe attacks that employ one fake node to subscribe to the topic. The attack was performed 1 time for 3 minutes.	990,096	12,950	1,003,046
Subscribing Flood	This scenario contains network traffic captured for about 6 minutes with subscribing flood attacks that employ 100 fake nodes to subscribe to the topic. The attack was performed 1 times for 1 minute.	2,664,940	543,481	3,208,421
TCP Injection of Without a Fake Node	This scenario contains network traffic captured for about 6 minutes with TCP Injection attacks. The attack was performed 3 times for 1 minute each.	1,028,542	183	1,028,725
UDP Injection of Without a Fake Node	This scenario contains network traffic captured for about 6 minutes with UDP Injection attacks. The attack was performed 3 times for 30 seconds each.	934,884	870,576	1,805,460

Table 2. Comparison between ROSIDS23 and our proposed datasets

	Real Robot System	Attack Type								Labeling
		Benign	DoS	With a Fake Node		Unauthorized Subscribe	Subscribing Flood	Without a Fake Node		
				SNST	DNST			TCP Injection	UDP Injection	
ROSIDS23[15]	O	O	O	X	O	O	O	X	X	O
Our Dataset	O	O	O	O	O	O	O	O	O	O

행한다면, 그림 9와 같이 LiDAR 패킷의 Channel Data 값을 모두 하나의 값으로 지정하거나 페이로드의 마지막 부분에 Magic Number를 삽입한다.

4.4 데이터셋 구성

본 논문에서 생성한 ROS 공격 데이터셋의 구성은 표 1과 같으며, 총 8가지의 시나리오가 포함되어 있다. Benign 시나리오는 공격이 포함되지 않은 정상 데이터셋을 의미하고, Benign 시나리오를 제외한 7가지는 공격이 포함된 데이터셋이다. 그중, 4가지의 공격 시나리오 (i.e., DoS, With a Fake Node (DNST), Unauthorized Subscribe, Subscribing Flood)는 ROSIDS23[15]에서 제안한 공격으로, 본 논문에서는 해당 4가지 공격 시나리오에 대해 ROSIDS23[15]에서 설명한 것을 기반으로 공격을 수행하고 데이터셋을 구축하였다. 참고로, ROSIDS23[15]은 데이터셋만 제공하고 소스코드는 제공하고 있지 않다. ROSIDS23[15] 데이터셋과 비교하였을 때, 표 2를 보는 바와 같이, 본 논문에서 구축한 데이터셋에는 3가지의 공격 시나리오 (i.e., With a Fake Node (SNST), Without a Fake Node (TCP Injection, UDP Injection))가 더 포함된 것을 볼 수 있다. 이로써, ROSIDS23 데이터셋 보다 더 다양한 공격 시나리오가 포함된 데이터셋을 제공하였다.

본 논문에서도 ROS에 대한 침입 탐지 시스템 개발 및 연구에 도움을 제공할 수 있도록 구축한 ROS 공격 데이터셋을 오픈 데이터셋으로 공개하며, 각 데이터셋마다 Raw 데이터 (*.pcap)와 가공된 데이터 (*.csv) 모두를 제공한다.

V. 결 론

ROS는 Robotics 분야의 산업용 통신 미들웨어 프로토콜 중에서 가장 널리 사용되고 있지만, 보안 메커니즘이 적용되어 있지 않다는 취약점이 존재한다. 이에 대응하기 위해 많은 연구가 진행되었으며, 특히 ROS 환경에서 공격을 수행하고 데이터셋을 구축하여 제공하는 연구가 진행되었다. 그러나 이러한 분야의 연구는 매우 부족한 상황이다. 이를 위해, 본 논문에서는 ROS 환경에서 발생할 수 있는 새로운 공격 시나리오를 제안하였으며, 제안한 공격 시나리오를 기반으로 실제 로봇 시스템 환경에서 공격을 수행하고 데이터셋을 구축하였다. 본 논문에서는 ROS를 사용하는 Robotics 분야의 보안 연구 발전에 기여할 수 있도록 구축한 공격 데이터셋을 오픈 데이터셋으로 제공하였다.

추가적으로, 제안한 공격 시나리오 모두 임베디드 보드 내부에서만 가능한 것이 아닌 임베디드 보드 간의 외부 통신에서도 공격이 가능한 것을 실험적으로 확인하였으며, 이는 공격 범위가 시스템 내부에 국한되지 않고 네트워크를 통한 외부 침입 또한 심각한 위협이 될 수 있음을 시사한다. 향후 연구로는 제안한 ROS 공격 데이터셋을 활용하여 침입 탐지 시스템에 대해 연구를 진행할 예정이며, ROS 이외에도 ROS2에 대해서 발생 가능한 새로운 공격 시나리오를 제안하고 공격 데이터셋을 구축할 예정이다.

References

[1] E. Garcia, M. A. Jimenez, P. G. De Santos, and M. Armada, "The evolution of robotics research," IEEE Robotics & Automation Magazine, vol. 14, no. 1, pp. 90-103, Mar. 2007.

[2] MQTT, "MQTT", https://mqtt.org/, 2024.06.07.
[3] M. Quigley, B. Gerkey, K. Conley, J. Faust T. Foote, J. Leibs, E. Berger, and R. Wheeler, "ROS: an open-source Robot Operating System," ICRA workshop on open source software, vol. 3, no. 3.2, p. 5, Jan. 2009.
[4] ZeroMQ, "ZeroMQ", https://zeromq.org/, 2024.06.07.
[5] S. Profanter, A. Tekat, K. Dorofeev, M. Rickert, and A. Knoll, "OPC UA versus ROS, DDS, and MQTT: Performance evaluation of industry 4.0 protocols," IEEE International Conference on Industrial Technology, pp. 955-962, Feb. 2019.
[6] E. Tsardoulias and P. Mitkas, "Robotic frameworks, architectures and middleware comparison," arXiv preprint arXiv:1711.06842, Nov. 2017.
[7] P. Estefo, J. Simmonds, R. Robbes, and J. Fabry, "The robot operating system: Package reuse and community dynamics," Journal of Systems and Software, vol. 151, pp. 226-242, Feb. 2019.
[8] J. McClean, C. Stull, C. Farrar, and D. Mascarenas, "A preliminary cyber-physical security assessment of the robot operating system (ros)," Proceedings of the SPIE - The International Society for Optical Engineering, vol. 8741, pp. 341-348, May. 2013.
[9] B. Dieber, B. Breiling, S. Taurer, S. Kacianka, S. Rass, and P. Schartner, "Security for the robot operating system," Robotics and Autonomous Systems, vol. 98, pp. 192-203, Oct. 2017.
[10] B. Dieber, R. White, S. Taurer, B. Breiling, G. Caiazza, H. Christensen, and A. Cortesi, "Penetration testing ROS," Robot Operating System (ROS) The Complete Reference (Volume 4), pp. 183-225, Jun. 2020.
[11] R. White, D. H. I. Christensen, and D. M. Quigley, "SROS: Securing ROS over the wire, in the graph, and through the kernel," arXiv preprint arXiv:1611.07060, Nov. 2016.
[12] N. Goerke, D. Timmermann, and I. Baumgart, "Who controls your robot? an evaluation of ros security mechanisms," In 2021 7th International conference on automation, robotics and applications (ICARA), pp. 60-66, Feb. 2021.
[13] S. Lagraa, M. Cailac, S. Rivera, F. Beck, and R. State, "Real-time attack detection on robot cameras: A self-driving car application," In 2019 Third IEEE International Conference on Robotic Computing (IRC), pp. 102-109, Feb. 2019.
[14] R. A. Antunes, B. L. Dalmazo, and P. L. J. Drews, "Detecting data injection attacks in ROS systems using machine learning," In 2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE), pp. 1-6, Oct. 2022.
[15] E. Değirmenci, Y. S. Kırca, İ. Özçelik, and A. Yazıcı, "ROSIDS23: Network intrusion detection dataset for robot operating system," Data in Brief, vol. 51, p. 109739, Nov. 2023.
[16] I. F. Kilincer, F. Ertam, and A. Sengur, "Machine learning methods for cyber security intrusion detection: Datasets and comparative study," Computer Networks, vol. 188, Feb. 2021.
[17] S. Rajapaksha, G. Madzudzo, H.

Kalutarage, A. Petrovski, and M. O. Al-Kadri, “CAN-MIRGU: A comprehensive CAN bus attack dataset from moving vehicles for intrusion detection system evaluation,” In Network and Distributed Systems Security (NDSS) Symposium, Feb. 2024.

〔18〕 GitHub, “ROS2”, https://github.com/ros2, 2024.06.07.

〔19〕 ROS Metrics, “ROS Metrics”, https://metrics.ros.org/rosdistro_rosdistro.html, 2024.06.07.

〔20〕 V. Mayoral-Vilches, M. Pinzger, S. Rass, B. Dieber, and E. Gil-Uriarte, “Can ros be used securely in industry? red teaming ros-industrial,” arXiv preprint arXiv:2009.08211, Sep. 2020.

〔21〕 GitHub, “CICFlowMeter”, https://github.com/ahlashkari/CICFlowMeter, 2024.06.07

〔22〕 ROS Wiki, “ROS Shutdown”, https://wiki.ros.org/rospy/Overview/Initialization%20and%20Shutdown, 2024.06.07.

〔23〕 Ouster, “VLP-16 User Manual”, https://velodynelidar.com/wp-content/uploads/2019/12/63-9243-Rev-E-VLP-16-User-Manual.pdf, 2024.06.07.

〔24〕 D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, “Inferring internet denial-of-service activity,” vol. 24, no. 2, pp. 115-139, ACM Transactions on Computer Systems (TOCS), May. 2006.

〈저자소개〉

김 형 훈 (Hyunghoon Kim) 학생회원
2019년 8월: 한림대학교 컴퓨터공학과 졸업
2021년 8월: 숭실대학교 일반대학원 석사 졸업
2024년 3월~현재: 연세대학교 정보대학원 박사과정
〈관심분야〉 자동차 보안, IoT/CPS 보안, 네트워크 보안

이 승 민 (Seungmin Lee) 학생회원
2024년 2월: 숭실대학교 소프트웨어학부 졸업
2024년 3월~현재: 연세대학교 정보대학원 석사과정
〈관심분야〉 자동차 보안, 시스템 보안

허 재 웅 (Jaewoong Heo) 학생회원
2021년 2월: 한림대학교 컴퓨터공학과 졸업
2023년 8월: 숭실대학교 일반대학원 석사 졸업
〈관심분야〉 자동차 보안, 네트워크 보안

조 효 진 (Hyo Jin Jo) 종신회원
2009년 2월: 고려대학교 산업공학과 졸업
2016년 2월: 고려대학교 정보보호대학원 박사 졸업
2018년 8월: University of Pennsylvania 박사 후 연구원
2020년 8월: 한림대학교 정보과학대학 조교수
2024년 2월: 숭실대학교 소프트웨어학부 조교수
2024년 3월~현재: 연세대학교 정보대학원 부교수
〈관심분야〉 자동차 보안, IoT/CPS 보안, 프라이버시