UAV에서의 로봇 운영 시스템에 대한 사이버 공격과 보안 플랫폼을 이용한 방어 실증 연구

An Empirical Study on Cyber Attack and Defense of Robot Operating System using Security Platform in UAV

저자 (Authors)

윤지영, 이효준, 박경준

Jiyoung Yoon, Hyojun Lee, Kyung-Joon Park

출처 (Source)

제어로봇시스템학회 논문지 27(2), 2021.2, 111-117 (7 pages)

Journal of Institute of Control, Robotics and Systems 27(2), 2021.2, 111-117 (7 pages)

발행처 (Publisher)

제어로봇시스템학회

Institute of Control, Robotics and Systems

URL

http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE10525849

APA Style

윤지영, 이효준, 박경준 (2021). UAV에서의 로봇 운영 시스템에 대한 사이버 공격과 보안 플랫폼을 이용한 방어 실증 연구. 제어로봇시스템학회 논문지, 27(2), 111-117.

Journal of Institute of Control, Robotics and Systems (2021) 27(2):111-117
https://doi.org/10.5302/J.ICROS.2021.20.0189 ISSN:1976-5622 eISSN:2233-4335

UAV에서의 로봇 운영 시스템에 대한 사이버 공격과 보안 플랫폼을 이용한 방어 실증 연구

An Empirical Study on Cyber Attack and Defense of Robot Operating System using Security Platform in UAV

윤 지 영, 이 효 준, 박 경 준*
(Jiyoung Yoon[1], Hyojun Lee[2], and Kyung-Joon Park[2,*])
[1]INTUSEER Inc.
[2]Department of Information and Communication Engineering, Daegu Gyeongbuk Institute of Science and Technology

Abstract: Recently, UAVs (Unmanned Aerial Vehicles), which represent a key application of cyber -physical systems, have been used for malicious intent. Accordingly, research on the neutralization of UAV has become critical. The UAV neutralization process can be divided into three phases. The identification stage is aimed at distinguishing whether the UAVs are friendly or adversarial. Next, the actual mission of the UAVs is neutralized. Subsequently, post-processing is performed to guide the UAV to safe areas. Because the existing UAV neutralization studies did not consider phase 3, secondary damage such as that to people and property may occur after an attack. Therefore, we propose a UAV neutralization method in which phase 3 applied. In this paper, to implement functions such as autonomous driving, navigation, and collision avoidance, we investigate the vulnerability of the MAVROS environment commonly used in unmanned vehicles and clarify the method of attack that exploits the vulnerability. Additionally, we propose a MAVROS API to enhance the security to prevent these vulnerabilities. Finally, we verify the proposed attacks and security through empirical research by considering UAVs with MAVROS.

Keywords: unmanned aerial vehicles, cyber-physical systems, network attack, security

I. 서론

드론이라 불리는 무인항공기(UAV, Unmanned Aerial Vehicle)는 레저, 미디어 및 엔터테인먼트, 군사 등의 목적과 함께 건설, 광산업, 생명 구조 활동 등 활용 범위가 점차 넓어지고 있다. 국토부에 따르면 세계 드론 시장은 연 29%씩 성장하여 2026년 약 820억 달러에 이를 것으로 예상되고 있다. 드론은 4차 산업혁명의 테마인 CPS (Cyber-Physical Systems)의 중요한 어플리케이션 중 하나로 대두되고 있다[1,2].

2019년 9월 사우디아라비아의 정유시설과 원유 생산 기지에 대한 드론 공격 사건뿐만 아니라 자폭 드론 공격을 통한 남예멘 군 사살 사건 등 UAV를 테러 수단으로 삼은 공격이 빈번해지고 있다. 또한 군사 목적으로 쓰이는 UAV는 임무에 대한 정보 보안 역시 중요하다. 그러나 UAV 네트워크에 대한 취약성은 여러 연구를 통해 실험으로 증명되었고 이러한 취약점을 보완하기 위해 여러 방어 방법이 연구되고 있다[3-7].

UAV 무력화 단계는 크게 세 단계로 나눌 수 있다. 첫 번째 단계는 UAV를 공격 대상 또는 아군으로 식별하는 것이다. 다만 핵 시설, 군사기지, 인구밀집지역 등 시설 보호 목적으로 드론을 운용할 수 없는 No-Drone Zone에서는 이 단계를 생략할 수 있다. 두 번째 단계는 UAV를 실제로 무력화하는 것이다. 무인 항공기는 교란, 프로토콜 취약성을 통한 네트워크 공격, 스푸핑 등의 방법을 통해 연구되고 있다[8-12]. UAV 무력화 이후의 단계인 세 번째 단계는 사후처리 단계다. 무인기 무력화로 인해 추락이나 호버링하는 무인기를 2차 피해 없이 안전하게 처리하는 단계를 의미한다. 높은 고도에서 비행하는 무인기가 3단계 과정 없이 무력화되면 인명피해나 재산피해, 핵 시설 추락 등 심각한 2차 피해가 발생할 수 있다. 1단계와 2단계는 여러 연구가 수행되었으나, 3단계는 아직 연구가 잘 되지 않았다. 본 연구는 3단계가 고려된 UAV 무력화를 고려한다.

본 논문에서는 무인이동체에서 보편적으로 사용되는 MAVROS 환경의 취약점을 분석하고 이를 이용한 공격방법을 제안한다. 해당 공격의 방어를 위한 보안용 API 또한 제안한다. 마지막으로, 우리는 UAV를 이용한 실증 연구를 통해 MAVROS의 취약성과 제안된 API의 보안을 검증한다.

*Corresponding Author
Manuscript received November 20, 2020; revised December 11, 2020; accepted December 29, 2020
윤지영: (주)인투시어 프로젝트 매니저(intuseer.jyyoon@gmail.com, ORCID 0000-0003-2504-8476)
이효준: 대구경북과학기술원 정보통신융합전공 대학원생(hj.lee@dgist.ac.kr, ORCID 0000-0002-7611-4296)
박경준: 대구경북과학기술원 정보통신융합전공 교수(kjp@dgist.ac.kr, ORCID 0000-0003-4807-6461)
※ This work was partly supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government (MSIP)(2014-0-00065, Resilient Cyber-Physical Systems Research)

II. 배경

1. Unmanned Aerial System

UAS (Unmanned Aerial System)는 무인항공기와 통신장비, 지상국, 통신 등을 포함한 모든 시스템을 통칭한다. UAV는 RC transmitter, Bluetooth, Wi-Fi 등 다양한 매체를 이용하여 제어할 수 있으며, UAV는 해당매체를 통해 GCS (Ground Control Station)와 연결되어 비행한다. 우리는 Wi-Fi를 통한 UAV와 GCS의 네트워크 환경을 고려하여 UAS를 구성하였다. GCS와 UAV는 Wi-Fi에 연결되어있고 MAVLink 프로토콜을 이용한다[13]. MAVLink 프로토콜은 UAV와 GCS 뿐만 아니라 UAV에 탑재된 여러 구성 요소들 사이에서도 통신하기 위해 사용되며 DJI, Parrot 사의 제품뿐만 아니라 많은 UAV에서 사용하는 대표적인 프로토콜이다. MAVLink 프로토콜은 헤더중심 프로토콜로 한정된 자원의 UAV가 가볍게 사용하기에 알맞은 프로토콜이다. 이렇게 연결된 GCS는 UAV에게 비행 임무와 관련된 명령을 내리고 이에 대한 UAV의 비행 상태 및 여러 센서들의 상태 정보를 모니터링 할 수 있다. 그림 1은 MAVLink protocol의 메시지를 나타낸다. MAVLink는 MAVLink 1.0과 MAVLink 2.0으로 크게 두 버전이 있다. MAVLink 2.0은 아직 지원하지 않는 장치들이 많이 있으나 하위 호환이 가능한 점이 있다. 그림 2는 QGroundControl로 UAS 구성 중 GCS를 나타낸다. 그림 3은 본 논문에서 사용하는 UAV를 나타낸다.

2. Robot Operating System

ROS (Robot Operating System)은 로봇을 위한 오픈소스 메타 운영체제로 하드웨어 추상화를 비롯한 로봇에서 주로 쓰이는 기능들이 구현되어 있으며 프로세스간 메시지 전달, 파일 관리 등을 제공한다. ROS는 간략하게 그림 4와 같이 표현할 수 있다. ROS는 크게 ROS master와 publisher, subscriber 노드로 구성되어 있다. ROS master는 초기에 노드들을 연결하기 위해 연결을 원하는 노드들로부터 연결과 관련된 정보들을 받는다. 서로의 IP 주소를 ROS master로부터 주고 받은 노드들은 후에는 ROS master 없이 직접 연결을 하게 된다. 이 때 데이터를 생산하고 제공하는 노드는 publisher 노드라고 하며 정해진 토픽에 데이터를 발행한다. Subscriber 노드는 원하는 토픽으로부터 publisher 노드가 발행한 데이터를 subscribing 할 수 있다. 최근에는 ROS2가 정식으로 출시 되었으며, 이전 버전과 가장 큰 차이점이자 특징은 미들웨어로 DDS (Data Distribution Service)를 채택하였다는 것이다. DDS에는 몇 가지 보안 사항이 요구되는데, ROS2에서는 그 중에 authentication, access control, cryptographic이 구현되어 있다[14]. 이를 통해 미션 크리티컬한 ROS 환경을 위해 보안이 중요하다는 사실을 알 수 있다. 하지만 ROS와 ROS2는 native한 환경에서 서로 호환되지 않기 때문에, 기존 ROS를 사용하는 시스템에는 여전히 보안 이슈가 남아 있다.

3. Network Attack on Cyber-Physical Systems

CPS는 보통 센서, 액추에이터와 같이 물리 시스템을 네트워크를 통해 사용자가 제어하는 시스템을 말하며 계산, 통신 및 제어 기술을 통합하여 프로세스를 모니터링하고 제어한다. 이때 물리 계층과 어플리케이션 계층을 연결하는 통신 네트워크가 취약점이 될 수 있으며 이는 CPS를 위협한다. 데이터 전송 계층에 대한 네트워크 공격으로는 대표적으로 spoofing attack, sniffing attack, DoS attack이 있다.

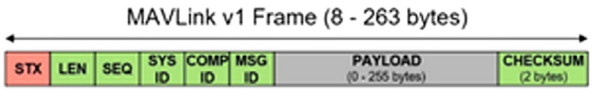

그림 1. MAVLink protocol 메시지[13].
Fig. 1. MAVLink protocol message [13].

그림 2. 지상관제소.
Fig. 2. Ground Control Station.

그림 3. 실험에 사용된 무인기.
Fig. 3. UAV used in an experiment.

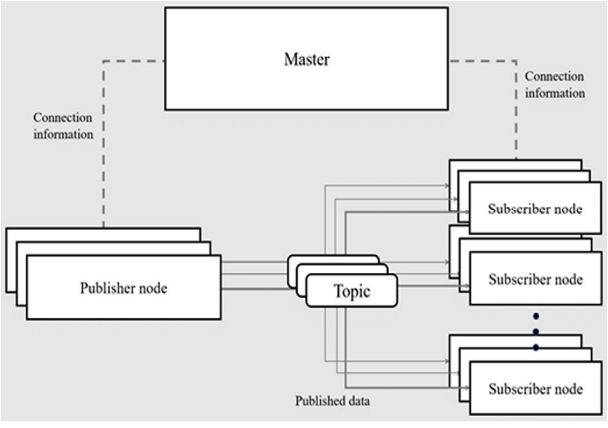

그림 4. ROS 구조.
Fig. 4. Robot Operating System structure.

III. UAV 모델 및 취약점

이 절에서는 자율비행, 군집비행 등의 특수 목적을 위해 ROS가 적용된 UAV의 모델과 그 환경에서 발생할 수 있는 취약점을 설명한다.

1. MAVROS가 적용된 UAV 모델

UAV의 자율비행, 군집비행 등 첨단 기술 운용을 위해서는 UAV 자세와 위치를 제어하는 flight controller 뿐만 아니라 추가의 컴퓨팅 기능이 가능한 하드웨어 및 소프트웨어가 필요하다. 이렇게 flight controller 외부의 정보와 컴퓨팅 파워를 이용하여 비행에 부가적 도움을 주는 기능을 오프보드라고 한다. 본 논문에서 가정하는 UAS는 flight controller에 오프보드 컴퓨터가 연결되어 서로 통신하는 구조를 갖는다. 또한 오프보드 컴퓨터는 ROS를 통해 flight controller, 외부센서, 오프보드 컴퓨터 간 통신을 지원한다.

ROS는 로봇에 필요한 다양한 기능들을 제공하며 큰 생태계를 가지고 있다. 따라서 다양한 로봇과 센서, 어플리케이션 등에 적용할 수 있고 이용되고 있다. 특히 무인 이동체에서는 localization, navigation, mapping, vision 등 자율 주행을 위한 기능들을 개발하기 위해 ROS를 많이 사용하고 있다[15, 16]. MAVROS는 ROS를 사용하는 무인이동체에서 MAVLink 프로토콜을 사용하기 위해 제안된 ROS의 확장 패키지 중 하나이다. MAVROS를 이용하면 ROS를 사용하는 오프보드 컴퓨터와 MAVLink를 사용하는 다양한 무인이동체와도 통신이 가능하다. 본 논문에서는 PX4 flight stack과 ROS를 사용하는 오프보드 컴퓨터 사이에 통신을 위해 MAVROS를 적용하였다.

그림 5는 위의 절차를 통해 MAVROS가 적용된 UAV의 시스템 모델을 나타낸다. 외부에 존재하는 External sensor와 External computer에 /Sensor, /Computer 노드가 특정 토픽을 매개로 Offboard computer 내의 /Process 노드로 message를 publish 한다. /Process 노드는 센서와 명령 데이터를 바탕으로 UAV 제어를 위한 명령 message를 /MAVROS 노드에 전달한다. /MAVROS는 해당 데이터를 MAVLink로 flight controller에 전달한다. Flight controller는 Pixhawk을 의미하며 전달받은 데이터를 바탕으로 비행에 필요한 자세 및 위치를 제어한다.

2. 취약점 분석

ROS에는 네트워크 공격에 대한 취약점이 있다고 연구되어 왔다. ROS는 publisher 노드와 subscriber 노드로 나뉠 수 있는데, 이때 신원확인이나 허가에 대한 기능이 없으므로 패킷 인젝션 공격을 통해 허가되지 않은 publishing 혹은 허가되지 않은 subscribing을 할 수 있다. 이는 정보 보안의 3요소라고 알려진 기밀성, 무결성, 가용성을 모두 위배할 수 있다. 공격자는 이러한 취약점을 이용하여 허가되지 않은 publishing을 통해 허위 데이터를 로봇에 주입할 수 있으며 이는 로봇의 행위에 치명적인 위험을 초래할 수 있다. 또한 허가되지 않은 subscribing을 통해 로봇의 상태를 모니터링 할 수 있으며 로봇의 정보 처리와 관련되거나 어떠한 행동을 요구하는 가짜 데이터를 계속해서 publishing하여 DoS 공격을 할 수 있다.

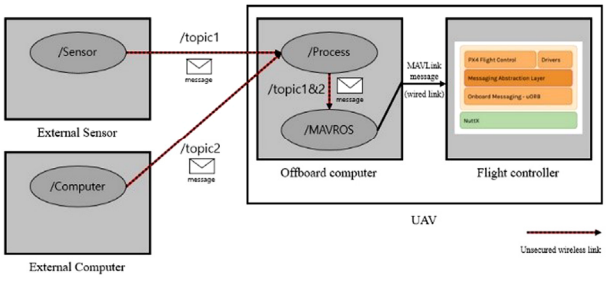

그림 5. MAVROS가 적용된 UAV 모델.

Fig. 5. The model of UAV with MAVROS.

IV. 공격 및 방어 방법

이 절에서는 ROS의 취약점 분석을 토대로 MAVROS가 적용된 UAV에 어떠한 종류의 공격이 가해질 수 있는지 보인다. 또한 해당 공격을 방어하는 입장에서는 어떠한 접근법으로 방어할 수 있는지 설명한다.

1. 제안된 공격 방법

MAVROS 또한 ROS의 확장 패키지로써 ROS의 취약점을 물려받는다. 특히 이러한 네트워크 공격에 대한 ROS의 취약점은 UAV와 같이 제어에 민감한 무인이동체에 대해 더욱 위협적이며 즉각적인 2차 피해를 초래할 수 있다.

MAVROS에는 무인이동체의 주행을 위한 많은 수의 토픽들이 존재한다. MAVROS가 적용된 UAV의 센서와 액추에이터들은 이 토픽들을 통해 데이터를 주고 받는다. 본 논문에서는 이 토픽들 중 setpoint_position 토픽을 이용하여 UAV를 무력화할 수 있다는 것을 발견하였다. 이는 CIA 중 무결성을 위배하는 공격이다. 이 취약점을 이용하여 우리는 setpoint_position 토픽에 허가되지 않은 데이터 인젝션을 통한 공격을 한다. 해당 취약점을 이용한 공격을 수행하기 위해서는 ROS 통신 절차, 포트 스캔, ROS 소스 분석에 대한 이론적, 기술적 이해가 필요하다.

그림 6은 공격 상황을 간략히 나타낸다. MAVROS와 Pixhawk 보드를 이용하는 UAV와 GCS가 Wi-Fi를 통해 연결된 상황에서 공격이 가해진다. UAV 내부에서는 공격자가 UAV 오프보드 컴퓨터의 MAVROS 에 침입하여 ROS master를 통해 공격자의 publisher 노드를 생성 및 연결하게 된다. 이후 setpoint_position 토픽에 악의적인 값을 포함한

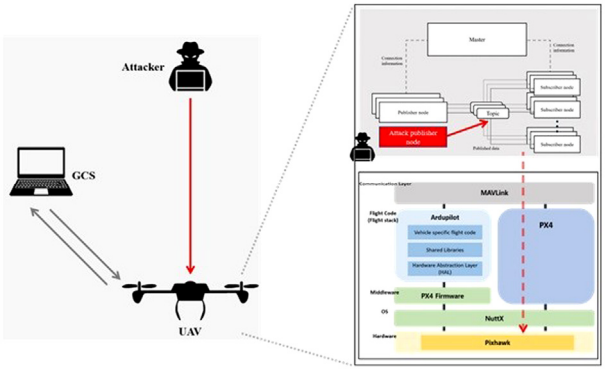

그림 6. 제안된 공격 방법.

Fig. 6. Proposed attack method.

데이터를 publishing한다. 해당 토픽에 대한 데이터를 수신하는 subscriber 노드는 이를 걸러내지 못하므로 악의적인 데이터의 영향은 Pixhawk board까지 미치고 UAV는 공격자의 데이터를 이용하여 UAV를 제어하게 된다. 현재 ROS는 접근제어, 인증 등 ROS에 접근하는 노드에 대한 식별이 없기 때문에 ROS master의 IP 주소만 알아낸다면 공격자는 손쉽게 UAV의 제어 권한을 장악할 수 있다. 이러한 공격 방법을 이용하여 조금 더 정교하게 조작한다면 UAV를 원하는 곳으로 납치하는 것도 가능하다.

본 논문에서 제안한 공격 방법은 무인기 무력화 단계 중 첫 번째 단계인 피아식별 단계를 행한 후의 두 번째 단계와 세 번째 단계에 해당하는 무인기 무력화 방법이다. 특히 우리가 제안한 공격 방법은 UAV의 현재 고도, 착륙 지점과 상관없이 안전하게 바닥에 착륙하도록 유도할 수 있다. 이것은 무인기 무력화 단계 중 세 번째 단계를 만족한다. 이 공격 방법을 사용한다면 무인기 무력화로 발생할 수 있는 재산피해, 인명피해를 예방할 수 있고 허가되지 않은 무인기를 안전하게 그대로 회수할 수 있다는 장점이 있다. 해당 공격은 V 섹션에서 실제 실험을 통해 보인다.

2. 제안된 공격에 대한 방어 방법

우리는 ROS의 취약점으로 UAV를 착륙시킬 수 있다는 것을 확인하였다. 이러한 MAVROS의 취약점을 막기 위해 본 논문에서는 무인 이동체를 위한 보안 API를 제안한다. 보안 API는 ROS의 프로세스 처리, 데이터 처리 등의 고유 업무를 수정하거나 삭제하지 않고 ROS 위에 올라가는 API이다. 이는 기존의 ROS를 사용하는 어플리케이션에서 우리가 제안하는 보안 API를 올리더라도 로봇의 행동에는 영향을 끼치지 않음을 의미한다.

본 논문에서 제안하고자 하는 보안 API는 허가되지 않은 노드의 등록을 제한하는 방법이다. 즉, 무인 이동체가 센서 및 액추에이터 등의 장착을 마친 후 비행이나 주행, 항해 등 임무를 시작하는 상황일 경우 노드 등록을 제한한다. 정상적인 상황에서 센서 및 액추에이터 등의 장착이 없다는 것은 더 이상 추가적인 노드 등록이 없다는 상황을 가정한다. 이 때 시스템 관리자는 MAVROS 어플리케이션에 대한 접근에 잠금을 걸 수 있으며 잠금이 걸린 어플리케이션은 더 이상 노드 등록을 할 수 없다. LOCK 모드와 UNLOCK 모드는 사용자가 동적으로 설정할 수 있으며 사용자가 미리 입력해둔 비밀번호를 이용하여 MAVROS 어플리케이션의 LOCK 모드 상태를 설정할 수 있다.

그림 7은 보안 API가 올라간 경우의 알고리즘을 간략히 나타낸다. 이를 설명하기 위해 한 가지 시나리오를 통해 설명한다. 비행 전, 비행에 필요한 모든 노드를 등록한 관리자는 더 이상의 노드 등록을 제한하기 위해, 초기 상태에서 LOCK을 수행하는 노드를 통해 master에게 LOCK을 요청한다. 그러면 master는 5번째 라인에 해당하는 조건에 해당하여 LOCK을 수행한다. 이후 받는 모든 등록 요청은 3번째 라인의 조건에 해당하여 모든 등록이 거절된다. 만약 비행 이후 새로운 노드를 추가해야 하는 상황이 발생할 때, 관리자가 UNLOCK을 master에 요청하면 1번째 라인 조건에 걸려 LOCK이 풀리게 된다. 이후 들어오는 노드 등록에 대해

Algorithm 1. The defense algorithm

```
1:  if node.name is UNLOCK
2:      release the lock
3:  if node registration is not available
4:      exit the process
5:  if node.name is LOCK
6:      acquire the lock
7:  registration procedure
```

그림 7. 방어 알고리즘.

Fig. 7. The defense algorithm.

서는 모든 조건을 뛰어넘고 7번째 라인을 수행하여 노드를 등록하게 된다. 소개된 시나리오 외에도 해당 방어법을 이용하여 관리자의 UAV 운용 정책에 따라 동적으로 LOCK/UNLOCK 모드 변경을 수행할 수 있다.

V. 실험

1. 실험환경 및 시나리오

본 논문에서는 우리가 제안하는 공격을 시행하기 위해 실험환경을 구성하였다. 그림 8은 우리가 구성한 실험환경을 간략히 나타낸다. UAS는 UAV와 GCS가 Wi-Fi를 통해 연결되어 구성되어 있다. UAV에는 Pixhawk2가 연결되어 있고 이 보드에는 flight stack이 적용되어 있어 비행을 위한 업무를 수행한다. 또한 Pixhawk 보드에는 communication layer로써 MAVLink도 포함되어 동작한다. 또한 UAV에 MAVROS를 장착하기 위해 오프보드 컴퓨터로써 Raspberry Pi 보드가 장착되어 있다. Raspberry Pi에는 MAVROS가 적용되어 있으며 Pixhawk board와 Wi-Fi를 통해 연결되어 있다.

공격 실험을 실행하기에 앞서, 공격자가 없는 정상적인 상황에서의 비행 준비 과정을 살펴본다. MAVROS를 적용시킨 UAV는 오프보드 모드로 비행을 하게 된다. 이는 Raspberry Pi 보드에 내장된 소스코드로 정해진 비행을 하게 되는 것을 의미한다. 정상적인 경우, 먼저 Raspberry Pi 보드에서 ROS CORE를 동작시키고 비행 소스코드를 MAVROS에 연결하기 위해 ROS LAUNCH 명령어를 실행

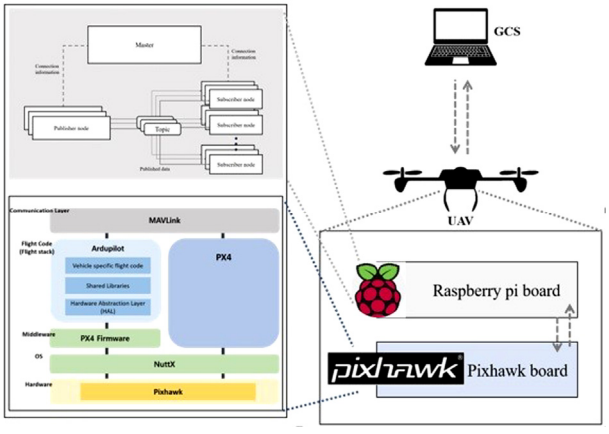

그림 8. 실험 환경.

Fig. 8. The experimental environment.

한다. 그리고 UAV에서는 arm 시킴으로써 비행준비를 마친다. 이렇게 비행준비를 마친 UAV는 ROSRUN 명령어를 실행시키게 되면 내장된 코드인 비행 소스코드대로 비행을 하게 된다.

이제 공격 실험에서의 비행 준비 과정을 살펴본다. 기존 상황에서 공격자를 추가한다. 공격자는 미리 UAV의 네트워크에 침입해있다는 것을 가정한다. 네트워크에 침입한 공격자는 포트 스캔을 통해 먼저 ROS master의 주소를 찾는다. ROS master의 URI를 찾은 공격자는 export ROS_MASTER_URI 명령어를 통해 외부에 있는 ROS master와 연결을 한다. 본 논문에서의 실험 환경에서는 export ROS_MASTER_URI = http://{ROS master IP address}:11311 명령어를 통해 외부에 있는 UAV의 ROS master와 연결한다. 이렇게 UAV의 MAVROS의 ROS master에 연결된 공격자는 UAV의 MAVROS 토픽 중 setpoint_position에 데이터를 주입한다.

우리는 무인기 무력화 단계 중 세 번째 단계를 만족시키기 위해 MAVROS의 setpoint_position 토픽을 이용했다. 엄밀히 말하자면 MAVROS의 setpoint_position 플러그인에서 setpoint_position/local과 setpoint_position/global 두 개를 이용하여 무인기 무력화 세 번째 단계를 만족하는 공격을 하고자 한다. 우리는 공격자가 setpoint_position/local 토픽을 이용하면 UAV를 RTL (Return To Launch) 모드로 바꾸어 시작점으로 돌아가게 할 수 있다는 것을 알아내었다. UAV를 RTL모드로 바꾸게 하는 방법은 다음과 같다. setpoint_position 토픽에는 UAV의 x, y, z좌표를 지정해 줄 수 있으며 첫 이륙지점이 (0, 0, 0) 지점이 된다. 이 토픽에 데이터를 publishing 할 때 (0, 0, 0)을 publishing한다면 UAV는 하던 첫 이륙지점으로 가는 비행을 하게 된다. 즉 공격자가 setpoint_position/local 토픽에 x, y, z값을 (0, 0, 0)으로 설정하여 가짜 데이터를 주입하는 공격을 한다면 UAV는 RTL 모드로 모드를 바꾼 것처럼 이륙지점으로 돌아가게 된다. setpoint_position/global 토픽을 이용하면 UAV의 상대 경로가 아닌 GPS 값을 입력하여 UAV를 제어할 수 있다.

2. UAV 공격에 대한 실험 결과

그림 9는 UAV의 비행 경로를 통해 정상적인 경우, setpoint_position/local, setpoint_position/global 토픽에 공격 값을 주입시킨 경우를 보여준다. UAV의 상대좌표 중 x축은 UAV의 x값을 y축은 UAV의 y값을 나타낸다. 정상적인 경우에는 (0, 0) 시작점에서 이륙하여 약 (170, 260) 지점에서 착륙하는 비행 경로이지만 공격자가 setpoint_position/local 토픽에 공격 값을 주입시킨 경우에는 약 (90, 210) 지점의 공격시점부터 다시 (0, 0) 이륙지점으로 UAV가 되돌아가는 것을 확인할 수 있다. 또한 공격자가 setpoint_position/global 토픽에 공격 값을 주입시킨 경우에는 약 (110, 220) 지점의 공격시점부터 원래의 임무 목적지가 아닌 공격자가 주입시킨 GPS로 UAV가 이동한 것을 확인할 수 있다. 즉 UAV에게 원래의 임무 목적지가 있음에도 불구하고 본 논문에서 제안하는 공격 방법을 이용했을 경우 공격자가 UAV를 이륙지점으로 돌려보내거나 공격자가 원하는 GPS 좌표로 UAV를 이동시킬 수 있다는 것을 확인할 수 있다. 해당 실험은 setpoint_position/local과 setpoint_position/global 토픽을

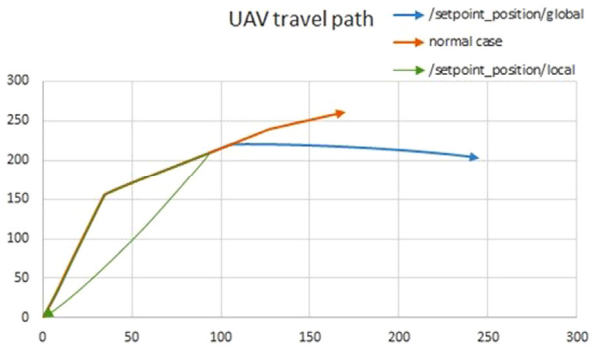

그림 9. 각 상황에 따른 UAV 비행 경로.
Fig. 9. UAV travel path for each situation.

통한 공격 가능 여부를 파악하기 위해 STIL (Software In The Loop)에서 이루어졌으며 이는 실제 실험 환경과 같은 SITL 환경에서 이루어졌다. 이후 나오는 실험결과는 실제 실험을 통해 이루어졌다.

두 번째 실험은 우리가 제안한 공격 방법이 앞서 말했던 무인기 무력화 단계 중 세 번째 단계를 만족하는 지 확인한다. 이와 비교하기 위해 무인기 무력화 단계 중 세 번째 단계를 만족하지 못하는 무력화 방법과 함께 비교한다. 세 번째 단계를 만족하지 못하는 무력화 방법은 MAVLink 프로토콜의 취약점을 이용한 무인기 무력화 방법으로써 무인기가 임무를 수행하는 중 공격자가 disarm 패킷을 주입시키는 것이다. 이렇게 공격받은 UAV는 공격자가 보낸 disarm 패킷을 걸러내지 못하고 비행을 하던 중임에도 불구하고 날개의 회전을 멈추고 비행하던 고도에서 추락하게 된다. 이 경우, 지역의 상황과 UAV의 비행 고도 상황에 따라 치명적인 재산피해, 인명피해가 발생할 수 있다.

우리는 앞서 말한 세 번째 단계를 만족시키지 못하는 공격 방법인 패킷 주입 공격과 우리가 제안한 공격 방법인 MAVROS의 취약점을 이용한 공격을 비교해 보았다. 먼저 UAV에게 비행 임무를 할당하여 비행을 하게 한 후 각각 약 15초 근처에서 공격을 해 보았다. 그림 10과 11은 UAV의 비행 임무 중 공격자가 공격한 상황을 고도를 통해 나타낸 그림이다. 그림 10은 MAVLink 프로토콜의 취약점을 통해 공격한 그림으로 UAV가 비행 임무 중 disarm 패킷을 보내 공격자가 UAV를 무력화시킨 그림이다. 그림을 보면 1m로 고도를 유지하며 비행하던 UAV가 약 1초도 안되어 고도가 0까지 떨어지며 땅에 부딪힌 것을 확인할 수 있다. 그에 반해 그림 11은 우리가 제안한 공격 방법을 사용했을 때의 UAV 고도를 나타낸 그림으로 1m의 고도로 잘 유지하던 UAV는 서서히 고도를 낮추어 착륙하게 되는 모습이다. 이를 통해 UAV의 추락을 통한 무력화가 아닌 보다 안전한 방법으로 무력화할 수 있는 것을 알 수 있다.

3. 방어 API 적용 실험

우리가 제안한 보안을 위한 API를 MAVROS에 적용시켰을 경우, 앞서 우리가 제안한 공격 방법을 시행해보았다. 그림 11에서는 공격을 보안을 위한 API를 적용시키지 않았을 경우, 약 15초 지점에서 공격을 받아 원래 수행하려고 하던 임무를 수행하지 못하고 공격자가 지정한 좌표로 이동

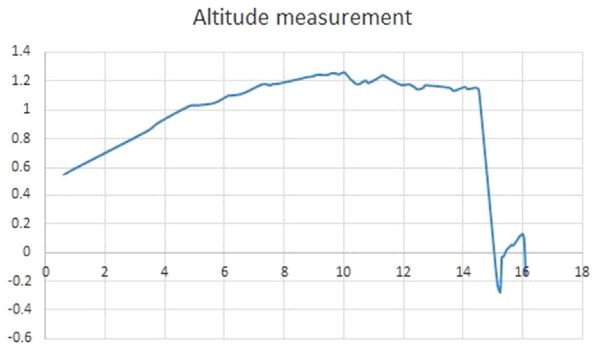

그림 10. 기존 공격에 대한 UAV 고도.
Fig. 10. UAV altitude to an existing attack.

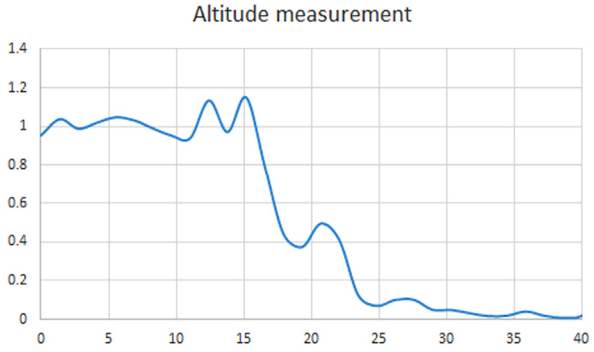

그림 11. 제안된 공격에 대한 UAV 고도.
Fig. 11. UAV altitude to a proposed attack.

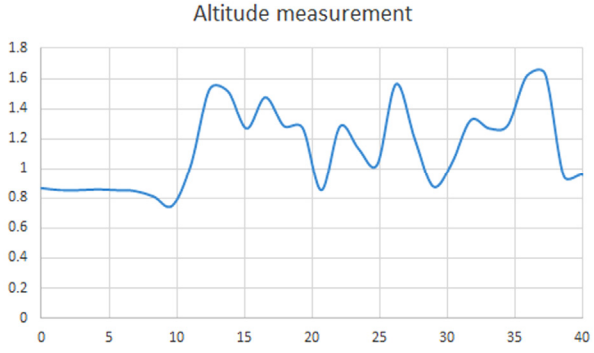

그림 12. 보안 API가 적용된 후 공격에 대한 UAV 고도.
Fig. 12. UAV altitude for the attack after security API is applied.

하는 것을 확인할 수 있었다. 하지만 보안 API를 적용한 후 위와 동일한 실험을 한 결과인 그림 12를 보면 1 m의 높이로 40초간 비행한 UAV의 고도를 확인할 수 있다. 우리가 제안한 보안 API를 적용시킨 UAV는 공격자가 무단으로 publisher 노드를 등록할 수 없기 때문에 해당 공격에 대해 방어할 수 있다.

VI. 결론

CPS의 주요 어플리케이션 중 하나인 무인 이동체가 용도를 넓힘에 따라 무인 이동체의 무력화 연구도 중요해지고 있다. 우리는 무인 이동체 중 UAV를 가정하여 연구하였다. 무인기 무력화 단계는 무인기 피아 식별 단계, 무인기 임무 무력화 단계, 무인기 안전 회수 및 사후 처리단계로 크게 세 단계로 나뉠 수 있다. 그러나 대부분의 연구는 무인기 안전 회수 및 사후 처리단계까지 고려하지 않고 있으며 무인기 임무 무력화 단계에 그치고 있다. 우리는 자율주행, navigation, 충돌 회피 등의 기능 구현을 위해 일반적으로 사용하는 MAVROS 환경에서 취약점을 찾고 그 취약점을 이용한 공격 방법을 제안하였다. 제안하는 공격 방법은 추락을 통한 무인기 무력화가 아닌 원하는 지점으로 착륙시키거나 이륙지점으로 돌려보내는 무력화 방법으로 이로 인한 2차 피해가 발생하지 않는다. 또한 이러한 취약점을 막기 위해 보안을 위한 가벼운 MAVROS API를 제안하였다. 이는 UAV의 상태를 LOCK, UNLOCK 두 가지 상태로 나누어 새로운 publisher 노드의 등록을 막음으로써 취약점을 해결하였다.

REFERENCES

[1] K.-J. Park, R. Zheng, and X. Liu, “Cyber-physical systems: Milestones and research challenges,” *Computer communications*, vol. 36, no. 1, pp. 1-7, Dec. 2012.

[2] K.-J. Park, J. Kim, H. Lim, and Y. Eun, “Robust path diversity for network quality of service in cyber-physical systems,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2204-2215, Nov. 2014.

[3] Y.-M. Kwon, J. Yu, and B.-M. Cho, “Empirical analysis of mavlink protocol vulnerability for attacking unmanned aerial vehicles,” *IEEE Access*, vol. 6, pp. 43203-43212, Aug. 2018.

[4] J. Yoon, H. Lee, and K.-J. Park, “Security enhancement in wi-fi communication between UAV and GCS,” *The Journal of Korean Institute of Communications and Information Sciences (in Korean)*, pp. 400-401, Jun. 2019.

[5] J. M. Yu, J. Yoon, and K.-J. Park, “Risk analysis of UAV and GCS for network attacks,” *Communications of the Korean Institute of Information Scientists and Engineers (in Korean)*, vol. 37, no. 1, pp. 29-37, Jan. 2019.

[6] A. Y. Javaid, W. Sun, V. K. Devabhaktuni, and M. Alam, “Cyber security threat analysis and modeling of an unmanned aerial vehicle system,” *Proc. of 2012 IEEE Conference on Technologies for Homeland Security (HST)*, Waltham, MA, USA, Nov. 2012.

[7] J. Yoon, J. Yu, and K.-J. Park, “UAV network security enhancement model using private blockchain,” *Proc. of KICS Winter Conference (in Korean),* Yongpyong, Korea, pp. 1259-1260, Jan. 2019.

[8] S. Deng, X. Gao, Z. Lu, and X. Gao, “Packet injection attack and its defense in software-defined networks,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 695-705, Oct. 2017.

[9] E. Bertino and N. Islam, “Botnets and internet of things security,” *Computer*, vol. 50, no. 2, pp. 76-79, Feb. 2017.

[10] A. K. Simpson, F. Roesner, and T. Kohno, "Securing vulnerable home IoT devices with an in-hub security manager," *2017 IEEE PerCom Workshops*, Kona, HI, USA, Mar. 2017.

[11] S. Bhattacharya and T. Başar, "Game-theoretic analysis of an aerial jamming attack on a UAV communication network," *Proceedings of the 2010 American Control Conference*, IEEE, 2010.

[12] M. Sliti, W. Abdallah, and N. Boudriga, "Jamming attack detection in optical UAV networks," *Proc. of 2018 20th International Conference on Transparent Optical Networks (ICTON)*, IEEE, 2018.

[13] MALink Protocol. Accessed: Dec. 14, 2020. [Online]. Available: https://mavlink.io/en/

[14] ROS2 DDS security. Accessed: Dec. 14, 2020. [Online]. Ava ilable: https://design.ros2.org/articles/ros2_dds_security.html

[15] J.-H Lee, K.-J Ahn, T.-G Lee, K.-I. Min, O.-S. Kwon, S.-W. Baek, M.-C. Park, D.-H. Cho, J.-R. Hwang, J.-K. Kang, S.-H. Lee, D.-Y. Seo, J.-H. Kim, and Y. Kang, "Developme nt of control system for international autonomous driving competition using robot operating system," *Journal of instit ute of control robotics and systems (in Korean)*, vol. 25, no. 8, pp. 363 – 373, May. 2019.

[16] C.-Y. Jung and T. Kim, "Development of ROS-based mobil e robot system and experiments on indoor autonomous drivi ng," *Journal of institute of control robotics and systems (in Korean)*, vol. 25, no. 5, pp. 438 – 444, May. 2019.

윤 지 영

2017년 계명대학교 컴퓨터공학과 졸업. 2020년 대구경북과학기술원 정보통신융합전공(공학석사). 2020년~현재 (주)인투시어 프로젝트 매니저. 관심분야는 Cyber-Physical Systems, Data science.

이 효 준

2017년 계명대학교 컴퓨터공학과 졸업. 2019년~현재 대구경북과학기술원 정보통신융합전공 석사과정 재학 중. 관심분야는 Cyber-Physical Systems.

박 경 준

1998년 서울대학교 전기공학부 졸업. 2000년 서울대학교 전기공학부(공학석사). 2005년 서울대학교 전기컴퓨터공학부(공학박사). 2011년~현재 대구경북과학기술원 정보통신융합전공 교수. 관심분야는 Resilient Cyber-Physical Systems.