PX4 Autopilot을 활용한 드론 악성 소프트웨어 개발

김성수° 조진성
경희대학교 컴퓨터공학과
korkeep@khu.ac.kr chojs@khu.ac.kr

Development of Malicious Drone Software Using PX4 Autopilot

Sungsu Kim° Jinsung Cho
Department of Computer Science and Engineering, KyungHee University

요 약

최초 군사용 목적으로 연구된 드론이 실생활 분야에 적용되기 시작한 것은 이미 오래전 일이다. 최근에는 인공지능, 클라우드, 빅데이터 분석 등 첨단기술과 결합한 서비스를 제공하는 등 드론의 쓰임새가 확장되고 있다. 드론을 이용한 서비스와의 연결은 사용자에게 편리함을 제공하는 동시에 많은 취약점을 낳는다. 본 논문에서는 드론 공격 사례를 바탕으로 공격 시나리오를 체계화하고, PX4 Autopilot을 활용해 시뮬레이팅 환경에서 각 시나리오에 해당하는 악성 소프트웨어를 구현함으로써, 서비스 개발자들에게 어떤 공격에 대비해야 하는지 방향을 제시하는 것을 목표로 한다.

1. 서 론

오늘날 드론을 이용한 서비스의 세계 시장 규모는 2019년 기준 약 114억 달러로 평가됐고, 2025년에는 약 202억 달러까지 성장할 수 있다고 전망된다[1]. 특히 인공지능, 클라우드, 빅데이터 분석 등의 첨단기술이 드론과 결합하면서 장애물 회피, 자동 충돌 방지, 이미지 처리 기능을 제공하는 등 드론의 기능과 쓰임새가 점점 확장되고 있는 상황이다. 드론을 이용한 서비스를 통해 사용자는 많은 편리함을 누리고 있지만, 동시에 드론의 보안 취약성 또한 증가하고 있다.

초기 군사용 목적으로 사용했던 드론이 경량화된 부품으로 사용자에게 대량 공급되면서, 드론 기기의 취약점이 지속적으로 발견되고 있다. Parrot사의 AR Drone의 경우, 시리얼 포트를 변조된 펌웨어와 통신하도록 변경한다면 드론의 제어권을 탈취할 수 있다는 취약점이 밝혀졌다[2]. DJI사의 경우, DJI 드론 기기와 통신하는 웹 애플리케이션에서 XSS 공격을 통해 쿠키에 있는 정보를 이용하여 사용자 권한을 획득하는 방법도 존재했다[3].

본 논문에서는 실제 드론의 공격 사례와 드론 취약점에 관련된 기존 연구를 종합해 드론 공격 시나리오를 체계화함으로써, 드론 서비스 개발자들에게 어떤 공격에 대비해야 하는지 방향을 제시하는 것을 목표로 한다. 논문의 구성은 다음과 같다. 2장에서 PX4 드론의 작동 흐름인 Flight Stack과 PX4에서 제공하는 메시징 프로토콜에 대해 소개하고, 시뮬레이팅 환경을 구축하는데 사용된 소프트웨어를 소개한다. 3장에서는 드론의 작동 흐름을 분석해 드론 공격 시나리오의 구상에 참고하여, 시뮬레이팅 환경에서 실제 적용해본다. 마지막으로 4장에서 논문의 기대 효과를 제시하면서 논문은 마무리된다.

이 논문은 교육부 및 한국연구재단의 기초연구사업(NRF-2017R1D1A1B04035914)과 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학 사업(2017-0-00093)의 지원으로 수행된 연구결과임.

2. 관련 연구

2.1 uORB

uORB는 PX4 드론 기기의 내부에서 사용되는 프로세스 사이의 통신(IPC; Inter Process Communication) 매커니즘으로, Publish/Subscribe 모델의 비동기적 메시징 API이다. PX4 드론의 내부를 구성하는 컴포넌트들은 uORB를 이용해서 데이터를 주고받는다. 개발자는 PX4에서 제공하는 uORB API를 이용해, PX4 드론을 구성하는 다른 컴포넌트와 데이터를 주고받는 드론 소프트웨어 개발이 가능하다.

2.2 MAVLink

MAVLink는 드론 기기 외부에 있는 컴포넌트와 통신을 위한 가벼운 메시징 프로토콜이다. 사용자가 드론 기기의 외부에서 정의한 명령은 MAVLink 프로토콜을 통해 드론 기기로 전달된다. 대표적으로 PX4 드론이 MAVLink 프로토콜을 이용해 2.3에서 설명하고 있는 QGroundControl과 데이터를 주고받는다.

2.3 QGroundControl

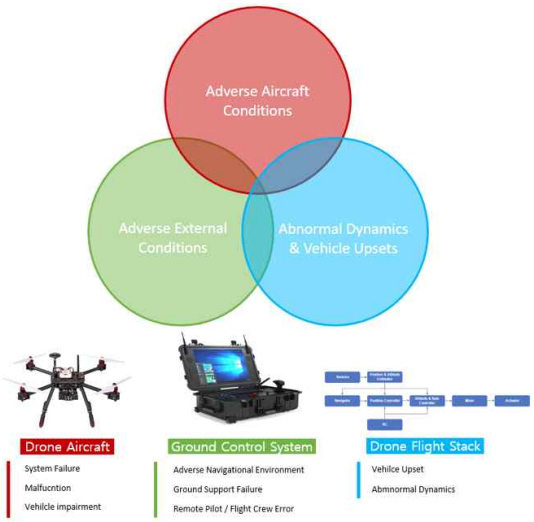
QGroundControl은 PX4와 같이 MAVLink 프로토콜을 지원하는 드론의 GCS(Ground Control System)이다. GCS는 지상에서 드론의 비행 상태를 확인하고, 사용자에게 드론을 조종할 수 있는 환경을 제공하는 관제시스템이다. 이외에도 QGroundControl은 PX4의 펌웨어 업로드, 미션 플래닝 기능을 통한 자동 비행, GCS와 연결된 드론의 위치를 실시간으로 보여주는 UI, 드론 기기 설정 및 캘리브레이션 등의 기능을 사용자에게 제공한다.

2.4 jMAVSim

jMAVSim은 MAVLink 프로토콜을 지원하는 멀티로터 유형의 가상 드론 시뮬레이터이다. 사용자는 자신이 작성한 소프트웨어를 드론에 직접 적용하기 이전에 jMAVSim 시뮬레이션 환경에서 미리 테스트해볼 수 있다.

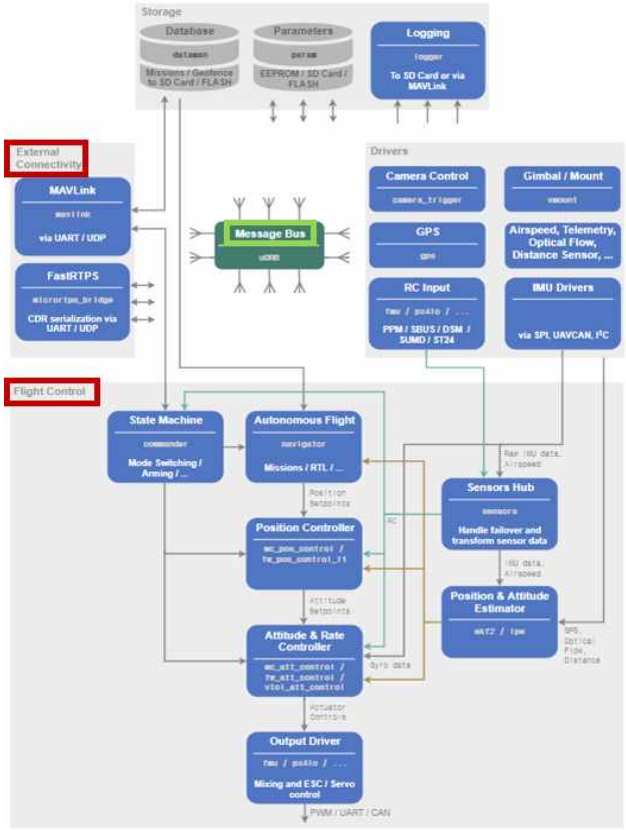
3. 시나리오 제안 및 구현

3절에서는 드론 공격 시나리오를 소개하고, 각 시나리오에 해당하는 악성 소프트웨어를 어떤 방식으로 구현했는지 설명한다. 시나리오에 대한 설명은 공격 유형, 공격 지점, 공격 방법 순으로 이어진다.



[그림 1] 드론 공격 유형 분류 3요소

[그림 1]에서 드론 공격유형에 대한 분류는 드론이 실제 공격된 사례를 바탕으로 드론의 공격 과정에 적용 가능한 모든 위협 요인을 도출하여, '드론 기기 내부의 상태 이상', '드론의 외부, GCS 상태 이상', '드론 비행 과정에서 동적 요소에 의한 상태 이상'이라는 세 가지 기준으로 구분할 수 있다[4].



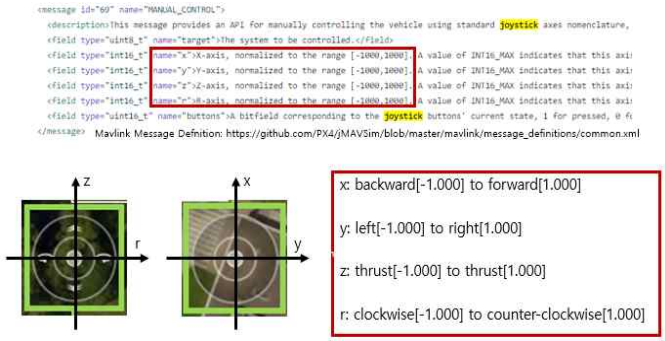
[그림 2] PX4 시스템 아키텍쳐

드론 공격 지점에 대한 분류는 [그림 2]의 PX4 시스템 아키텍쳐를 참고했다[5]. 특히 PX4 드론 기기에 내장된 펌웨어를 통해 데이터 전달이 이루어지는 Flight Control 부분과, QGroundControl 및 RC 컨트롤러와 같은 외부에서 데이터 전달이 이루어지는 External Connectivity를 중심으로 공격 지점을 분류했다.

3.1 드론 공격 시나리오

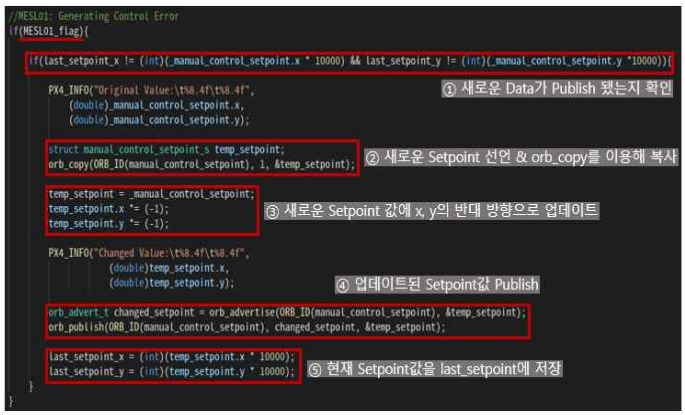
1) Generating Control Error

- 공격 유형: 드론 기기 내부의 상태 이상, 드론 비행 과정에서 동적 요소에 의한 상태 이상
- 공격 지점: 드론에 내장된 펌웨어
- 공격 방법:



[그림 3] MAVLink 메시지 정의

[그림 3]에서 PX4 드론은 'MANUAL_CONTROL'이라는 MAVLink 메시지의 x, y, z, r 파라미터의 값을 통해 움직임이 제어된다. 파라미터는 각각 -1.000에서 1.000까지의 범위를 가지고, 그 크기의 절댓값이 커질수록 드론이 이동하는 속도가 빨라진다.



[그림 4] Generating Control Error

[그림 4]는 RC 컨트롤러의 입력과 반대로 동작하는 기능을 구현한 부분이다. 먼저 새로운 Setpoint Data가 Publish됐는지 확인하고, 새로운 Data가 설정됐다면 이 값을 orb_copy 함수를 이용해 복사해온다. 복사해온 값의 x, y값에 -1을 곱함으로써, 앞, 뒤 / 좌, 우의 방향이 바뀌게 된다. 방향이 변경된 값을 orb_advertise, orb_publish 함수를 이용해 Publish하고, 변경된 값을 last_setpoint_x, y에 저장함으로써, 첫 번째 단계에서 앞, 뒤 / 좌, 우 방향이 바뀐 Data에 다시 -1을 곱하는 과정이 이루어지지 않도록 설정한다.

2) Unintended Mission Conduct

• 공격 유형: 드론 기기 내부의 상태 이상, 드론의 외부, GCS 상태 이상, 드론 비행 과정에서 동적 요소에 의한 상태 이상

• 공격 지점: 드론에 내장된 펌웨어

• 공격 방법:

[그림 5] Unintended Mission Conduct

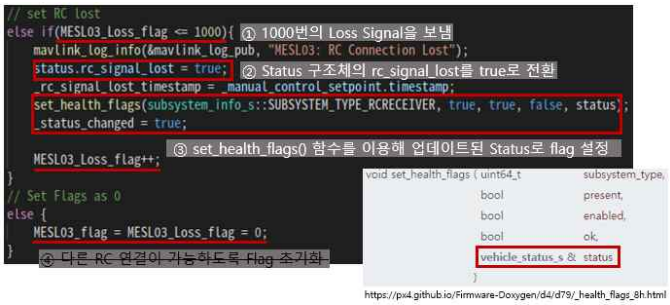
[그림 5]는 지정된 명령(Mission)을 수행하는 기능의 구현이다. 드론의 Mission에 관련된 정보를 저장하는 역할은 mission_s 구조체가 담당한다. mission_s 구조체를 이용하여 mission 변수를 선언하고, Dataman 모듈의 Mission, Setpoint 값을 변수에 업데이트한다. 공격자에 의해 Mission이 지정되어 있으면, 공격자가 지정한 Mission이 변수를 통해 Publish된다. 앞선 단계에서 Publish한 Mission이 정상적으로 Publish되었는지 확인하고, Mission을 수행할 수 있는 상태인지 체크한다. 이후 Commander의 State(상태) 전환 기능을 통해 Mission 수행 모드로 바뀌면서 공격자가 지정한 Mission이 수행된다. Mission이 끝나면 Mission_flag를 false로 전환함으로써 앞서 RC 컨트롤러의 입력을 받을 수 없도록 설정한 것을 원상태로 복구한다.

3) RC Connection Lost

• 공격 유형: 드론의 외부, GCS 상태 이상, 드론 비행 과정에서 동적 요소에 의한 상태 이상

• 공격 지점: 드론에 내장된 펌웨어

• 공격 방법:



[그림 6] Unintended Mission Conduct

[그림 6]은 컨트롤러와의 연결이 끊어진 상태로 변경하는 기능의 구현이다. 드론의 Status에 관련된 정보를 저장하는 역할은 vehicle_satus_s 구조체가 담당한다. RC 컨트롤러와의 연결 유지 여부를 판단하는 값은 rc_signal_lost로써, 이 값이 true면 드론은 RC 컨트롤러와의 연결이 끊어진 것으로 판단한다. 공격자의 trigger에 의해 RC Connection Lost 기능이 실행된다면, 악성 모듈에서 rc_signal_lost 값을 true로 전환하고, set_health_flag 함수를 이용해 RC 컨트롤러와의 연결이 끊어진 상태로 설정한다.

3.3 공격 결과 동영상 시연

각 시나리오 항목에 대한 시연을 동영상으로 제작하여 Youtube에 업로드했다[1]. 악성 소프트웨어의 소스코드는 참고문헌[6]의 Github 링크에서 상세히 확인할 수 있다.

4. 결 론

본 논문에서는 체계화된 드론 공격 시나리오를 제안하고, 각 시나리오에 해당하는 악성 소프트웨어를 구현함으로써 잠재된 위협을 시각화했다. 3절에서 소개한 드론 공격 시나리오를 통해 어떤 공격에 대해 보완이 필요한지 방향을 제시해주고, 본 논문을 시작으로 드론 보안의 필요성에 대한 담론이 형성되기를 기대해본다.

참고문헌

[1] 과학기술일자리진흥원,「드론 기술 및 시장동향 보고서」 S&T Market Report, VOL.67, 7-15 (2019)
[2] Rahul Sasi,「Maldrone the First Backdoor for Drones」grace4hackers.com (2015)
[3] Oded Vanunu, Dikla Barda and Roman Zaikin「DJI Drone Vulnerability」research.checkpoint.com (2018)
[4] Christine M. Belcastro, Richard L. Newman, Joni Evans, David H. Klyde, Lawrence C. Barr, Ersin Ancel「Hazards Identification and Analysis for Unmanned Aircraft System Operations」AIAA Aviation Technology, Integration, and Operations Conference, 10-14 (2017)
[5] PX4 Development Guide,「PX4 Architecture overview」https://dev.px4.io/master/en/ (2019)
[6] Github 링크, https://github.com/korkeep/PX4_MalSW

1) 시연① https://youtu.be/mBD8B45kf1c
시연② https://youtu.be/SDU53NbrXFk
시연③ https://youtu.be/xUdr50TkoaE