드론의 무선 네트워크 보안 취약점을 이용한 탈취 및 대응

손주환* · 심재범* · 이재구* · 정일안*

*한국정보보호시스템(주)

Hijacking Attack using Wireless Network Security Vulnerability in Drone and Its Countermeasure

Juhwan Son* · Jaebum Sim* · Jae-Gu Lee* · Il-Ahn Cheong*

*KINFOSEC Co. Ltd.

E-mail : son.ch@kinfosec.co.kr, sjb@kinfosec.co.kr, jgl@kinfosec, qubcia@kinfosec.co.kr

요 약

최근 초경량 무인비행장치 드론은 방송촬영, 재난현장, 레저, 관측 이외에도 최초 드론의 개발 목적에 맞게 군사용으로도 활발하게 이용하고 있다. 그러나 드론의 활용이 많아지는 만큼 해킹 위협도 높아지고 있다. 특히 비행중인 드론이 탈취될 경우 조종자의 의도와 상관없이 악용되는 위험한 상황이 발생할 수 있다. 이미 이란, 중국 등에서는 정찰중인 타 국가의 드론에 대해 해킹을 시도하여 탈취한 사례가 있다. 본 논문에서는 상용 드론 제품의 강제 탈취를 위해 드론에서 사용하고 있는 무선 네트워크 Wi-Fi와 Bluetooth 통신의 보안 취약점을 분석한다. 그 결과 외부 비인가 공격자에 의한 드론 탈취 방법을 보이고 드론을 안전하게 보호하기 위한 대응 방안도 제시한다.

ABSTRACT

In recent years, Drone(lightweight unmanned aerial vehicle) is used for broadcast shooting, disaster scene, leisure, observation and military purposes. However, as the use of drones increases the threat of hacking is also rising. Especially when a flying drone is seized, a dangerous situation can occur which is abused regardless of the driver's intention. Already in Iran and China, there is a case of hacking and stealing the drones of other countries under reconnaissance. In this paper, we analyze the security vulnerabilities of Wi-Fi and Bluetooth communication in wireless network which are used in drones for stealing the commercial drones. The results provide a countermeasure to safeguard the drones against attempts by the unauthorized attacker to take out the drones.

키워드

드론, 보안 취약점, 무선 네트워크, 탈취

Ⅰ. 서 론

초경량 무인비행장치인 드론은 방송촬영, 재난현장, 레저, 관측과 같은 상업적 용도 또는 개인적 용도로 활용도가 점차 높아져 가고 있다. 최초 드론의 개발 목적에 맞게 군사용으로도 활발하게 이용되고 있지만, 다수의 드론 장비들이 여전히 취약한 보안에 노출되어 있다[1]. 대다수 드론 비행 조종에 사용되는 무선 주파수는 비면허 대역으로 기존 무선 보안 취약점이 그대로 적용되고, 드론 자체에 보안을 고려하지 않은 설계로 악의적 이용이 가능하다. 알려진 드론에 대한 보안 위협은 드론과 무선 컨트롤러 사이의 무선 신호를 이용한 공격, Wi-Fi, Bluetooth와 같은 무선 프로토콜 상의 통신 보안 문제점을 이용한 공격, 그리고 드론에 탑재된 소프트웨어의 취약점을 이용한 공격 등 크게 3가지 공격 유형으로 분류할 수 있다.

무선 신호를 이용한 공격 중 대표적인 공격으로는 드론과 무선 컨트롤러 사이 흐르는 신호 자체를 교란(Jamming)하여 드론과 무선 컨트롤러 사이 정상적인 통신을 하지 못해 드론을 강제 착륙시키는 공격으로 드론을 방어하기 위한 용도(Anti-Drone)로 사용되기도 한다[2]. 무선 프로토콜 상의 통신 보안 공격은 Wi-Fi, Bluetooth 등 무선

프로토콜의 WEP/WPA/WPA2 인증 암호 탈취, 데이터 스니핑(Sniffing), DoS 공격, Ping of Death 공격 등의 보안 취약점을 악용하거나, 신호 교란 및 무선 프로토콜 통신을 이용하여 드론 강제 조작 및 탈취, 사진/영상을 탈취할 수 있는 공격이다. 드론 탑재 소프트웨어 공격은 암호화 되지 않은 데이터 탈취(비행 GPS 데이터 및 사진/영상 자료 등) 및 개발자의 실수를 찾아 악성코드 감염과 같은 방법으로 드론의 제어권을 탈취하는 공격이다[3].

드론 위협 사례로, 2011년 알카에다를 추종하던 레즈완 페르도스는 미국을 표적으로 지하드(성전)를 도모하기 시작해 드론을 이용한 폭탄 테러를 준비하여 미수에 그쳤으나, 최초로 드론이 테러에 이용될 수 있음을 알린 사건이 되었다. 2014년 Blackhat Asia에서는 비행 중인 드론을 이용하여 무선인터넷망으로 둔갑하여 근처에 있는 스마트폰들의 연결을 모두 허용한 후 스마트폰 내 통신되는 모든 내용을 확인하여 약 150여개의 정보를 탈취했다. 2015년 1월 미국 백악관 드론 충돌 사건, 2015년 4월 일본 도쿄 지요다구 총리 관저 옥상에서 미량의 방사선을 내뿜는 드론 발견, 2011년 12월 이란 핵 시설을 정찰 중이던 미국 드론 발견 등 모두 GPS 방해 공격(Jamming) 공격으로 추락하였다[2, 3, 4]. 이처럼 드론은 테러 및 해킹의 피해에서 벗어나지 못하고 있다. [그림 1]의 2017년 2월 가트너 조사에 따르면, 최근 개인/상업용 드론은 가파른 추세로 성장하고 있고, 그에 따라 성능 또한 고성능으로 충분히 테러 및 악의적인 용도로 활용이 가능하지만 취미용 드론으로 분류되기에 국토교통부의 신고절차기 불필요하고 비행승인조차 불필요하다[5, 6]. 이와 같은 규제 한계 때문에 이미 드론으로 인해 테러, 암살 등의 악의적인 행위가 반복되고 있는 상황에서 공격자가 강제 탈취된 드론을 이용하여 악의적 행위로 이용을 하게 되면 드론의 본소유자가 악의적 행위에 대한 장본인이 될 수 있다.

2016~2017년 개인용 상업용 드론 매출 전망(단위: 천 달러)

부문	2016년	2017년
개인용	1,705,845	2,362,228
상업용	2,799,272	3,687,128
총계	4,505,117	6,049,356
전체 매출 성장률	35.50%	34.30%

2016~2017년 개인용 상업용 드론 출하량 전망(단위: 천 대)

부문	2016년	2017년
개인용	20,419	28,173
상업용	1,103	1,741
총계	21,522	29,914
전체 매출 성장률	60.30%	39.00%

그림 1. 개인/상업용 드론의 성장률

본 논문에서는 일반적인 상용 드론 제품이 사용하고 있는 무선 네트워크의 보안 취약점을 이용하여 탈취에 대한 분석 과정을 기술하고, 그 결과를 통해 외부의 비인가 공격자로부터 드론에 대한 접근 및 탈취 시도로부터 드론을 안전하게 보호하기 위한 근본적 해결 방안을 제안한다.

Ⅱ. 본 론

본 장에서는 드론 제어 프로토콜을 이용하여 무선 컨트롤러 없이 강제 탈취가 가능하다는 것을 확인하기 위해 드론이 사용하는 무선 네트워크의 취약점을 이용하여 통신흐름을 분석하고 그 과정에 대해 기술한다.

드론은 무선 컨트롤러와 무선 네트워크로 연결하여 주기적으로 무선 컨트롤러가 전송하는 제어 명령을 받아 수행하고 그 결과를 무선 컨트롤러로 다시 전송한다. 드론 소유자조차 모르게 드론을 강제 탈취하기 위해서는 다음과 같은 과정이 필요하다. 드론 소유자 몰래 외부에서 비인가 공격자가 무선 네트워크를 기반으로 조종하는 드론의 제어 프로토콜을 파악한 후 드론과 무선 네트워크로 연결한다. 파악된 제어 프로토콜을 이용하여 드론으로 전달하고 드론이 해당 기능에 대해 동작하는지 확인한다. 다음 실험과 같이 제어된 드론은 언제든지 소유자 몰래 강제 탈취가 가능하며 이와 동시에 탈취된 드론은 악의적으로 이용하게 된다.

2.1 드론 강제 탈취 실험 환경

실험에 사용된 드론 및 무선 네트워크의 정보는 다음 [표 1]과 같다.

표 1. 분석 대상 드론 및 무선 네트워크

대상	제어용 무선 네트워크
Parrot Mambo[7]	Bluetooth
Parrot Bebop2[8]	Wi-Fi

[표 2]는 드론의 강제 탈취를 실험하기 위해 사용한 도구 목록 및 환경이다.

표 2. 사용 도구 및 환경

도구명	용도
Cain[9]	ARP-Spoofing 공격
Wireshark[10]	패킷 스니핑
NMAP[11]	포트 스캐닝
Ubertooth[12]	Bluetooth 스니핑
NodeJS[13]	드론 공격 스크립트 작성
Drone SDK[14-16]	드론 제어 SDK 및 API

2.2 Bluetooth를 사용하는 드론의 탈취 분석

Bluetooth는 2.4Ghz 대역을 사용하는 근거리 무선 네트워크이다. Bluetooth의 특징은 1:1 통신으로 드론과 무선 컨트롤러가 서로 페어링이 되는 순간 해당 기기들의 통신 중간 과정에 간섭할 수 없지만, [그림 2]와 같은 “Ubertooth”라는 별도의 장비를 이용하면 Bluetooth로 연결되어 통신하는 장치들의 통신과정을 모니터링할 수 있다.

그림 2. Ubertooth

또한, [그림 3]과 같이 Ubertooth 제조사 측에서 제공하는 라이브러리와 Wireshark를 합쳐서 Parrot Mambo와 무선 컨트롤러 사이에 통신하는 제어 프로토콜에 대한 모든 내용을 가로챈다.

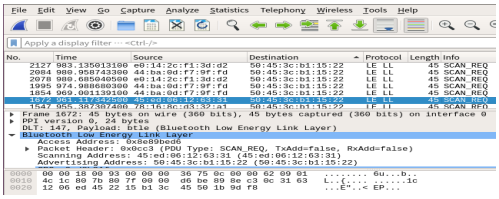

그림 3. Ubertooth + Wireshark를 이용한 Bluetooth 통신 내용 가로채기

Parrot社는 자사의 제품을 다양한 형태로 제어할 수 있도록 SDK를 제공하고 있다. Parrot社 이외 최대 드론 제조 기업인 DJI社도 SDK를 제공하여 드론 사용자들이 자유롭게 드론을 제어할 수 있도록 제공하고 있다. Parrot社 개발자센터로 접속을 하게 되면 공개된 여러 가지 오픈소스 프로젝트를 확인할 수 있다.

가로챈 패킷은 Parrot 개발자센터의 공개된 오픈소스 중 ARSDK-XML[15]프로젝트를 기반으로 [그림 4]와 같이 패킷을 직접 사용할 수 있는 NodeJS의 스크립트로 변환한다.

```
<cmd name="Landing" id="3">
        Ask the MiniDrone to land
</cmd>
<cmd name="Emergency" id="4" buffer="HIGH_PRIO" timeout="RETRY">
        Put drone in emergency state
</cmd>
<cmd name="AutoTakeOffMode" id="5">
        Set MiniDrone automatic take off mode
        <arg name="state" type="u8">
                State of automatic take off mode
        </arg>
</cmd>
<cmd name="FlyingMode" id="6">
        Set drone FlyingMode. Only supported by WingX
        <arg name="mode" type="u8">
                Drone Flying Mode
                <enum name="quadricopter">
                        Fly as a quadrictopter
                </enum>
                <enum name="plane_forward">
                        Fly as a plane in forward mode
                </enum>
                <enum name="plane_backward">
                        Fly as a plane in backward mode
                </enum>
        </arg>
</cmd>
```

그림 4. ARSDK-XML 내 프로토콜

NodeJS로 작성된 스크립트가 무선 컨트롤러 보다 빠르게 페어링 시 패스워드가 없는 것을 이용하여 Parrot Mambo와의 Bluetooth 페어링을 우선 선점하고, [그림 5]와 같이 NodeJS의 noble (Bluetooth 제어 모듈)을 사용하여 Parrot Mambo가 탐지되는 순간 연결을 가로챈 후 지정된 명령을 제어 프로토콜을 통해 전송함으로써 드론을 강제로 탈취한다.

```
kinfosec@drone-system:~/Drone/Exploit$ sudo node mambo.js
command > info: Peripheral found Mambo_486373
info: Device connected Mambo_486373
info: Setting max altitude to 2m
info: Setting max tilt to 40% (20° max)
info: Setting max vertical speed to 0.5 m/s
info: Setting max rotation speed to 150 °/s

command >
```

그림 5. NodeJS로 제작한 Parrot Mambo 강제 탈취 도구

2.3 Wi-Fi를 사용하는 드론의 탈취 분석

Wi-Fi는 2.4Ghz/5Ghz 대역을 사용하는 1:N 통신이 특징이고, 연결 시 WPA/WPA2와 같은 암호인증으로 보안을 제공하는 무선 네트워크이다.

그러나 [그림 6]과 같이 Parrot Bebop2는 Wi-Fi를 사용하지만, 암호 인증이 설정되지 않은 상태로 Wi-Fi를 사용하고 있어 누구나 해당 드론의 Wi-Fi로 접근이 가능한 상태이다.

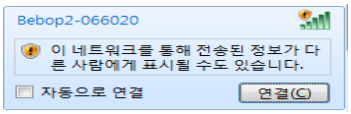

그림 6. Parrot Bebop2 Wi-Fi

Wi-Fi로 접속 후 드론 내 동작하는 서비스를 확인하기 위해 [그림 7]과 같이 포트스캔 도구인 NMAP을 통해 조사한 결과, TCP에서 사용되는 포트는 HTTP 서비스(8080)과 FTP 서비스(21)이고, UDP에서 사용되는 포트는 드론 제어를 위한 54321, 43210, 44444로 총 3개의 포트가 존재한다. 여기서, TCP에서 사용되는 HTTP 서비스는 드론의 상태를 상세하게 출력하기 위해 사용하는 프로토콜이고, FTP는 사진 또는 영상 등 다양한 파일을 사용자와 드론 사이에서 파일을 쉽게 전송하기 위해 사용되는 프로토콜이다. [그림 8]과 같이 UDP에서 사용되는 모든 포트들도 드론 제어에 사용되지만, 해당 포트들이 강제 탈취에도 악용된다.

```
Nmap scan report for 192.168.42.1
Host is up (0.0043s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
8080/tcp open  http-proxy
MAC Address: A0:14:3D:D1:31:0A (Parrot SA)
```

그림 7. NAMP을 통한 포트스캔 결과

```
\texttt{\{ "status":0, "c2d\_port":54321, \\
"arstream\_fragment\_size":65000, \\
"arstream\_fragment\_maximum\_number":4, \\
"arstream\_max\_ack\_interval":-1, "c2d\_update\_port":51, \\
"c2d\_user\_port":61 \}}
```

그림 8. SDK 내 정의된 제어 UDP 포트

Parrot Bebop2의 Wi-Fi 접속, Parrot Bebop2와 무선 컨트롤러 사이의 통신 내용은 [그림 9]와 같이 Cain 도구를 이용하여 ARP-Spoofing 공격을 통해 서로 통신하는 내용을 Wireshark 도구로 가로챈다.

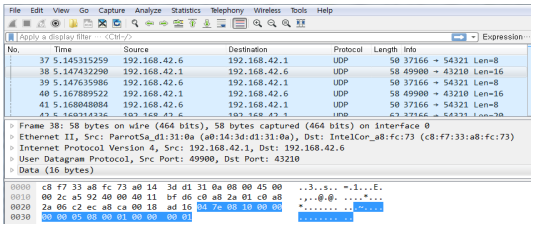

그림 9. ARP-Spoofing을 통해 프로토콜 확인

스니핑된 패킷을 Parrot 개발자센터 내 오픈소스인 BebopSample[16] 소스를 참고하고, [그림 10]과 같이 제어 프로토콜을 NodeJS의 스크립트를 이용하여 직접 전송할 수 있도록 구성한다. 그 다음 Parrot Bebop2의 Wi-Fi로 연결하고 제어 프로토콜을 통해 명령을 전송하여 강제로 탈취한다.

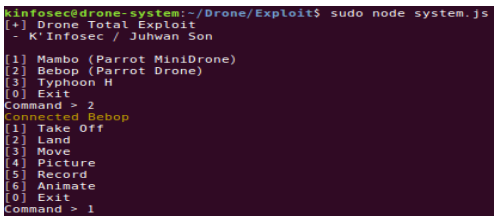

그림 10. NodeJS로 제작된 Parrot Bebop2 강제 탈취 도구

Ⅲ. 결 론

본 논문에서는 드론의 취약점 중 대부분의 드론이 사용하는 Bluetooth와 Wi-Fi의 무선 네트워크 통신 보안 취약점을 악용하여 드론을 강제 탈취할 수 있는지를 실험을 통해 확인하였다. 이러한 보안 취약점에 대해 안전한 드론 사용을 위한 대응 방안을 다음과 같이 제시한다.

첫째, Bluetooth와 Wi-Fi에 대한 패스워드가 설정되어 있지 않고 변경할 수 없도록 고정되어 출시된 제품은 사용자가 전혀 대처할 수 없으므로 주의해야 한다. Bluetooth의 경우 페어링 패스워드를 설정하거나 변경할 수 있도록 하고, Wi-Fi의 경우 인증 암호를 설정하거나 패스워드를 변경할 수 있도록 기본적인 인터페이스를 제공해야 한다.

둘째, Bluetooth나 Wi-Fi 통신의 UDP 데이터는 모두 평문으로 통신하므로 최소한의 인코딩 변환을 거쳐 전송하도록 해야 한다. 평문으로 통신하는 프로토콜은 공격자에게 쉽게 분석이 가능하다.

셋째, SDK를 제공할 때 프로토콜에 대한 상세한 내용은 직접적인 소스 코드로 공개되지 않아야 한다. 소스 코드 공개를 통해 누구나 드론 구조를 쉽게 파악하고 악의적인 행위를 하는 도구를 개발하여 본 논문에서와 같이 탈취 행위에 악용될 수 있다. 프로토콜에 대한 상세한 내용은 소스 코드 형태가 아닌 바이너리 또는 분석하기 어려운 형태로 제공되어야 한다.

넷째, 드론의 운영체제를 보안 기능이 적용된 OS를 적용해야 한다. 드론의 운영체제를 설계할 때 대부분 보안을 고려하지 않은 상태에서 설계하기 때문에 보안 기능이 적용된 드론 해킹 방지용 시큐어 임베디드 4(seL4) 마이크로 커널[17]과 같은 OS를 적용하는 것을 고려해야 한다.

상기와 같이 드론의 무선 통신 상 보안 취약점을 악용하여 강제 탈취하는 위험성을 점검하고 제조사에도 보완 대책을 마련할 수 있도록 정부 차원에서 제도적으로 적극 권고해야 한다. 또한, 드론 사용자에게도 이러한 보안 위험성을 알리고 드론을 안전하게 조종할 수 있도록 안내해야 한다. 드론의 강제 탈취로 인해 드론 소유자의 의도와 다르게 조정되어 불의의 사고로 이어지는 위험성을 사전에 방지하기 위한 최소한의 제도 개선이 필요하다.

ACKNOWLEDGMENT

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 ICT유망기술개발지원사업의 일환으로 수행하였음. [1711045083, 드론 비행 탐지 및 통제를 위한 지향성 안테나 기반 이동형 드론 분석기 개발]

참고문헌

[1] Platum, “사생활 침해, 안전성... 드론 대중화를 위해 고민해야 할 부분”, 2015.11.05.
[2] 백종현, IoT 융합 서비스 보안위협 및 대응 기술 동향, 2015.08.12
[3] 손충호, 심재범, 정일안, “드론의 보안 취약점 분석 및 대응방안 연구”, 2016.06.23.
[4] 조선일보, 좋은 드론, 나쁜 드론, 이상한 드론, 2017.04.22.
[5] Gartner, http://www.gartner.com/newsroom
[6] 구태언, 보안의 양면성을 가진 드론, 2015.09.18.
[7] Parrot Mambo, https://www.parrot.com/us/minidrones/parrot-Mambo
[8] Parrot Bebop2, https://www.parrot.com/us/Drones/Parrot-Bebop-2
[9] Cain, http://www.oxid.it
[10] Wireshark, http://wireshark.org
[11] NMAP, http://nmap.org
[12] Ubertooth, http://ubertooth.sourceforge.net
[13] NodeJS, https://nodejs.org
[14] Parrot Developer, http://developer.parrot.com
[15] ARSDK-XML, https://github.com/Parrot-Developers/arsdk-xml
[16] BebopSample, https://github.com/Parrot-Developers/Samples/tree/master/Unix/BebopSample
[17] seL4, https://sel4.systems