Master's Thesis
석사 학위논문

Vulnerability analysis of the Mavlink protocol for Unmanned Aerial Vehicles

Young-Min Kwon (권 영 민 權 寧 珉)

Department of
Information and Communication Engineering

DGIST

2018

Master's Thesis
석사 학위논문

Vulnerability analysis of the Mavlink protocol for Unmanned Aerial Vehicles

Young-Min Kwon (권 영 민 權 寧 珉)

Department of
Information and Communication Engineering

DGIST

2018

Vulnerability analysis of the Mavlink protocol for Unmanned Aerial Vehicles

Advisor: Professor Kyung-Joon Park
Co-advisor: Professor Min-gyu Cho

By

Young-Min Kwon
Department of Information and Communication Engineering
DGIST

A thesis submitted to the faculty of DGIST in partial fulfillment of the requirements for the degree of Master of Science in the Department of Information and Communication Engineering. The study was conducted in accordance with Code of Research Ethics[1]

Nov. 23. 2017

Approved by

Professor Kyung-Joon Park (Signature)
(Advisor)

Professor Min-gyu Cho (Signature)
(Co-Advisor)

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of DGIST, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.

Vulnerability analysis of the Mavlink protocol for Unmanned Aerial Vehicles

Young-Min Kwon

Accepted in partial fulfillment of the requirements for the degree of Master of Science.

Nov. 23. 2017

Head of Committee _____ (Signature)
Prof. Kyung-Joon Park
Committee Member _____ (Signature)
Prof. Min-gyu Cho
Committee Member _____ (Signature)
Prof. Jihwan Choi

MS/IC
201622002

권영민. Young-Min Kwon. Vulnerability analysis of the Mavlink protocol for Unmanned Aerial Vehicles. Department of Information and Communication Engineering. 2018. 26p. Advisors Prof. Kyung-Joon Park, Co-Advisors Prof. Min-gyu Cho

ABSTRACT

Recently, interest in the Unmanned Aerial Vehicle (UAV) has increased, and unmanned aircraft are utilized in various fields. Especially UAVs are used for rescue systems, disaster detection, and military purposes, as well as for leisure and commercial purposes. However, since UAVs are increasingly used not only for positive purposes but also for negative ones, it is necessary to detect and neutralize malicious drones.

In this paper, we proposed a method of controlling UAVs and analyzed its operation structure, the MAVLink protocol which is a communication protocol of UAVs. We also experimented with ICMP flooding and packet injection attacks which disables UAVs by exploiting the vulnerability of the MAVLink protocol. Especially, we exploited the vulnerability of the MAVLink waypoint protocol to perform an experiment to disable a UAV executing a mission. As a result of the experiment, we confirmed that the attacked UAV was stopped and the mission disabled.

Keywords : UAV, UAS, Drones, MAVLink, Network attack, DoS, Packet injection

Contents

List of tables

List of figures

I. INTRODUCTION

Recently, the term cyber-physical systems (CPS) has gained great interest and substantial research on it has been conducted [1, 2]. Unmanned Aerial Vehicles (UAV), an application of CPS, have been widely used around the world for the last decade. Especially, they are used in various fields such as rescue systems [3], disaster monitoring [4, 5], commercial use, military mission and so on.

An example of a commercial service using UAVs is Amazon's project Prime-Air, which was released in 2015 [6]. This system aims to design a future delivery service using UAVs. Since then, various services utilizing UAVs such as Fleetlight [7] and Matternet [8] have been released, as shown in Figure 1.1. In this way, services using UAVs are mainly performed in environments that are controlled over networks. Controlling the UAV over a network allows the UAV to perform its mission by completing the mission without user control. Figure 1.2 shows a UAV system controlled over a network.

However, UAVs are not always used for positive purposes. They can be abused for the purpose of crime, such as drug smuggling into prisons, and bombings and other types of terrorism. Especially, terrorism, is especially frightening because it can take lives. Therefore, malicious UAVs should be detected and disabled. In this paper, we analyze a UAV system controlled by a network and verify a method of disabling the UAV by exploiting the vulnerability of MAVlink, a communication protocol used for UAVs.

The rest of this paper is organized as follows. In Section II, we provide background information on drone controls, the MAVLink protocol, and network attack methods. In Section III, we summarize existing work on disabling UAVs. In Section IV, we introduce the proposed method to disable a UAV. The experimental environment and the experiment scenarios are presented in Section V. Finally, Section VI concludes this paper.

Figure 1.1 Fleetlight and Matternet Service.

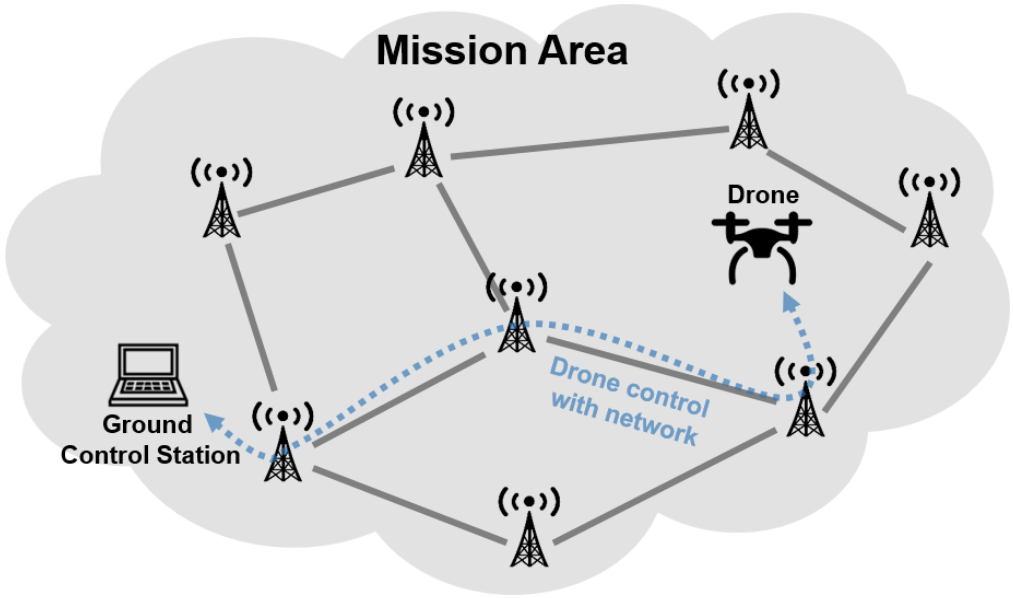

Figure 2.2 UAV system controlled over network.

II. BACKGROUND

2.1 Drone control structure

There are two ways to control a UAV: using a controller and using a GCS (Ground Control Station). In a controller-based control, the user views the UAV directly or watches through a camera mounted on the UAV and controls it using the controller. The UAV and the controller are connected to a communication module, and the UAV is controlled by transmitting the controller's signal to the UAV in real time. Generally, the communication modules used are telemetry, Wi-Fi, ZigBee, and so on. On the other hand, GCS-based control uses a computer to connect the software and the UAV; GCS then performs mission commands uploaded by the user. GCS can monitor the status of the UAV by receiving information of various sensors mounted on the UAV such as current altitude, speed, map position, and current mission status. The controller-based method can control the UAV in real time, but using GCS enables stable flight as well as unassisted flight to complete autonomous missions. Figure 2.1 shows the structure of a general UAV control system.

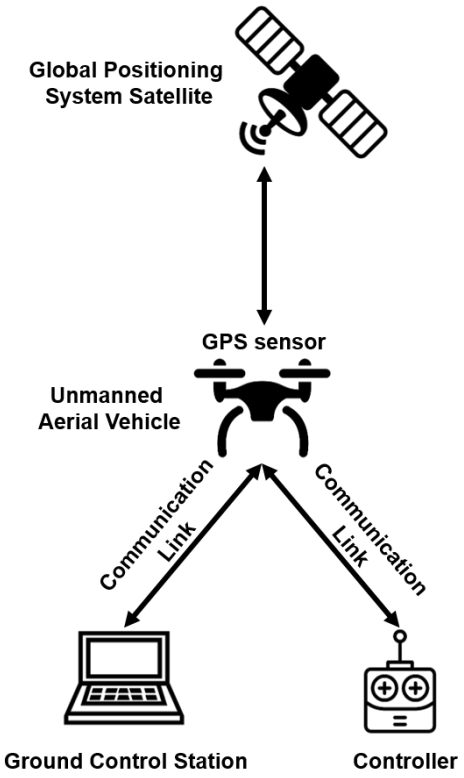

Figure 2.1. General drone control structure.

2.2 MAVLink Protocol

The MAVLink (Micro Air Vehicle communication) protocol is a message-based UAV communication protocol developed by Lorenz Meier in 2009 [9]. The MAVLink protocol is part of the current DroneCode project and is used by thousands of developers. It is also used in numerous Autopilot-based systems such as ArdupilotMega, pxIMU Autopilot, and SLUGS Autopilot [10]. MAVLink packets are bidirectionally transferred between UAV and GCS as header-based messages. The GCS sends mission commands to the UAV, and the UAV transmits state information including the sensor value, and current position to the GCS. Figure. 2.2 shows the message structure of the MAVLink protocol and Table 2.1 shows the meaning of the MAVLink frame [9].

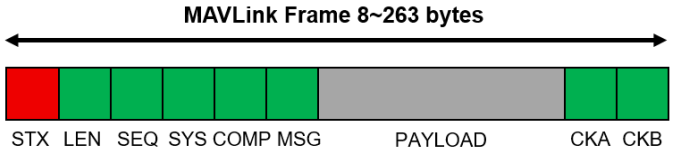

Figure 2.2. MAVLink protocol data frame structure [9].

Table 2.1. Meaning of the MAVLink frame [9].

Byte Index	Content	Value	Explanation
0	Packet Start Sign (STX)	0xFE	Indicates start of a new packet
1	Payload Length (LEN)	0-255	Indicates length of the following payload
2	Packet sequence (SEQ)	0-255	Packet transfer sequence information for detecting packet loss
3	System ID (SYS)	1-255	ID of the sending system; Allows to identify multiple platforms on the same network
4	Component ID (COMP)	0-255	ID of the sending component; Allows to identify multiple components on the same platform
5	Message ID (MSG)	0-255	ID of the message; Define what payload means, and how to decode it
6 to (n+6)	Data (Payload)	0-255 (bytes)	Data of message; depends on the message ID
(n+7) to (n+8)	Checksum (CKA and CKB)	ITU X.25/SAE AS-4 hash of bytes 1 to (n+6); It includes MAVLINK_CRC_EXTRA parameter computed from message fields	

Since the MAVLink message is a header-based protocol, it checks the first frame of the data packet and classifies the message. Therefore, it checks the STX value which is the initial frame and recognizes whether it is a MAVLink packet. In order to improve transfer speed and efficiency, the MAVLink message does not perform encryption [9]. When a message is encrypted because the value of the header of the packet changes, the system does not recognize is as a MAVLink packet. Also it takes additional time to decrypt the data. Therefore, since the message cannot be encrypted, there can be a security vulnerability.

2.3 Network Attack

Network attacks violate the confidentiality, integrity and availability of the system. Confidentiality allows information on the system only to authorized users. If confidentiality is violated, it is possible to eavesdrop on information and spoof the system. Integrity means the original information and signals transmitted, stored, and converted are maintained and not changed afterwards. Violation of integrity allows attacks such as message injection, replay attack, and so on. Availability allows the system to function for the time required by the user. In terms of maintenance, service must not be interrupted; performance must be maintained. Also, in terms of access to the system, the service must be accessible whenever the user needs it. Denial of service attacks are possible if availability is violated.

2.3.1 Man-In-The-Middle

MITM is an attack that violates the confidentiality or integrity of the system [11, 12]. As can be seen from the name, the attacker is located in the middle of the hosts and sniffs information [13]. The attacker can cause hosts to communicate information to the attacker. This is possible because system allows host to set the destination address to the attacker's address for ARP

poisoning. When MITM is applied to the UAV system, it is possible to eavesdrop on all of information transmitted between the UAV and GCS.

2.3.2 Eavesdropping

Eavesdropping is an attack that violates the confidentiality of the system; it means that an attacker steals and listens to information of other users. If an MITM attack succeeds, eavesdropping can be enabled [13]. As a method to protect the system from eavesdropping, it is necessary to encrypt the message.

2.3.3 Denial-of-Service

Denial-of-Service (DoS) attacks violate availability, monopolizing the resources of the system; using both DoS and MITM, it is possible to prevent other users from using system services [14]. In case of a DoS attack on a UAV system, control message, sensor information, and mission information are not correctly transmitted. Therefore, not only is the UAV not maintained in the stable state, but also the mission execution can not be performed correctly.

2.3.4 Potential threats to UAV systems

In the UAV system, it is possible to have different vulnerabilities for each component of the system. Therefore, the potential threats that may occur for each component may differ. The threats that can occur for each component of the UAV system are classified by the security objective [15, 16, 17, 18]. Table 2.2 shows the potential threats that may occur for each component of the UAV system.

Table 2.2. Potential threats on UAV systems.

Security objective	System objective	Attack method
Confidentiality	GCS	Virus
		Malware
		Keyloggers
		Trojans
	UAV	Hijacking
	Communication Link	Eavesdropping
		Man-In-The-Middle
Integrity	Communication Link	Packet injection
		Replay attack
		Man-In-The-Middle
		Message deletion
Availability	GCS	Denial of Service
	UAV	Fuzzing
	Communication Link	Jamming
		Flooding
		Buffer overflow

III. RELATED WORK

One way to disable a UAV is to use a sensor and hardware attack on the UAV, or a network attack. Sensor and hardware attacks make use of UAV sensor vulnerabilities to disable the UAV. In general, communication link jamming and GPS spoofing are used for sensor attacks in UAV systems. Jamming prevents the communication link between the UAV and the GCS or the controller from operating correctly as shown in Figure 2.1, so that the control message of the UAV cannot be transmitted. In the structure of the UAV system shown in Figure 2.1, GPS spoofing is a scheme utilizing the vulnerability of the communication between the GPS satellite and the UAV GPS sensor. A GPS spoofing attack is used to trick the UAV by broadcasting a fake GPS signal [10, 16]. In the case of a real GPS signal, the distance between the satellite and the sensor is long, so the GPS signal power can be weakened. Thus, it is possible to transmit fake GPS information to the UAV by generating GPS signals near the UAV. In [19], the authors studied a GPS spoofing attack that successfully attacked the GPS receiver.

In [11], the authors conducted research to disable a UAV by attacking access point in Wi-Fi networks. In this research, the authors used the vulnerability of wired equivalent privacy (WEP), which is one of the WiFi security protocols. WEP encryption has a vulnerability that makes it possible to crack the pre-shared key by collecting a certain amount of data. In particular, using the password crack tool aircrack-ng, it is easy to crack the pre-shared key value in WEP encryption. Using aircrack-ng, the authors disabled the UAV by sending de-authentication packets to the UAV.

In [20], the authors conducted an experiment to disable a UAV using a Man-in-the-middle attack. In this system, the authors used Zigbee API mode, which can send broadcast packets to UAV networks. The broadcast packets collect the initial vector values, which are used to crack the WEP. As in [11], the authors used the vulnerability of WEP to hack the UAV.

In [21], a method to hijack a UAV using a vulnerability of the MAVLink protocol was proposed. When using the telemetry module to control the UAV via MAVLink, it is necessary to enter the NetID to connect to the UAV. Therefore, if the NetID is known, it is easy to hijack the UAV. Using this, the authors of [22] executed an attack by using an antenna with the same NetID to repeatedly send malicious MAVLink packets.

In [23, 24], the authors hijacked a UAV using a vulnerability of the AR drone. In particular, in [23], the authors used port scanning of the FTP port, and then sent a malicious code to the UAV to access the UAV's private pictures and information without permission. Also, in [24], the authors performed an attack using an AR drone's telnet port vulnerability to re-install the shell script and restart the AR drone. In this way, they easily stole the authority of the AR drone.

IV. PROPOSED METHOD

In this paper, we disabled a UAV by exploiting a vulnerability of the MAVLink protocol. We exploited a vulnerability in which the MAVLink message was not encrypted and was injected after sniffing the UAV network packets. We assumed a system in which the UAV and GCS were connected via a network and the attacker had already hacked into the network.

To sniff UAV-GCS packets, it is necessary to know the network information of the UAV and GCS. Therefore, we used Cain & Abel to obtain the network information of UAV and GCS. Also, we developed a Jpcap-based monitoring tool to eavesdrop packets on the UAV network. Using the packet monitoring tool, we were able to analyze the information transferred between the UAV and the GCS in real time.

There are 160 kinds of common MAVLink packets; these packets send UAV state information or GCS commands in the MAVLink payload. By analyzing the packets to be transmitted, it is possible to identify whether the UAV is currently in flight, the state of the battery, what mission is being executed, and so on. Based on the obtained information, we can investigate the real-time state of the UAV and disable the UAV by using network attack and packet injection.

4.1 Cain & Abel

In order to decide on an attack target, it is necessary to have information about the hosts connected to the network. Using Cain & Abel [25] as a network sniffing tool operating on Windows OS, we can obtain information on the hosts connected to the network. We used Cain & Abel to learn the network IP address of the UAV and the GCS. Also, we obtained the GCS and UAV packets by using an ARP-poisoning attack, which sends fake ARP information to the host and causes the packet to be forwarded to the attacker. Therefore, in UAV networks, packets of UAV and GCS can be transmitted to an attacker.

4.2 Jpcap

Jpcap [26] is a Java-based library that captures network packets. Using Jpcap to monitor the state of the UAV, in this research we developed a packet capture tool. Figure 4.1 shows the developed program. As can be seen in Figure 4.1, the program shows the network interface, source ip address, destination ip address and payload. The payload indicates the type of MAVLink data. Making it possible to check the Message_ID of the MAVLink data. Using this program, we can estimate the state of the UAV in real time. For example, it is possible to confirm the MISSION_SET_CURRENT packet and determine what mission is currently being executed and whether or not the UAV is in flight. Therefore, we can know when to attack the UAV by monitoring the state information of UAV.

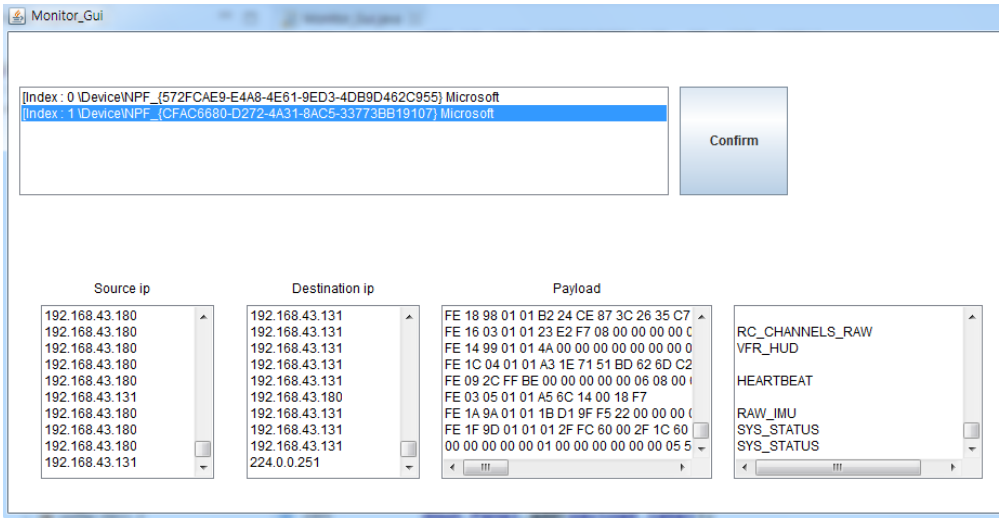

Figure 4.1. Monitoring program developed using Jpcap library.

4.3 Packet Sender

We used a Packet Sender [27] to inject attack packets into the UAV. This program can send network packets such as UDP and TCP; system runs on Linux, Windows, and MAC OS. Using this program, it is possible to transfer packets by changing to the payload desired by the user. Also, because it is a familiar GUI design, it is easy to use.

V. EXPERIMENTS

5.1 Testbed Configuration

In order to perform experiments in the UAV network, we constructed the testbed shown in Figure 5.1. We installed hostapd [28] in raspberry-pi3 to use the wireless access point. We constructed the environment so that UAV and GCS are connected using this access point. The UAV used for the experiment is a 3DR X8 + drone. Since this drone uses pixhawk, it can be controlled using the MAVLink protocol. In order to allow the drone to connect to the access point, we used raspberry-pi3, which included installing mavproxy [29]. The GCS used for the experiment is mission planner [30].

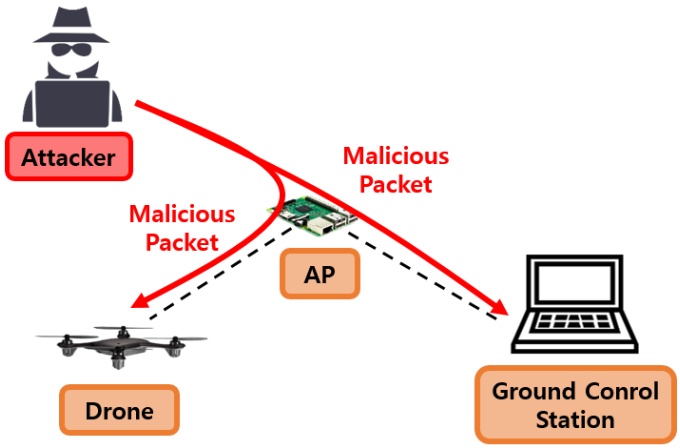

Figure 5.1. Testbed configuration with AP, GCS, and drone.

Figure 5.2. 3DR X8+ drone used for experiments.

Figure 5.3. Mission planner used for experiments.

5.2 ICMP flooding attack

Internet Control Message Protocol (ICMP) checks the connection status of the hosts in the network and reports when there is a problem with packet transfer. Using the ping command with Windows command or Linux kernel, an ICMP message can be sent. When sending an ICMP message, the sender will send an ICMP request packet to the receiver. The receiver that has successfully received the request message will respond to the sender. If the sender sends a large number of request messages, the receiver will be too overloaded to check and send replies. In this way, the ICMP flooding attack overloads the target system and invalidates the service.

In an environment connected to an access point, we experimented with the effect of an ICMP flooding attack on a UAV. First, when the attacker sends ICMP request packets to the GCS and the UAV at 7Mbps. Figure 5.4 shows the change in the inter-reception time of sensor values when sending ICMP packets to the UAV. In this experiment, we selected pitch values for the UAV. The normal case is shown in Figure 5.4; it is confirmed that the inter-reception time does not greatly deviate from the average time of 0.24, but that this value changes greatly

in the case of ICMP attack. In the normal case, the variance of the inter-reception time was measured at about 0.238×10^{-3}; in the case of ICMP attack, the variance of the inter-reception time was measured at about 8.4×10^{-3}. The variance of the inter-reception time during the ICMP attack is about 35 times larger than that of the normal case. Figure 5.5 shows the change in the inter-reception time of pitch values when sending ICMP packets to the GCS. In this figure, the variance of the inter-reception time in the normal case was measured at about 0.238×10^{-3}; in the case of ICMP attack, the variance of the inter-reception time was measured about 2.42×10^{-3}. The variance of the inter-reception time for the ICMP attack is about 10 times larger than that of the normal case. In this experiment, we can confirm that the variance of the packet inter-reception time is larger for an ICMP flooding attack on the UAV for such an attack on the GCS.

We also conducted an experimental ICMP flooding attack on a UAV that was executing a mission. In this experiment, we confirmed that the UAV's sensor values were not transmitted well, and the mission commands delivered by the GCS were also not transferred properly. A heartbeat message is sent between the GCS and the UAV in one second period to maintain the connection. If the heartbeat message is not received for more than 3 seconds, the UAV will operate in failsafe mode. In this experiment, because of the ICMP flooding attack, the UAV can not have received a heartbeat message within 3 seconds. However, the UAV crashed without operating failsafe mode due to error in failsafe mode.

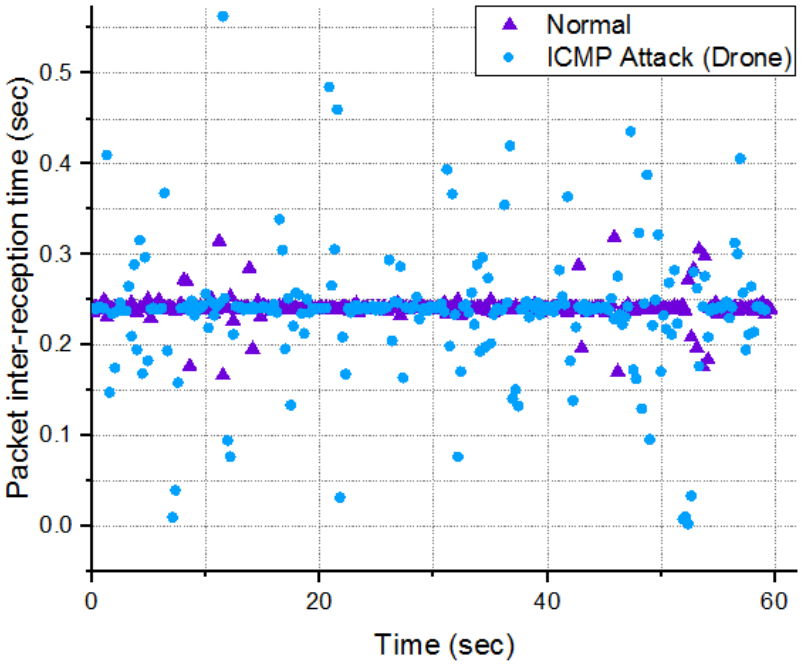

Figure 5.4. Packet inter-reception time in normal state and during ICMP attack on UAV.

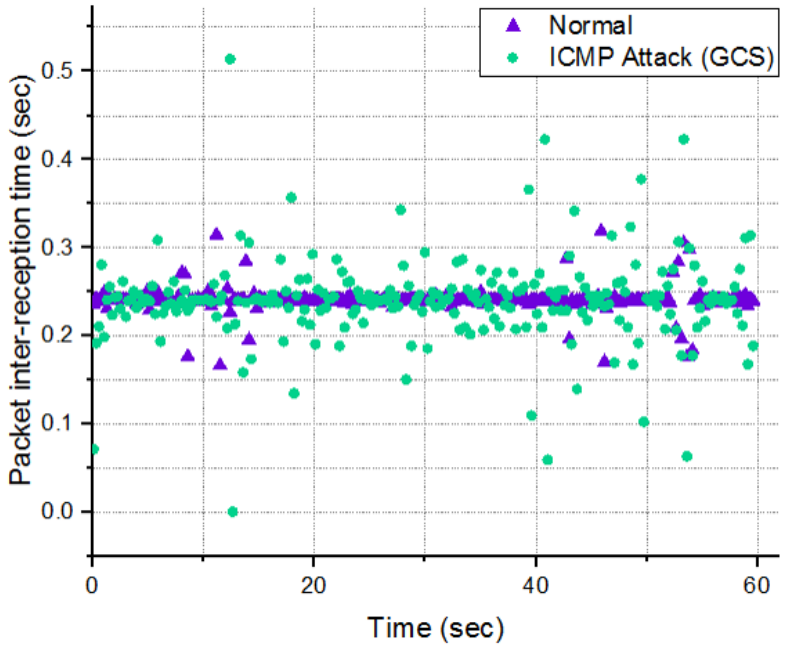

Figure 5.5. Packet inter-reception time in normal state and during ICMP attack on GCS.

5.3 Packet injection attack

When using GCS to control the UAV, UAV executes the mission commands sent by GCS. At this time, mission commands are executed based on the waypoint protocol [31] in the MAVLink protocol. Figure 5.6 shows the MAVLink waypoint protocol procedure.

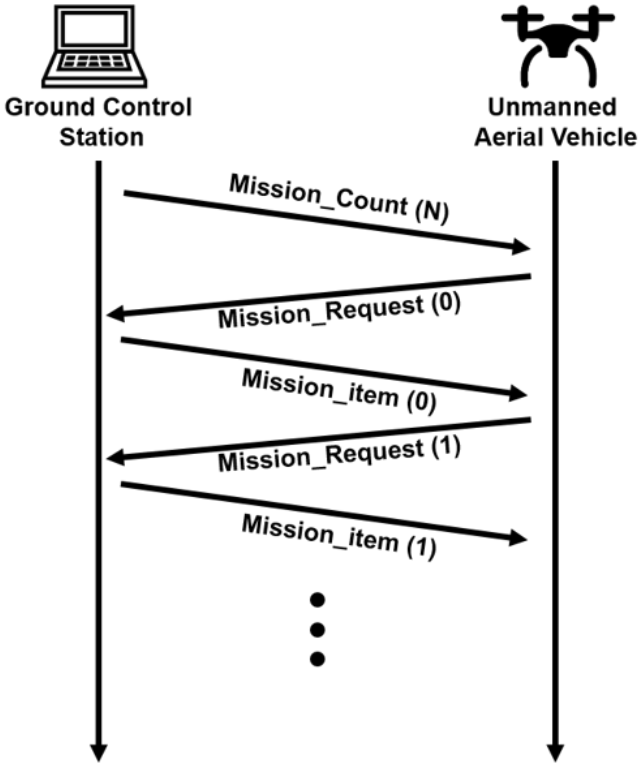

Figure 5.6. MAVLink waypoint protocol procedure.

When the user completes the mission commands setting, the GCS sends information on the total number of missions as a MISSION_COUNT (N) message. Upon receiving this message, the UAV requests the first mission information using the MISSION_REQUEST (0) message. In response to this message, the GCS sends the first mission information with a MISSION_ITEM (0) message. In this way, the GCS sends a total of N pieces of mission information to the UAV. Upon completion of the mission information transfer, the UAV transmits a MISSION_ACK message to the GCS to notify that the transmission is completed.

We exploited the vulnerability of the waypoint protocol and experimented with packet injection attack. When the GCS sends a MISSION_COUNT (N) packet, the UAV erases the stored mission information and prepares to receive new mission commands. Using these features, we constructed the experiment scenario as follows. Because the attacker had intruded into the network, the attacker was able to eavesdrop the information between GCS-UAV and obtain the mission information. After that, when the UAV executed the mission and started the flight, the attacker sent an eavesdropped MISSION_COUNT (N) packet to the UAV and initialized the mission information. UAV sends MISSION_REQUEST to GCS to request mission information, but GCS has already sent mission information so it will not transmit. Therefore, the UAV enters a standby state waiting for mission information.

We conducted experiments to transmit MISSION_COUNT (N) packets to the UAV executing its mission. As a result of the experiment, we confirmed that the UAV started to hover immediately after receiving the MISSION_COUNT (N) packet. This is because all of the mission information that the GCS had sent before had been deleted due to the MISSION_COUNT (N) packet that had been forwarded. Figure 5.7 shows the console screen of the UAV mavproxy that received the packet of MISSION_COUNT (N). In Figure 5.7, "not loading waypoint" appears on the console screen after receiving the MISSION_COUNT (N) packet while waypoint 2 is executing. In this state, the UAV continuously hovers unless the battery is exhausted or a new mission command is transmitted. If, when the UAV is in hovering state, an attacker injects a packet containing mission information, the UAV will execute the mission sent by the attacker.

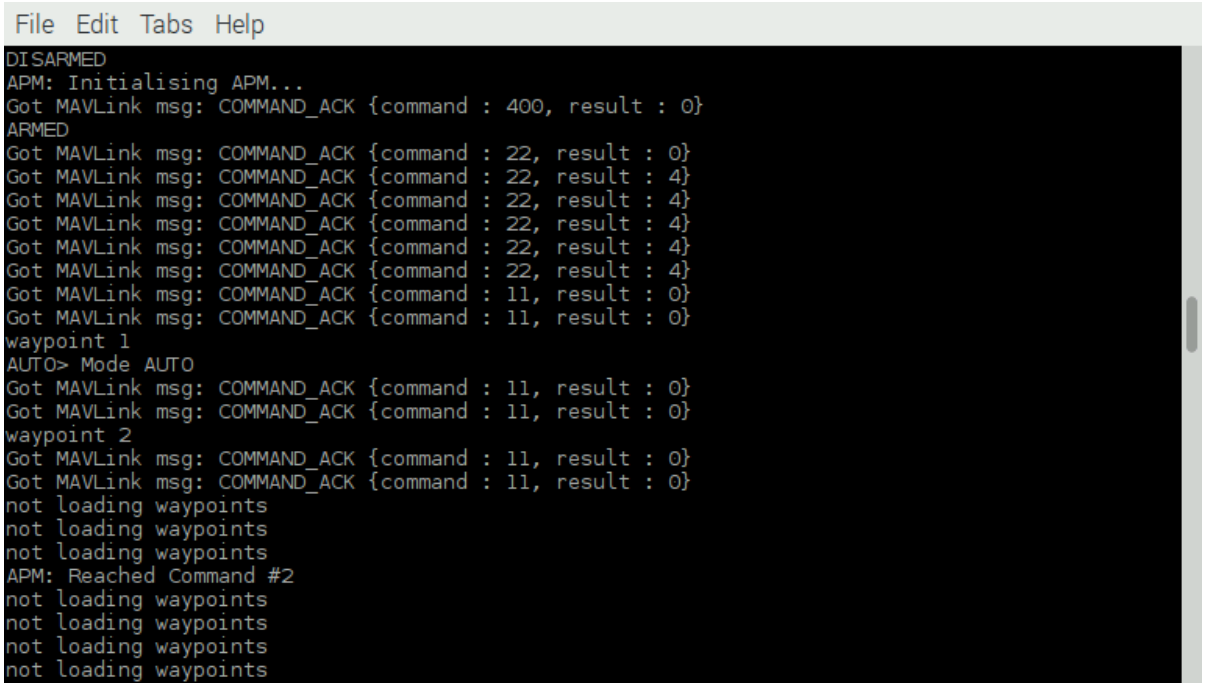

Figure 5.7. Mavproxy command screen.

5.4 Software In The Loop (SITL) Simulator

In the Software in the loop (SITL) simulator [32], the experiment scenario conducted in 5.2, 5.3 was performed in the same way. We used the mission planner as the GCS and connected the UAV to mavproxy in SITL.

First, we conducted experiments with SITL on how ICMP flooding affects the UAV. As in the previous experiment, it was confirmed that the packet inter-reception time greatly fluctuated. However, in a simulator different from those used in previous experiments, the UAV did not crash.

In addition, the same scenario as used for the packet injection experiment conducted previously was used with SITL. Figure 5.8 shows the packet injection experiment in SITL. Figure 5.9 shows the UAV mavproxy console screen after execution of SITL. As in the previous experiment, when the UAV receives the MISSION_COUNT (N) packet, we can confirm that "not loading waypoints" is displayed on the command screen.

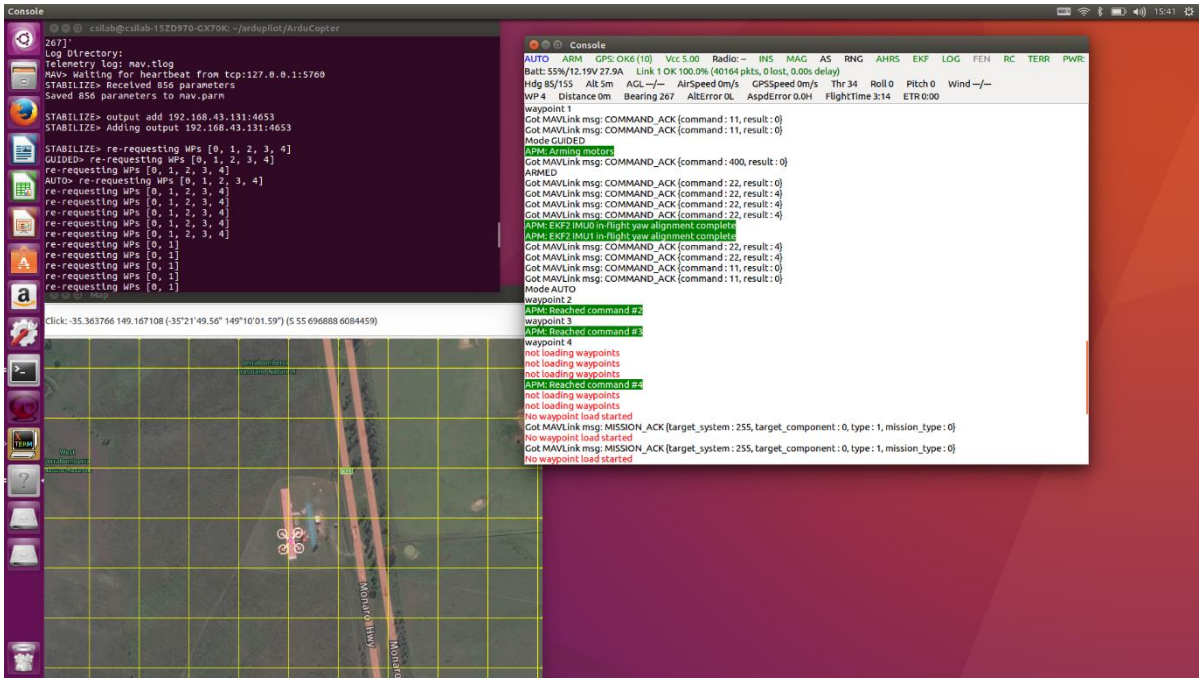

Figure 5.8. Experiment using SITL simulator.

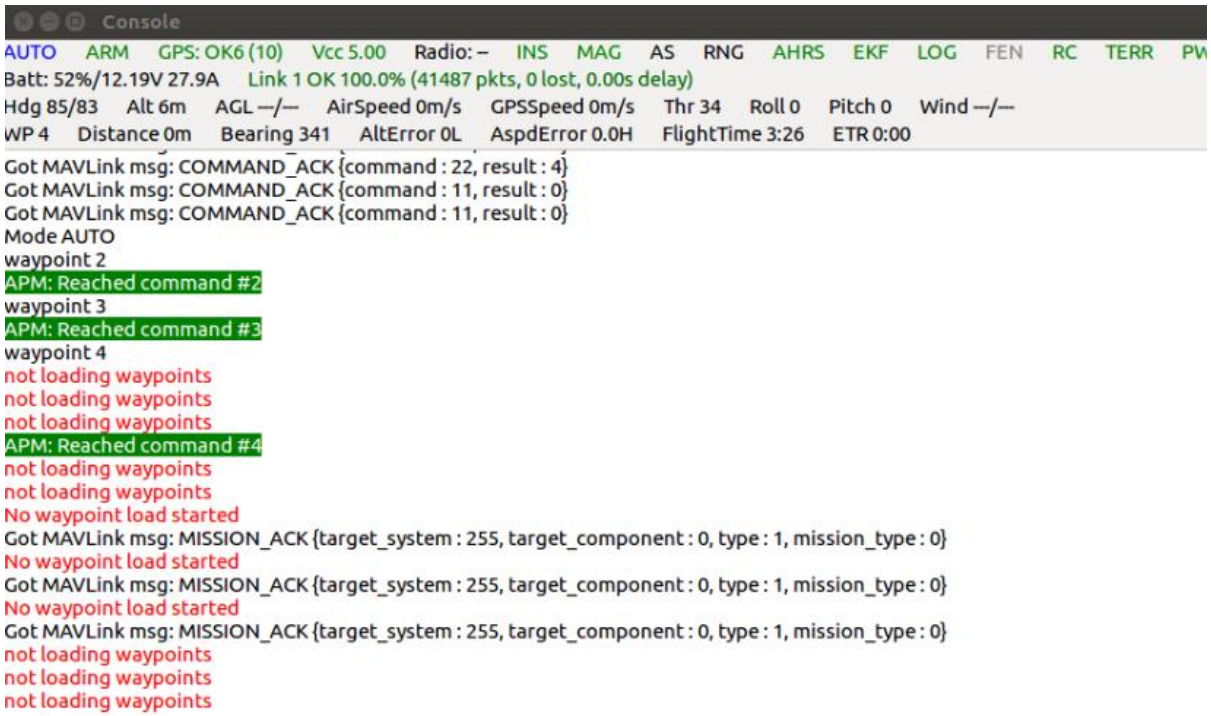

Figure 5.9. UAV mavproxy console screen executed in SITL simulator.

VI. CONCLUSIONS

In this paper, we exploited the vulnerability of the MAVLink protocol that is not encrypt messages, and experimented with a network attack to disable a UAV. We sniffed data between the UAV and the GCS and confirmed the real-time state of the UAV. In the attack, we used ICMP flooding and packet injection. In the case of the ICMP flooding attack experiment, we confirmed that the packet inter-reception time greatly fluctuated on average, and that this fluctuation can cause a fatal error in the UAV. In the case of packet injection experiments, we conducted an experiment to exploit the vulnerability of the waypoint protocol to send malicious packets to delete all mission information of the UAV. As a result of the experiment, we confirmed that the UAV, which was executing a mission, stopped and hovered immediately after receiving the malicious packet. We performed the same experiment in the simulator and verified that the UAV was disabled.

REFERENCES

[1] 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, 2014.

[2] K.-J. Park, R. Zheng, and X. Liu, “Cyber-physical systems: milestones and research challenges,” *Computer Communications*, vol. 36, 2012, pp. 1-7.

[3] S. Waharte, and N. Trigoni, “Supporting search and rescue operations with UAVs,” *International Conference on Emerging Security Technologies (EST)*, 2010, pp. 142-147.

[4] S. M. Adams, and C. J. Friedland, “A survey of unmanned aerial vehicle (UAV) usage for imagery collection in disaster research and management,” *In 9th International Workshop on Remote Sensing for Disaster Response*, 2011, pp.8.

[5] A. J. S. McGonigle, A. Aiuppa, G. Giudice, G. Tamburello, A. J. Hodson, and S. Gurrieri, “Unmanned aerial vehicle measurements of volcanic carbon dioxide fluxes,” *Geophysical research letters*, vol. 35(6), 2008.

[6] Amazon prime-air projects.

https://www.amazon.com/Amazon-Prime-Air/b?node=8037720011

[7] Fleetlights.

https://www.directline.com/fleetlights

[8] Matternet.

https://mttr.net

[9] MALink protocol.

http://qgroundcontrol.org/mavlink/start

[10] Domin., Karel., E. Marin, and I. Symeonidis, “Security Analysis of the Drone Communication Protocol: Fuzzing the MAVLink protocol,” *Proceedings of the 37th Symposium on Information Theory in the Benelux*, 2016, pp. 198-204.

[11] C. Rani, H. Modares, R. Sriram, D. Mikulski, and F. L. Lewis, “Security of unmanned aerial vehicle systems against cyber-physical attacks,” *The Journal of Defense Modeling and Simulation*, vol. 13(3), 2016, pp. 331-342.

[12] O. Alberto, and M. Valleri. "Man in the middle attacks," *Blackhat Conference Europe*, 2003.

[13] J. A. Marty, “Vulnerability analysis of the mavlink protocol for command and control of unmanned aircraft,” *MS. Thesis*, Air Force Institute of Technology, WPAFB, Ohio, United States, 2013, 142pages.

[14] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, “Inferring internet denial-of-service activity,” *ACM Transactions on Computer Systems (TOCS)*, vol. 24(2), 2006, pp. 115-139.

[15] M. D. Nguyen, N. Dong, and A. Roychoudhury, “Security Analysis of Unmanned Aircraft Systems,” National University of Singapore, 2017.

[16] H. Kim, and C. Steup. "The vulnerability of UAVs to cyber attacks-An approach to the risk assessment," *Cyber Conflict (CyCon)*, 2013, pp. 1-23.

[17] A. Y. Javaid, W. Sun, V. K. Devabhaktuni, and M. Alam, “Cyber security threat analysis and modeling of an unmanned aerial vehicle system,” *IEEE Conference on Technologies for Homeland Security (HST)*, 2012, pp.585-590.

[18] K. M. Mansfield, T. J. Eveleigh, T. H. Holzer, and S. Sarkani, “DoD comprehensive military unmanned aerial vehicle smart device ground control station threat model,” Defense Technical Information Center, 2015.

[19] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, “On the requirements for successful GPS spoofing attacks,” *In Proceedings of the 18th ACM conference on Computer and communications security*, 2011, pp. 75-86.

[20] N. M. Rodday, R. D. O. Schmidt, and A. Pras, “Exploring security vulnerabilities of unmanned aerial vehicles,” *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2016, pp. 993-994.

[21] Hijacking drones with a MAVLink exploit.

http://diydrones.com/profiles/blogs/hijacking-quadcopters-with-a-mavlink-exploit

[22] K. Highnam, K. Angstadt, K. Leach, W. Weimer, A. Paulos, and P. Hurley, “An Uncrewed Aerial Vehicle Attack Scenario and Trustworthy Repair Architecture,” *IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W)*, 2016, pp. 222-225.

[23] F. Samland, J. Fruth, M. Hildebrandt, T. Hoppe, and J. Dittmann, “AR. Drone: security threat analysis and exemplary attack to track persons,” *In Proceedings of the International Society for Optical Engineering (SPIE)*, vol. 8301, 2012.

[24] J. S. Pleban, R. Band, and R. Creutzburg, “Hacking and securing the AR. Drone 2.0 quadcopter: investigations for improving the security of a toy,” *In IS&T/SPIE Electronic Imaging*, vol. 9030, 2014, pp. 90300L1-12.

[25] Cain & abel.

http://www.oxid.it/cain.html

[26] Jpacap.

http://jpcap.gitspot.com/index.html

[27] Packet sender.

https://packetsender.com

[28] Hostapd.

https://w1.fi/hostapd

[29] Mavproxy.

http://ardupilot.github.io/MAVProxy/html/index.html

[30] Mission planner.

http://ardupilot.org/planner

[31] MAVLink waypoint protocol.

http://qgroundcontrol.org/mavlink/waypoint_protocol

[32] Software In the Loop simulator.

http://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html

요 약 문

MAVLink 프로토콜의 취약점 분석 및 무인기 무력화

최근 무인기에 대한 관심이 증가하며 무인기를 다양한 분야에서 활용하고 있다. 특히 인명 구조 시스템, 재난 감지, 군사적 목적뿐만 아니라 레저, 상업적인 목적으로 까지 사용되고 있다. 하지만 무인기를 긍정적인 용도로만 쓰이지 않고, 악용하는 경우도 증가하고 있으므로 악의적인 무인기를 탐지하여 무력화시켜야 할 필요가 있다.

본 논문에서는 무인기를 제어하는 방법과 동작 구조, 무인기의 통신 프로토콜인 MAVLink 프로토콜을 분석하고 MAVLink 프로토콜의 취약점을 이용하여 무인기를 무력화하는 패킷 injection 공격을 실험하였다. 특히 MAVLink waypoint 프로토콜의 취약점을 이용하여 mission 을 수행중인 무인기를 무력화하는 실험을 수행하였다. 실험 결과, 임무를 수행중인 무인기가 임무를 중단하여 가만히 호버링 하며 무력화 된 것을 확인하였고, 동일한 시나리오를 시뮬레이터를 통해 확인하였다.

핵심어 : UAV, UAS, Drones, MAVLink, Network attack, DoS, Packet injection