Security Vulnerability Analysis of Wi-Fi Connection Hijacking on the Linux-Based Robot Operating System for Drone Systems

Jinyeong Kang and Inwhee Joe(✉)

Department of Computer and Software, Hanyang University, Seoul, South Korea
{achieve365, iwjoe}@hanyang.ac.kr

Abstract. In this paper we describe the security vulnerabilities of the Erle-Copter quadcopters. Due to the fact that it is promoted as a toy with low acquisition costs, it may end up being used by many individuals which make it a target for harmful attacks. In addition, the video stream of the drone could be of interest for a potential attacker due to its ability of revealing confidential information. Therefore, we perform a security threat analysis on this particular family of drones. We set the focus mainly on obvious security vulnerabilities like the unencrypted Wi-Fi connection or the user management of the GNU/Linux operating system which runs on the drone. We show how the drone can be hacked in order to hijack the Erle-Copter. Our aim is to sensitize the end-user of Erle-Copters by describing the security vulnerabilities and to show how the Erle-Copter can be secured from unauthorized access. We provide instructions to secure the drones Wi-Fi connection and its operation with the official smartphone app or third party PC software.

1 Introduction

The Erle-Copter is a remote controlled quadrocopter from the Spain company Erle-Robotics. The drone is controlled by the user via an app which is currently available for iOS and Android. Unofficial software is available for bada, Symbian and Windows Phone. The second version of the Erle-Copter was unveiled at CES Las Vegas in 2012. This drone offers an upgraded camera and more sensors. It was first presented at the International Consumer Electronics Show in Las Vegas in the year 2015 (winning the CES Innovations Award for Electronic Gaming Hardware). The drone consists of plastic and foam material and measures about 30 cm. The connection to the drone is realized using Wi-Fi and the drone is then controllable from smartphones and tablets using iOS or Android operating systems, which allows the transmission of the video stream from both cameras. The drone features include object tracking and the compatibility for AR-Game applications.

Quadrocopter and other remote controlled drones gain more and more in popularity and are getting cheaper in price, for this reason the Erle-Copter which is a popular device at low cost is used in this investigation and tested for its security vulnerabilities. In this paper the focus will be set on how to hijack the Erle-Copter and how to secure them.

© Springer Nature Singapore Pte Ltd. 2019
J. H. Park et al. (Eds.): PDCAT 2018, CCIS 931, pp. 473–482, 2019.
https://doi.org/10.1007/978-981-13-5907-1_49

Even though the Erle-Copter is not a professional drone, its features turn it into an interesting object for research and gaming purposes. It is easily controlled through the mobile application where the user gains access to the various sensors. The available API and documentation allows to create own applications and projects. Bonus features include the AR.Drone academy [1] for SHARING flights, movies and pictures and augmented reality games (which are only supported under iOS). After powering up the drone, it will set up an open Wi-Fi Access Point.

This paper is structured as follows: Sect. 2 introduces the Erle-Copter and its technical aspects. In Sect. 3 the security vulnerabilities of the Erle-Copter using different attacking scenarios are described. A way of securing the drone is explained in Sect. 4. Section 5 will introduce possible future work and finally, in a summary is given.

2 Technical Specifications of the Erle-Copter

The Erle-Copter uses an OMAP 3630 CPU. This processor is based upon a 32 bit ARM Cortex A8 and runs with 1 GHz, it also uses a PowerVR SGX530 GPU with a frequency of 800 MHz on the System on a Chip (SoC) constructed by Texas Instruments. The drone memory includes 1-GB DDR2-RAM, running with 200 MHz and 128 MB NAND-Flash. The drone runs the Kernel Linux uclibc 2.6.32.9-gbb4d210 using uclibc as c-library. This allows a lightweight and fast reacting system. The quadrocopter supports the WLAN standards b/g/n for Video and flight data. For navigation, an Inertial Navigation System (INS) is used. The vertical QVGA (320 × 240) camera is used for ground speed measurement and video stream (Fig. 1).

Fig. 1. The Erle-Copter

A flight recorder is to be released by Erle-Copter, offering added GPS support, improved stability in flight and flights after set courses, a coming-home safe return function, review flights in 3D and sharing flights with in-flight data with other Erle-Copter Academy users [2]. If one part of the drone gets broken, every single part of the drone can also be bought from the Erle-Robotics store and the user can then replace

them individually. The app offers an easy to use interface, which makes the control of the drone relatively easy, compared to more professional RC controlled drones. Most of the controlling aspects are handled automatically, like the start- and landing phases, calibration and position holding. That way, the user only needs to issue 'high level' commands, and can totally forget about the complex handling required by other drones.

2.1 Sensors and Inertial Navigation System (INS)

The Erle-Copter is equipped with multiple sensors, which are placed under the hull. This includes a miniaturized Inertial Navigation System (INS) which measures in six degrees of freedom. This allows the drone to perform yaw, pitch and roll movement. Other sensors perform flight stabilization. The height control is realized through the usage of a pressure sensor and ultrasonic sensors.

2.2 Monitoring

Occupy-Wall-Street-protester Tim Pool configured one of the first generation Erle-Copter, which he described as Occucopter, for the purpose of monitoring the police by the citizens. As soon as the police notice the controller of the drone, they might interrupt it but the control automatically switches to the next person [3, 4]. Using a 3rd or 4th generation mobile communication standard (like UMTS/LTE) it is also possible to control such drones from a much larger distance than just the normal Wi-Fi range. A proof of concept has already been shown by Alcatel-Lucent Bell Labs Acceleration Program and Parrot R&D [5], piloting the Erle-Copter from a 1000 m (3280 ft) distance with a smartphone. This increase in control distance will increase the potential privacy threat.

3 Security Vulnerabilities of the Erle-Copter

We investigated the security vulnerabilities of the Erle-Copter running in the standard out of the box configuration using firmware version 2.2.9. The following interesting details regarding the security have been found.

3.1 Port Scan

After connecting the battery to the drone, it automatically boots up and is ready to use within seconds. It will check the motors and set up an unencrypted Wi-Fi hotspot named erle-robotics-frambuesa followed by a random number with 6 digits. After connecting to this Wi-Fi hotspot, a port scan using the Nmap software on the drones IP has been performed. This scan already shows two interesting open well-known system ports, port 21 (FTP) and port 23 (Telnet). The other ports are related to the drone operation. Table 1 will show the usage of all ports.

Table 1. Open ports running on the Erle-Copter

Port	Explanation
21(TCP)	FTP Server which serves video and image files recorded by the drone
23(TCP)	Telnet Server offering a root shell
5553(TCP)	VIDEO: The H264-720p frames of the camera are available here if the phone application is recording
5554(UCP)	NAVDATA: Current telemetry data (status, speed, rotor speed) is sent to the client here (15 cmds/s demomode, 200 cmds/s full/debug mode)
5555(TDP)	VIDEO: The video stream of the drone is available to clients here
5556(UDP)	ATCMD: The drone is controlled in the form of AT commands. These control commands are sent periodically to the drone (30 cmds/s)
5559(TCP)	CONTROL port: Some critical data (ROS to ROS), such as configurations are transferred here

3.2 Port 5559

The connection to the Control bridge automatically running is not password protected, allowing anonymous access to the/data subdirectory of the drone. After connecting an anonymous bridge to the ROS in the drone, it is automounted to/services/. This allows direct access to the connected services and possibly gain access to confidential data or place malicious files on it (Fig. 2).

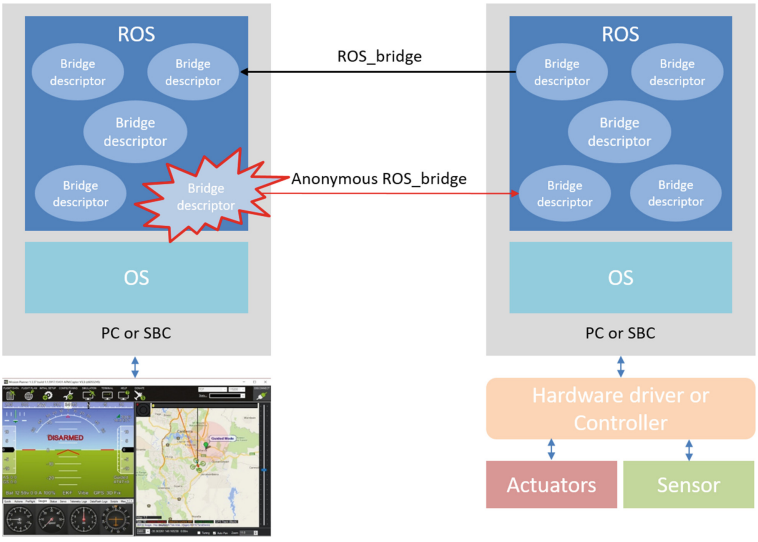

Fig. 2. Message communication of Erle-Copter using 5559 port

In the next step one could connect to the Telnet port which leads to a root shell. The root account is not password protected which gives an attacker free access to the entire drone operating system. This will allow an attacker to perform malicious actions like changing important config files or in the worst case scenario, wipe the filesystem to make the drone unusable. After investigating the filesystem, the following interesting shell scripts have been found:

3.3 Filesystem Backdoor

Using these files, an attacker has even more options to cause harm. This could include automatic moving of malicious files to a connected USB device every time a device is connected, changing the configuration of the reset-script. It also prevents initialization script execution by preventing drone pairing for unauthorized users. The shellscript/bin/reset_config.sh shows that, after using the reset button below the batteries, only the config file/data/config.ini will be reset. All other files remain untouched. This allows an attacker to always regain access to the drone, even after the user might use the reset button if he notice something is wrong with his drone (Table 2).

Table 2. Interesting shellscripts and Linux files on the Elre-Copter

File path	Explanation
/bin/check_update.sh	Update script
/bin/init_gpios.sh	Initialisation of GPIO ports used for connecting the navigation board to the SoC
/bin/pairing_setup.sh	Shell script for pairing using the Smartphone app
/bin/parallel-stream.sh	Camera streams
/bin/reset_config.sh	Reset config.ini while keeping total flighttime value
/bin/Wifi_setup.sh	Start of Wi-Fi connection and other services

3.4 Combination of Attacks

A combination of attacks - like the virus-copter [7] or the SkyJack-Hack, can be used by a malicious attacker to take over the drone entirely. This hack actively searches for hotspots generated by the Erle-Copter. This is done by “seeking out any wireless connections from MAC addresses owned by the Parrot company” (refer to [8]). If the hack finds a drone nearby, it disconnects the real user by deauthenticating him, then initiates a new connection to it while the drone is waiting for the new connection from the real owner. Additional files will be transferred to the drone to control it from the attacker’s machine (Fig. 3).

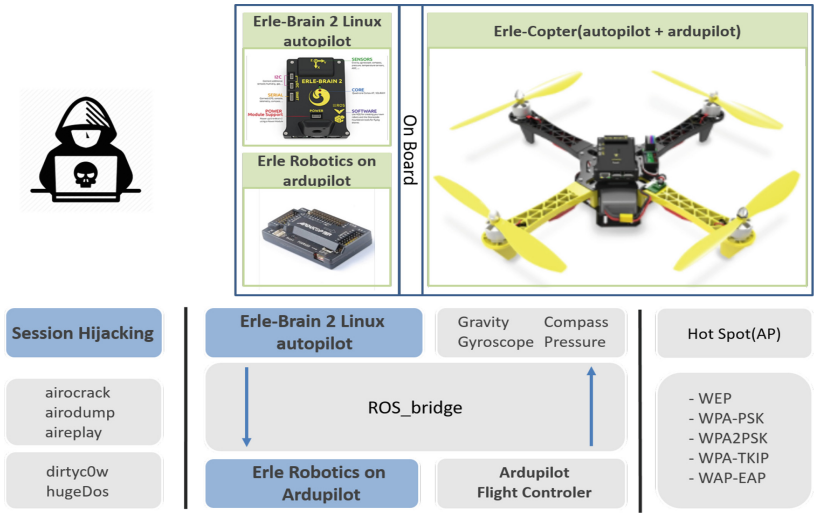

Fig. 3. Combination attacks scenario

4 Hacking Test

The objective is to capture the WPA/WPA2 authentication handshake and then use aircrack-ng to crack the pre-shared key. This can be done either actively or passively. "Actively" means you will accelerate the process by deauthenticating an existing wireless client. "Passively" means you simply wait for a wireless client to authenticate to the WPA/WPA2 network. The advantage of passive is that you don't actually need hijacking capability and thus the Linux version of aircrack-ng can be used (Fig. 4).

Now at this point, aircrack-ng will start attempting to crack the pre-shared key. Depending on the speed of your CPU and the size of the dictionary, this could take a long time, even days (Fig. 5).

Here is what successfully cracking the pre-shared key looks like (Figs. 6 and 7):
Here are the basic steps we went through:

1. Start the wireless interface in monitor mode on the Erle-Brain autopilot AP channel.
2. Start airodump-ng on Erle-Brain autopilot AP channel with filter for bssid to collect authentication handshake.
3. Use aireplay-ng to deauthenticate the wireless client.
4. Run aircrack-ng to crack the pre-shared key using the authentication handshake.
5. Penetration and Conquer of ROS control system using 5559 port.

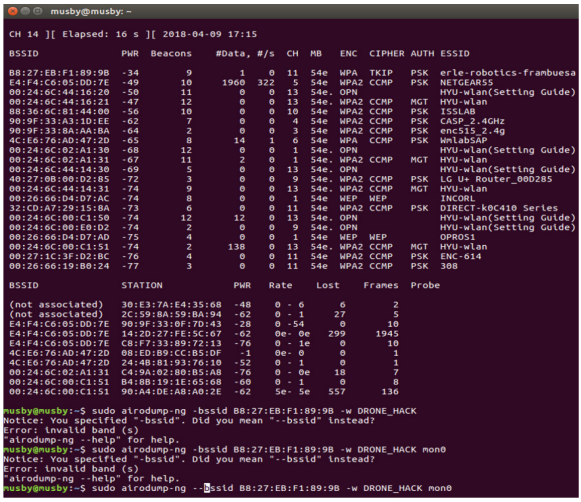

Fig. 4. Start airodump-ng to collect authentication handshake

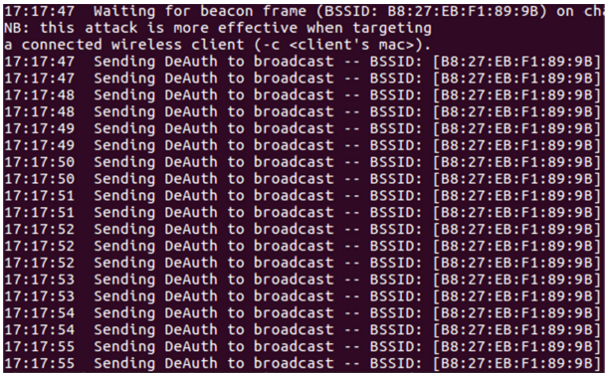

Fig. 5. Use aireplay-ng to deauthentication the wireless client

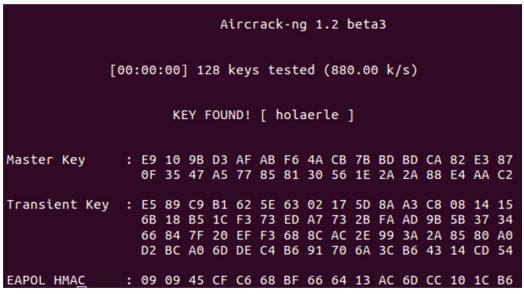

Fig. 6. Run aircrack-ng to crack the pre-shared key

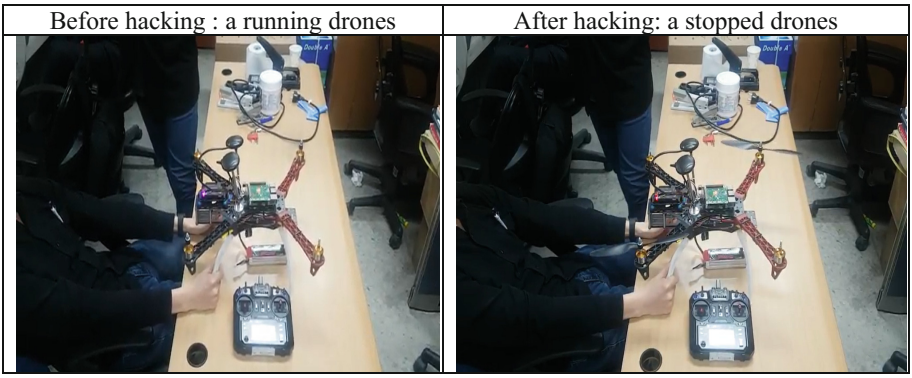

Fig. 7. Control System conquer of Erle-Copter through ROS-bridge between autopilot and ardupilot

5 Securing the Erle-Copter

Secure operation of drones in general is an important factor [9, 10]. The only security feature currently active is the MAC address filter. This system can easily be tricked by spoofing the MAC address of the owner's device when connecting to the drone. To get the needed address the attacker simply monitors and reads out the network traffic while the victim is operating the drone. We will show how one can secure the Erle-Copter in such a way that an attack is almost not possible. Our method follows a similar way as in [11] and [12]. This is realized by cross compiling the needed programs.

5.1 Securing the Wi-Fi Connection

As previous investigation of the first version of the Elre-Copter has shown, it is possible to eavesdrop the videosteam of the drone (refer to [6]). This is possible due to the encrypted Wi-Fi network hijacking set up by the drone on start. In this chapter, a way of securing this network connection is shown. Our approach to a more secure concept that can be used to control the drone is that the drone is just a client and the control device acts as the access point. We will show how the wpa_supplicant can be used on the Erle-Copter to let it connect to an encrypted Wi-Fi network. This tool is a WPA Supplicant for Linux, Mac OS X, and Windows systems and supports WPA and WPA2 (IEEE 802.1X [13]). To use the wpa_supplicant on the drone, we first have to cross-compile it to run on the ARM architecture.

5.2 Using the Secure Network

As stated before, we are going to use an external (secured) access point in order to use encrypted Wi-Fi. This hotspot can be generated by the control device as current mobile phones and tablets offer the possibility to create such hotspots. To keep the app running, the drone must maintain its original IP address (192.168.1.1). This is the only IP address that must remain free if the user does not want to use a third party app to

control the drone. The following method of starting the secure connection requires a third device to initiate the connection from the drone to the secured network. First, the drone should start up normally after connecting the battery. Now the secure hotspot can be started on the control device that will be later used to control the drone. The third device must now connect to the unsecure drone network which is called ardrone2_ followed by a random number. Still on the third device a telnet connection must be initiated to the drone. After issuing the commands:

```
$ wpa_passphrase $ESSID $PASSWORD > /etc/wpa_supplicant.conf
$ ifconfig ath0 $ADDRESS
$ iwconfig ath0 essid $ESSID
$ wpa_supplicant –B –Dwext –i ath0 –c /etc/wpa_supplicant.conf
```

via telnet. $ESSID must be replaced with the name of the secure network and the password must be set. $ADDRESS will be the new address the drone uses in the new network. To keep the original app running, the user should set this to 192.168.1.1. The drone will close the hotspot and connect to the control device on which the secured hotspot is running. By keeping the initiation of the connection to the secure network manual, we ensure the normal operation is still possible after reconnecting the battery, so even in case of a misconfiguration no special operations are required.

5.3 Benefits of Using an External Network

Since the new (secured) network can be connected to the internet, different new possibilities are possible. This includes the operation of the drone over the internet. The videostream could also be transmitted over the internet like a flying webcam which can be controlled from far away. The only problem is the reduced operating time of the drone due to the low battery time.

6 Conclusions

In this paper we have shown that the Erle-Copter can be secured in the way that it uses encryption instead of an unencrypted Wi-Fi to operate. Several attacking scenarios that a malicious attacker could use to disrupt normal operation have been shown. Using the option to use an external network connection secures the drone from such attacks. Additionally, the risk of showing the contents of a connected USB device is not given anymore.

Future work will include adding scripts to the drone or second device to remove the need for a third device to initiate the connection to the network. The security could be improved even more with a randomly generated passphrase which is given to the drone operator by touching a NFC device on the drone with the smartphone app. An app for tablets and smartphones that is allowing the drone to use a custom IP to operate would be of interest too. If this custom app is given, a way of automatically securing the drone by applying our method to it would be the next thing to develop.

Acknowledgment. This work was supported by the Technology development Program (S2521883) funded by the Ministry of SMEs and Startups (MSS, Korea).

References

1. Demgen, A.: AR.Drone-Academy: Soziales Netzwerk für Drohnen-Flieger verfügbar, August 2012. http://www.netzwelt.de/news/93209-ar-drone-academy-soziales-netzwerk-drohnen-flieger-verfuegbar.html
2. AR.Drone 2.0 flight-recorder. http://ardrone2.parrot.com/apps/flight-recorder/
3. Sharkey, N., Knuckey, S.: OWS Fights Back Against Police Surveillance by Launching "Occucopter" Citizen Drone, December 2011. http://www.alternet.org/story/153542/ows_fights_back_against_police_surveillance_by_launching_occucopter_citizen_drone
4. Wagstaff, K.: Occupy Wall Street's New Drone: 'The Occucopter', December 2011. http://techland.time.com/2011/12/21/occupy-wall-streets-new-drone-the-occucopter/
5. Méchaly, A.: One flew over the cornfield, October 2012. http://www2.alcatel-lucent.com/blogs/corporate/2012/10/one-flew-over-the-cornfield/
6. Samland, F., Fruth, J., Hildebrandt, M., Hoppe, T., Dittmann, J.: Erle-Copter: security threat analysis and exemplary attack to track persons. In: Casasent, D.P. (eds.) SPIE Proceedings, Intelligent Robots and Computer Vision XXIX: Algorithms and Techniques, Juha Röning; 83010G, vol. 8301 (2012)
7. Ackerman, E.: AR Drone that infects other drones with virus wins DroneGames, December 2012. http://spectrum.ieee.org/automaton/robotics/diy/ar-drone-that-infects-other-drones-with-virus-wins-dronegames
8. Kamkar, S.: SkyJack, December 2013. https://github.com/samyk/skyjack
9. Hacking Drones and the Dangers It Presents, July 2012. http://www.npr.org/2012/07/08/156459939/hacking-drones-and-the-dangers-it-presents
10. Drone hack explained: Professor details UAV hijacking, July 2012. http://rt.com/usa/news/texas-professor-drone-hacking-249/
11. node-cross-compiler. https://github.com/felixge/node-cross-compiler/
12. ardrone-wpa2. https://github.com/daraosn/ardrone-wpa2
13. IEEE_802.11. http://en.wikipedia.org/wiki/IEEE_802.11