Medicon Engineering Themes
Volume 5 Issue 4 October 2023
Article Type: Research Article
ISSN: 2834-7218

Security in UAV Ecosystem: An Implementation Perspective

Nikita Malik*, Menal Dahiya and Harsh Sinha

Department of Computer Applications, Maharaja Surajmal Institute, New Delhi, India

***Corresponding Author:** Nikita Malik, Department of Computer Applications, Maharaja Surajmal Institute, New Delhi, India.

Received: August 11, 2023; **Published:** October 02, 2023

Abstract

This paper focuses on implementational security measures to protect the UAV (Unmanned Aerial Vehicles) ecosystem from malicious adversaries reducing the surface of vulnerability against the malicious intent of these threat actors. To accomplish this goal, we have discussed five basic security measures. These measures include lightweight cryptographic function to encrypt firmware and other PCI (Peripheral Component Interconnect) buses, disabling non-utilized ports on the ground station, blocking inorganic traffic with an intrusion detection and prevention system, obscuring the Service Set Identifier (SSID) from broadcast scans, and implementing filter scrubs and dynamic whitelisting. These measures will be discussed in detail in this paper in parallel to the configurational implementation. The results of this paper were both challenging and rewarding. The proposed measures are helpful in improving the security of the UAV ecosystem and protecting it from different attacks, however, it still remains vulnerable to malicious actors. By preventing malicious requests generated by attackers, the lifespan and security of UAVs can be prolonged.

Keywords: Unmanned Aerial Vehicles; Internet of Drones; Security Measures; Intrusion Detection

Introduction

Unmanned Arial Vehicle (UAV) or Drones are rapidly becoming an integral part of modern infrastructure, as dependency on this emerging technological has increased over the decade in the fields of civilian health, surveillance-security, logistics supply, connectivity, smart agriculture, industry, safety, and the military-usage. Therefore, it is important to ensure the security of not just the drones themselves, but also all other connected technologies that make up the UAV's ecosystem. This expansive use of drones in the internet of things (IoT) has created a separate term as Internet of Drones (IoD) (Abualigah et al. 2021) has become increasingly popular due to their convenience and affordability (Yaacoub et al. 2020). However, with the increase of drone usage comes an increase in security concerns.

This research paper explores the security measures needed to protect drones and its ecosystem from malicious activities, such as hacking and data breaches. The focus is on the use of lightweight cryptographic functions to encrypt the firmware (Sinha et al. 2023), disabling non-utilized ports on the ground station, blocking inorganic traffic using SNORT, a network-based intrusion detection system, hiding the SSID (Service Set Identifier) from broadcast scans, and implementing filter scrubbing and dynamic white listing to protect the UAVs web application interface (WAi) (Mekdad et al. 2021) from Remote File Inclusion (RFI) and API abuse attacks (Sinha et al. 2023). The strengths and weaknesses of each security solution have been discussed a comprehensive overview of the best practices for drone security has been provided.

This research paper is an implementational extension to a previous paper by Sinha et al. (Sinha et al. 2023), and the structure of the paper has been defined in the Figure 1.

Figure 1: Structure of the Paper.

Related Work

The methodology used and the research gap identified in the related literature are tabulated in Table 1.

Paper Title & Year	***Methodology***	***Research Gap***
Sinha, H., Malik, N., Dahiya, Menal. (June 2023). Drone Ecosystem: Architecture for Configuring and Securing UAVs. Proceedings of Fourth International Conference on Computing, Communications, and Cyber-Security (IC4S 2022)	This paper proposes a novel architecture for configuring and securing unmanned aerial vehicles (UAVs).	The paper does not discuss the potential scalability and performance issues associated with the proposed architecture.
Tiwari, M., Kumar, R., Bharti, A., & Kishan, J. (2017). Intrusion detection system. International Journal of Technical Research and Applications	This paper presents a novel intrusion detection system that utilizes fuzzy logic and Artificial Neural Networks (ANNs) to detect malicious activities.	The paper does not discuss the potential false positives or false negatives that may arise from using the proposed system.
Keleman, L., Matić, D., Popović, M., & Kaštelan, I. (2019, September). Secure firmware update in embedded systems. In 2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin)	This paper proposes a secure firmware update approach for embedded systems, based on a combination of digital signatures and symmetric cryptography.	The paper does not discuss the potential risks posed by the proposed approach.
Wu, Y., Noonan, J. P., & Agaian, S. (2011). Shannon entropy based randomness measurement and test for image encryption	This paper proposes a randomness measurement and testing technique for image encryption, based on Shannon entropy.	The paper does not discuss the potential security risks posed by using the proposed technique.
Mekdad, Y., Aris, A., Babun, L., Fergougui, A. E., Conti, M., Lazzeretti, R., & Uluagac, A. S. (2021). A Survey on Security and Privacy Issues of UAVs	This paper provides a survey of the security and privacy issues associated with unmanned aerial vehicles (UAVs).	The paper does not discuss potential solutions to the identified security and privacy issues.
Abualigah, L., Diabat, A., Sumari, P., & Gandomi, A. H. (2021). Applications, deployments, and integration of internet of drones (iod): a review	This paper provides a comprehensive review of the applications, deployments, and integration of Internet of Drones (IoD).	The paper does not discuss the potential challenges that may arise in IoD applications.

Haider, S. K., Nauman, A., Jamshed, M. A., Jiang, A., Batool, S., & Kim, S. W. (2022). Internet of Drones: Routing Algorithms, Techniques, and Challenges	This paper provides an overview of the routing algorithms, techniques, and challenges associated with the Internet of Drones (IoD).	The paper does not discuss potential solutions to the identified challenges.
Lin, C., He, D., Kumar, N., Choo, K. K. R., Vinel, A., & Huang, X. (2018). Security and privacy for the internet of drones: Challenges and solutions	This paper provides an overview of the security and privacy challenges associated with the Internet of Drones (IoD) and proposes potential solutions.	The paper does not discuss the potential scalability and performance issues associated with the proposed solutions.
Yaacoub, J. P., Noura, H., Salman, O., & Chehab, A. (2020). Security analysis of drones systems: Attacks, limitations, and recommendations	This paper provides an overview of attacks and limitations associated with drone systems and proposes potential recommendations.	The paper does not discuss the potential effectiveness of the proposed recommendations in mitigating the identified attacks and limitations.
Dabrowski, A., Pianta, N., Klepp, T., Mulazzani, M., & Weippl, E. (2014, December). IMSI-catch me if you can: IMSI-catcher-catchers	This paper presents a novel approach for detecting IMSI-catchers.	The paper does not discuss the potential scalability and performance issues associated with the proposed approach.
Langley, A., Riddoch, A., Wilk, A., Vicente, A., Krasic, C., Zhang, D., ... & Shi, Z. (2017, August). The quic transport protocol: Design and internet-scale deployment	This paper presents a novel transport protocol called Quick UDP Internet Connections (QUIC).	The paper does not discuss the potential security risks posed by using the proposed protocol.
Roy, M., Ahsan, S., Kumar, G., & Vimal, A. (2020). Implementation of Quick UDP Internet Connections (QUIC) Protocol	This paper presents an implementation of the Quick UDP Internet Connections (QUIC) protocol.	The paper does not discuss the potential security risks posed by using the proposed protocol.

Table 1: Comparative overview of related literature.

Lightweight Cryptographic Function

While implementing a Light Weight Cryptographic Function (Lin et al. 2018) onto an Arduino-based drone running embedded Linux, we aimed to achieve Firmware Encryption to encrypt the entire firmware, onboard flight module, telemetry data, and other non-volatile memory storage areas of the UAV(s) like hard disks and other long-term storage areas (Sinha et al. 2023) to prevent an attacker from gaining critical information about the drone like the version of certain programs, libraries and Peripheral Component Interconnect (PCI) information or components information of the Main Remote Controller Board (onboard computer system) and carve out exploits for undetected zero-day vulnerabilities. This security feature will allow us to defend UAVs against reverse engineering attacks on the firmware by a malicious adversary if they gain physical access to a Missing-In-Action (MIA) drone.

There are three types of firmware encryption techniques- Symmetric, Asymmetric and Authenticated Encryption (i.e. AES-GCM). In our research, we found that using asymmetric encryption for firmware encryption on a drone can lead to boot time limitations. This is because asymmetric encryption algorithms require more computational power and time compared to symmetric encryption algorithms. As a result, using asymmetric encryption can cause delays during the boot process of a drone, which is not desirable in critical applications where fast boot times are essential. Therefore, we recommend using symmetric encryption techniques such as AES for firmware encryption on drones. Additionally, authenticated encryption techniques such as AES-GCM can provide both confidentiality and integrity of the firmware, making it a suitable choice for drone firmware encryption.

Exploiting Look Out

Once a malicious adversary obtains a UAV(s) firmware, they can use tools such as BinWalk to analyze the binary images (.bin) for embedded files and executable codes, or exploit component information using Linux utilities such as 'lspci' to display information about PCI buses in the Drone's system and devices connected to them.

Another method of exploiting UAV(s) firmware is by the use of binary diffing technique that involves comparing two binaries of different versions of the same software and using diffing tools/utilities like 'diff' to understand the new functions introduced or old removed in the new version of the firmware.

Whether a drone's firmware is encrypted or not can be determined by entropy calculation using the Shannon entropy formula as represented in equation (1) (Wu et al. 2011). Entropy is a measure of randomness or information density, which is expressed as a value between 0 and 1. A higher entropy value indicates a higher degree of randomness, with values near 1 being considered high entropy and values near 0 indicating less entropy. Encrypted data typically has a high entropy value close to 1.

$$H(X) = -\sum_{i=1}^{n} P(x_i)\log_b P(x_i) \quad \ldots (1)$$

Disabling Non-Utlized Ports on The Ground Station

The Ground Station (GS) serves as the command center UAVs, responsible for overseeing their operations. It is typically a ground-based computer system that runs specialized software known as Ground Station Control (GSC) software. This software can be installed on any Linux-based distribution or version of the Windows operating system. However, given the numerous vulnerabilities present in older versions of Windows, it is recommended that Windows 10 or higher be used. For instance, vulnerabilities such as CVE-2022-30190[1] & CVE-2020-0822[2] have been reported in previous versions. Therefore, selecting an appropriate operating system for the GS is essential for maintaining security and protecting the UAV ecosystem from potential attacks.

Securing GSC/GS relies on disabling all connection-oriented (TCP- Transmission Control Protocols) & connectionless protocols (UDP- User Datagram Protocols) (Langley et al. 2017) that are non-essential to UAV(s) communication, connection, and services such as FTP –21, SSH –22, SMTP –25, HTTP –80, POP3 –110, POP3 SSL –995, IMAP –143, IMAP SSL –993, SQL –1433, RDP –3389 (Sinha et al. 2023).

It is advisable to run GSC software on a Linux GS-dedicated computer system to avoid other preinstalled software having vulnerabilities and could potentially risk the UAV's security and integrity. Windows Remote Desktop Protocol (RDP), active on TCP port 3389, has historically been commonly vulnerable to various attack vectors, allowing hackers to breach into GSs and other UAV utility environments.

For our Linux GS, we used the 'apt-cache pkgnames' command to check for vulnerable and unnecessary preinstalled packages on our Linux GS and removed them using 'apt purge <package_name>', i.e. 'apt purge font-georgewilliams'.

Identifying open ports on Linux can be achieved by running the netstat utility to display various network-related information for active or open ports, connections more descriptively using 'netstat' with '-antp' flag, or else we can use the 'ss' (socket statistics), another Linux utility that dumps socket statistics information of the running Linux system.

In order to filter out TCP and UDP ports, one can use the 'ss -tl' and 'ss -ul' flags individually, or combine both flags using 'ss -tul'. However, if the objective is to identify actively listening ports and their associated service names, the command 'ss -tuln | grep LISTEN' can be used. This command effectively filters out actively listening ports and displays their corresponding service names. Such information can be particularly useful in identifying potential security threats or network performance issues. This method of port filtering can be implemented in various network monitoring and analysis tools.

In a Linux OS, the manual way to close an open port is very time-consuming and tedious, so a better way would be to disable the processes that the port is actively running or use 'sudo ss --kill state listening src :<port_number>' which will send a SOCK_DESTROY request to the kernel that will disable this port until otherwise, i.e. 'sudo ss --kill state listening src :1234'.

Blocking Inorganic Traffic Using IDS

For our Linux-based GS, we have been using the network-based IDS SNORT, which is equipped with a set of predefined rules that can identify and categorize malicious network activity and inorganic traffic (Sinha et al. 2023). It continuously monitors all active ports on the network, looking for packets that match against the predefined rules. In the event of a match, SNORT generates alerts to notify the network administrator of potential security threats before an attacker can cause damage to the network (Tiwari et al. 2017).

SNORT looks for attack patterns within network traffic by analyzing the packets' exchange. Large collections of related items that are of a certain type originating from single or multiple sources could indicate a denial-of-service (DOS) or distributed denial-of-service (DDOS) (Mekdad et al. 2021). SNORT looks for the exchange of a sequence of related packets in a certain pattern, which could indicate that a port scan is in progress using NMAP or any other network scanners (Tiwari et al. 2017).

Limited by resources, we installed SNORT NIDS on our Linux-based GS. SNORT NIDS (Network based Intrusion Detection System) consists of four main functions- data collection, feature selection, analysis, and action. It is typically installed on a separate computer on a network-connected device like a router, so that it can monitor the traffic entering and leaving a particular network segment.

After installing SNORT (preinstalled in Linux), we can customize the main SNORT configuration file to suit our needs. To do this, we can enter 'sudo gedit /etc/snort/snort.config' in the terminal. For testing purposes, we can use the default configuration settings and only add our HOME_NET to our network IP address range to 192.168.0.1/24, indicating a range of 1 to 254 addresses.

Additionally, we can use the default RULES or configure them to suit our UAV's ecosystem's requirements. After making these changes, we need to run a configuration check to ensure all settings are correct using 'sudo gedit /etc/snort/snort.conf'. Finally, we can run SNORT using 'sudo snort -A console -q -u snort -c /etc/snort/snort.config -i enp0s3' (*'enp0s3' is our interwork interface card*) to monitor the network for inorganic traffic and attack vectors.

To test the effectiveness of the SNORT IDS (Intrusion Detection System), we conducted a network scan using Nmap from an attacker's perspective. The SNORT IDS provided an alert output, which was captured in the image as shown in Figure 2.

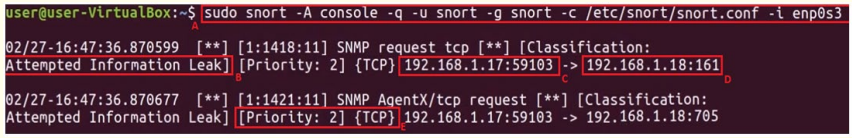

Figure 2: Alert output of SNORT.

A. Is the command used to run and enable (IDS) SNORT on network monitor mode.
B. Is the type of attempt on the network Information Leak, Bad-Unknown, Potentially Bad Traffic, Default-login-*Attempt*, etc.
C. Is the IP address of the Attacker.
D. Is the IP address of the Victim.
E. Type of Priority 1 (high) is the most severe and 4 (very low) is the least severe and type of port used like TCP and UDP (Roy et al. 2020) for us secure ports is (UPD 443).

Obfuscating SSID from Network Broadcast Scans

Disabling the broadcast SSID is a straightforward method to obfuscate or hide it from network broadcast scans. By disabling SSID broadcasts, detecting the network's SSID through tools such as Wireshark and airodump-ng becomes more difficult (Sinha et al. 2023). Wireless network cards used in network routers often run on Unix or Linux embedded systems (Keleman et al. 2019). To disable SSID broadcast on a Unix system, the first step is to identify the interface name of the wireless adapter. Once the interface name is determined, the wireless adapter can be connected to the network router using an FTP (File Transfer Protocol) connection to disable the SSID broadcast.

To disable SSID broadcast on a Unix system, we must first determine the interface name of the wireless adapter and connect it to the network router via an FTP connection. After establishing a connection with the router, we can then run the command 'iwconfig'. This will list all of the available wireless interfaces. To disable SSID broadcast on a Unix system, the first step is to determine the interface name of the wireless adapter and connect it to the network router via an FTP connection.

Once connected, the command 'iwconfig' can be run to list all available wireless interfaces. With the interface name identified, the 'iwpriv' command can be used to disable SSID broadcast with the syntax 'iwpriv <interface_name> set SSID_Hide=<1|0>', where <1|0> is either 1 to disable SSID broadcast or 0 to enable it. For instance, if the interface name is "wlan0", the command to disable SSID broadcast would be 'iwpriv wlan0 set SSID_Hide=1'.

To disable SSID broadcast on a Linux system, we first need to establish an FTP connection to the router using the 'ftp connect' command in the terminal. Once connected, we need to edit the wireless network configuration file by running the command 'sudo nano /etc/hostapd/hostapd.conf' and add the following line to the end of the file: 'ssid_broadcast=0'. After saving the changes, we need to restart the wireless network configuration file by running 'sudo service network-manager restart'. By doing so, we can successfully disable the SSID broadcast of our router.

However, this method also has drawbacks. As it could also prevent legitimate users from connecting to the UAV Ecosystems network. Additionally, it does not protect against active scanning, which can be used to detect hidden networks. Disabling probe response is another option for hiding the SSID from broadcast scans. However, while this method can protect the UAV Ecosystem against passive scans, the SSID will still be visible in the beacon frames. Therefore, further research will be conducted to develop more effective methods for hiding the SSID from broadcast scans. This will be a focus of our future work.

Implementing Filter Scrubs and Dynamic Whitelisting

Filter scrubs and dynamic whitelisting are techniques to protect UAV ecosystem's Web Applications Interface (Haider et al. 2022) and Application Program Interface (APIs) from malicious input parameters and file inclusions (Sinha et al. 2023) basically to prevent malicious adversaries to target vulnerabilities in web applications aiming to infect and uploading malware or backdoor exploit using Remote File Inclusion (RFI) and API abuse attack.

We protected our web applications and APIs (Haider et al. 2022) from malicious input parameters by applying input validation to check that the data received from a user is in the expected format and to reject any input that is not. This can also limit the length of input parameters to prevent Buffer Overflows, Command, or SQL (Structured Query Language) injection attacks (Mekdad et al. 2021).

To enhance the security of our system, we have implemented rate-limiting and request throttling measures. These measures help restrict the number of requests that can be sent within a specified timeframe and enable us to detect any suspicious patterns that may emerge from potential attacks. By doing so, we can reduce the likelihood of our system being overwhelmed or compromised by malicious requests.

Conclusion and Future Scope

This research paper has proposed a set of security measures to enhance the security, efficiency, and functionality of the IoD (Dab-

rowski et al. 2014). The proposed measures include firmware encryption, intrusion detection systems, disabling non-utilized ports, blocking inorganic traffic using IDS SNORT, obfuscating SSID from network broadcast scans, and implementing filter scrubs and dynamic whitelisting for web application interfaces.

While these measures can provide a strong level of security, there is still room for further research to explore additional security measures can be implemented to enhance the security of the UAV ecosystem. Furthermore, it is essential to study the potential implications of these security measures on the overall performance and efficiency of the system. Future research can also focus on developing more robust and sophisticated security solutions to effectively address the evolving security threats in the IoD. Future work will include the exploration of utilizing the IoD for secure cloud-based operations. Specifically, the focus will be on developing an efficient security framework for the cloud-based infrastructure of the IoD and UAV architecture. Additionally, further research will be conducted to explore options for hiding an SSID from broadcast scans.

To prevent passive scans from compromising the security of UAV ecosystems, disabling probe response can be an effective measure, However, it's important to note that the SSID will still be visible in the beacon frames. Moreover, we will look into ways of preventing active scanning, which detects hidden networks and prevents legitimate users from connecting to the UAV ecosystems' network. These measures can be implemented to ensure security and integrity of UAV ecosystem.

Note

1. https://www.cvedetails.com/cve/CVE-2022-30190/
2. https://www.cvedetails.com/cve/CVE-2020-0822/

References

1. Abualigah L., et al. "Applications, deployments, and integration of internet of drones (iod): a review". IEEE Sensors Journal (2021).
2. Dabrowski A., et al. "IMSI-catch me if you can: IMSI-catcher-catchers". In Proceedings of the 30th annual computer security applications Conference (2014): 246-255.
3. Haider SK., et al. "Internet of Drones: Routing Algorithms, Techniques, and Challenges". Mathematics 10.9 (2022): 1488.
4. Keleman L., et al. "Secure firmware update in embedded systems". In 2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin) IEEE (2019): 16-19.
5. Langley A., et al. "The quic transport protocol: Design and internet-scale deployment". In Proceedings of the conference of the ACM special interest group on data communication (2017): 183-196.
6. lin C., et al. "Security and privacy for the internet of drones: Challenges and solutions". IEEE Communications Magazine 56.1 (2018): 64-69.
7. Mekdad Y., et al. "A Survey on Security and Privacy Issues of UAVs". arXiv preprint arXiv:2109.14442 (2021).
8. Roy M., et al. "Implementation of Quick UDP Internet Connections (QUIC) Protocol". International Journal of Engineering and Computer Science 9.01 (2020): 24921-24924.
9. Sinha H., et al. "Drone Ecosystem: Architecture for Configuring and Securing UAVs". Proceedings of Fourth International Conference on Computing, Communications, and Cyber-Security (IC4S 2022). Springer, Singapore (2023).
10. Tiwari M., et al. "Intrusion detection system". International Journal of Technical Research and Applications 5.2 (2017): 38-4.
11. Wu Y, Noonan JP and Agaian S. "Shannon entropy based randomness measurement and test for image encryption". arXiv preprint arXiv:1103.5520 (2011).
12. Yaacoub JP., et al. "Security analysis of drones systems: Attacks, limitations, and recommendations". Internet of Things 11 (2020): 100218.

Volume 5 Issue 4 October 2023