Sigma Journal of Engineering and Natural Sciences
Web page info: https://sigma.yildiz.edu.tr
DOI: 10.14744/sigma.2024.00031

Review Article

Security in UAV ecosystem: An implementation perspective

Nikita MALIK[1,*], Harsh SINHA[1], Menal DAHIYA[1]

[1]Department of Computer Applications, Maharaja Surajmal Institute, Delhi, 110058, India

ARTICLE INFO

Article history
Received: 06 July 2023
Revised: 18 August 2023
Accepted: 06 January 2024

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

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. The primary objectives of this paper are incorporated in five security measures to enhance the security of UAVs by implementing lightweight cryptographic functions for firmware encryption and safeguarding Peripheral Component Interconnect (PCI) buses, fortifying UAV ground stations by disabling non-utilized ports, thereby minimizing potential attack vectors, mitigating threats from malicious actors by implementing an intrusion detection and prevention system (IDPS) to block inorganic network traffic, obscuring the Service Set Identifier (SSID) from broadcast scans, and to reinforce security through filter scrubs and dynamic whitelisting to protect UAV systems against unauthorized access. This paper dives into different methods of the detailed implementation of the aforementioned security measures, providing insights into UAV's configurational practices. Each measure is described and analyzed in parallel to ensure comprehensive coverage of the UAV-Ecosystem's security. The results of this paper are both challenging and rewarding. The proposed measures help improve the security of the UAV ecosystem and protect it from different attacks. In conclusion, this paper has demonstrated the importance of implementing security measures to safeguard the UAV ecosystem from malicious threats. It is crucial to recognize that UAVs may not be entirely immune to sophisticated attackers. Nevertheless, by implementing these security measures and maintaining vigilance, we can prolong the lifespan of UAVs and the entire UAV-Ecosystem and improve their security against malicious intent.

Cite this article as: Malik N, Sinha H, Dahiya M. Security in UAV ecosystem: An implementation perspective. Sigma J Eng Nat Sci 2024;42(6):1986−1994.

INTRODUCTION

Unmanned Aerial Vehicle (UAV) or Drones are rapidly becoming an integral part of modern infrastructure, as dependency on this emerging technology 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 [1]. This expansive use of drones in the Internet of Things (IoT) has created a separate term as the Internet of Drones (IoD) [2] has

*Corresponding author.
*E-mail address: nikitamalik@msijanakpuri.com

This paper was recommended for publication in revised form by Editor-in-Chief Ahmet Selim Dalkilic

Published by Yıldız Technical University Press, İstanbul, Turkey

become increasingly popular due to their convenience and affordability [3]. 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 their ecosystem from malicious activities, such as hacking, eavesdropping attacks [4] and data breaches. The focus is on the use of lightweight cryptographic functions to encrypt the firmware [5], 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 whitelisting to protect the UAVs web application interface (WAi) [6] from Remote File Inclusion (RFI) and API abuse attacks [5]. 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. [5], and the structure of the paper has been defined in Figure 1.

Related Work

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

Lightweight Cryptographic Function

While implementing a Light Weight Cryptographic Function [11] 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 [5] 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 [1].

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, Encrypting the firmware of the UAV can discourage an attacker from a direct attack but a side-channel attack is still an imminent threat where the attacker aims to exfiltrate critical information as cryptographic keys, by measuring coincidental hardware emissions of the UAV such as Electromagnetic waves to measure the electromagnetic radiation emitted by the UAV trying to reconstruct the entire signal packets and another side channel attack is the Power dump attack where the attacker attempts to leverage the power consumption of the UAV hardware depriving, or lowering power source in an attempt to cause a processing error or segmentation fault.

These side-channel attacks can weaken the security of the firmware and PCI buses as this attack aims to corrupt underlying encryption keys and cryptographic processes to create various openings for future attacks, such as privilege escalation attacks on the UAV. Side-channel attacks can be mitigated without compromising the Light Weight Cryptographic Function efficiency by increasing the system noise in the electromagnetic waves emitted by the isotopic radiator of the UAV the only drawback of using this method is that the signal sent by the UAV will have to be processed to subtract the spectral content of the noise from the signal (filtered signal = unfiltered signal - noise) that can result in heavy processing time and cost extra power usage on the power-limited UAV but even these mitigate methods can only decrease the likelihood of occurring of these hardware vulnerabilities.

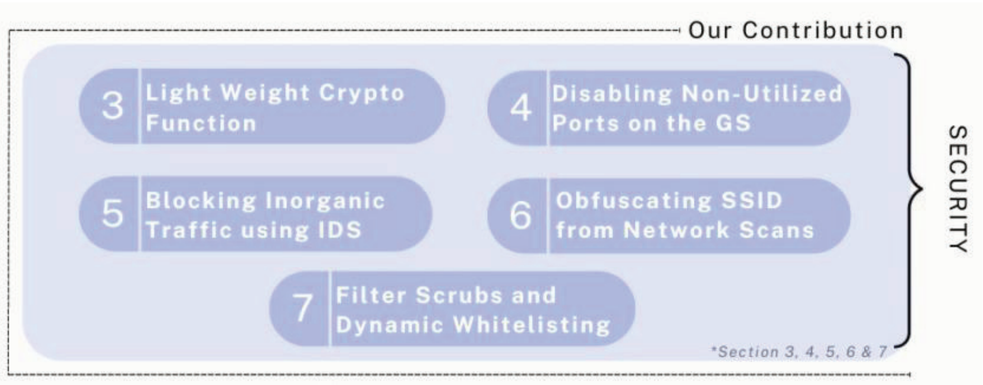

Our Contribution
3 Light Weight Crypto Function
4 Disabling Non-Utilized Ports on the GS
5 Blocking Inorganic Traffic using IDS
6 Obfuscating SSID from Network Scans
7 Filter Scrubs and Dynamic Whitelisting
*Section 3, 4, 5, 6 & 7
SECURITY

Figure 1. Structure of the Paper.

Table 1. Comparative Overview of Related Literature

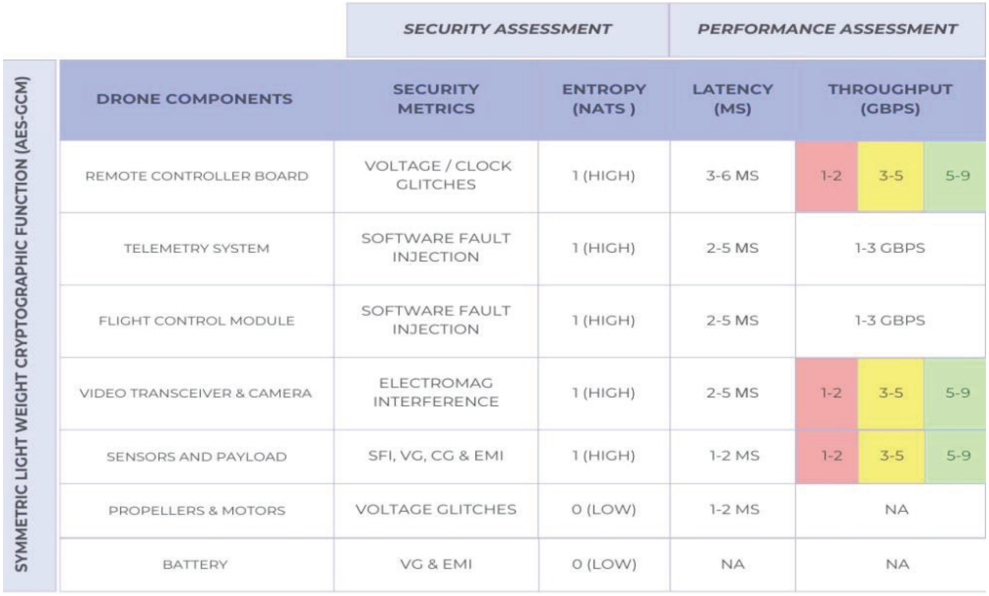
Research Work	Methodology	Research Gap
Sinha, H., Malik, N., Dahiya, Menal. (June 2023) [5]	This paper proposes a novel architecture for configuring and securing unmanned aerial vehicles (UAVs).	The paper does not discuss the potential scalability and performance issues associated with the proposed architecture.
Tiwari, M., Kumar, R., Bharti, A., & Kishan, J. (2017) [7]	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) [8]	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) [9]	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) [6]	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) [2]	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) [10]	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) [11]	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) [3]	This paper provides an overview of attacks and the 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) [12]	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) [13]	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) [14]	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.

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, overall providing high throughput, high entropic value and low latency, as shown in Figure 2. This makes it a suitable choice for drone firmware encryption as neither the security nor performance of the UAV is compromised with the use of this Light Weight Cryptographic Function.

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 the binary diffing technique that involves comparing two binaries of different versions of the same software and



SYMMETRIC LIGHT WEIGHT CRYPTOGRAPHIC FUNCTION (AES-GCM)

	SECURITY ASSESSMENT		PERFORMANCE ASSESSMENT			
DRONE COMPONENTS	SECURITY METRICS	ENTROPY (NATS)	LATENCY (MS)	THROUGHPUT (GBPS)		
REMOTE CONTROLLER BOARD	VOLTAGE / CLOCK GLITCHES	1 (HIGH)	3-6 MS	1-2	3-5	5-9
TELEMETRY SYSTEM	SOFTWARE FAULT INJECTION	1 (HIGH)	2-5 MS	1-3 GBPS		
FLIGHT CONTROL MODULE	SOFTWARE FAULT INJECTION	1 (HIGH)	2-5 MS	1-3 GBPS		
VIDEO TRANSCEIVER & CAMERA	ELECTROMAG INTERFERENCE	1 (HIGH)	2-5 MS	1-2	3-5	5-9
SENSORS AND PAYLOAD	SFI, VG, CG & EMI	1 (HIGH)	1-2 MS	1-2	3-5	5-9
PROPELLERS & MOTORS	VOLTAGE GLITCHES	0 (LOW)	1-2 MS	NA		
BATTERY	VG & EMI	0 (LOW)	NA	NA		

Figure 2. Quantitative analysis of different PCI buses (components of UAV) for measuring the security and performance of the AES-GCM.

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 and Figure 3 [9]. 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 hardware typically has a high entropy value close to 1.

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

Disabling Non-Utilized Ports on the Ground Station

The Ground Station (GS) serves as the command centre 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) [11] 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 [5].

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. Therefore, it is important to close non-essential ports such as port 3389 and others to reduce the surface of vulnerability by reducing potential entry points for attackers, unprotected ports can be exploited by attackers to gain unauthorized access to the UAV’s systems control. The idea here is to discourage attackers and make it more challenging for them to find vulnerabilities providing an additional layer of defence for the UAV and its ground station.

For our Linux GS, we used the ‘apt-cache pkg names’ 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 [5] primary reason to use an SNORT IDS over any other IPS for our UAV-Ecosystem is to minimise the potential halt caused in the event of a false positive (legitimate actions misidentified as a security threats to the entire ecosystem) disrupting critical ground station functions whereas the IDS will notify the network administrator and wait for an assessment of the event as it continuously monitors all active ports on the network, looking for packets that match against the predefined rules, list of known threats and their indicators of compromise (IOCs). 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 [7]. This type of detection is known as signature-based detection used for identifying known threats.

SNORT looks for attack patterns within network traffic by analysing 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) [6]. SNORT looks for the exchange of a sequence of related packets in a certain pattern (signature-based detection), which could indicate that a port scan is in progress using NMAP or any other network scanners [7].

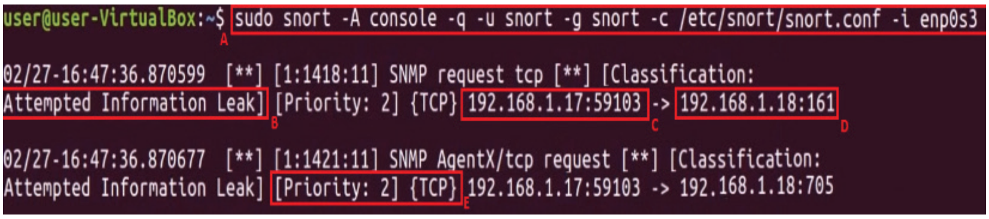
Anomaly-based detection used by the SNORT NIDS identifies inorganic traffic by establishing a normal behaviour baseline for an entire UAV-Ecosystem's network activity. So that SNORT NIDS can be effective at identifying unusual patterns of activity or out-of-the-ordinary behaviour to trigger alerts. The baseline is dynamic and updated in real-time to suit the needs of the ever-evolving UAV-Ecosystem (addition of new components like new UAVs or SGS). Ensuring that the anomaly detection system remains accurate even in the ever-evolving UAV-Ecosystem environment and reduces the risk of false positives.

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 3 an example of signature-based detection.



```
user@user-VirtualBox:~$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i enp0s3

02/27-16:47:36.870599  [**] [1:1418:11] SNMP request tcp [**] [Classification:
Attempted Information Leak] [Priority: 2] {TCP} 192.168.1.17:59103 -> 192.168.1.18:161

02/27-16:47:36.870677  [**] [1:1421:11] SNMP AgentX/tcp request [**] [Classification:
Attempted Information Leak] [Priority: 2] {TCP} 192.168.1.17:59103 -> 192.168.1.18:705
```

Figure 3. Alert output of SNORT.

[1] *https://www.cvedetails.com/cve/CVE-2022-30190/*

[2] *https://www.cvedetails.com/cve/CVE-2020-0822/*

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, 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 [14] 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 [5]. Wireless network cards used in network routers often run on Unix or Linux-embedded systems [8]. 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 overly aggressive dynamic adaptation can lead to an increase in false positives, 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. Integrating these dynamic adaptation mechanisms into existing security systems can be challenging and a daunting task. However, while this method can protect the UAV Ecosystem against passive scans, the SSID will still be visible in the beacon frames which could result in gaining system access via frag attack (fragmentation and aggregation attacks) [15]. 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 the UAV ecosystem's Web Applications Interface [10] and Application Program Interface (APIs) from malicious input parameters and file inclusions [5] 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 [10] 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 [6].

To enhance the security of our system, we have implemented rate-limiting and request throttling measures that improve the system resilience of the UAV. 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. Ensuring security measures can grow with the UAV-Ecosystem, using automation and centralization will allow future scalability possible.

RESULTS AND DISCUSSION

Through this paper, we have proposed baseline guidelines that can be followed with all types of UAVs regardless of the size, type and operational environment and, without any overhead implementational complexities.

- The proposed measures include firmware encryption with AES-GCM- a lightweight cryptographic function, which is an efficient encryption algorithm that helps meet security requirements without significantly affecting the drone's performance.
- To understand the risk associated with firmware encryption of PCI buses, we have done a security and performance assessment of the UAV, as shown in Figure 2, based on the criteria of Throughput (in Gbps) based on the old, medium or high-end model, Latency (in ms)

Table 2. Comparative Study of Advantages and Shortcomings of our Paper

Advantages of the Study	Shortcomings	Future Works
We have used Authenticated Encryption (AES-GCM) to avoid boot time limitations. AES-GCM requires less computational power and bootup time compared to Symmetric and Asymmetric Encryption. This encryption provides secure encryption to the firmware of the UAV preventing it from direct attacks.	AES-GCM is still perceptible to Side channel attacks such as (i) measuring coincidental hardware emissions from the UAV, and (ii) Power dump attacks these attacks aim to corrupt underlying encryption keys and cryptographic processes to create various openings for future attacks.	Future work could include working on more sophisticated countermeasures against hardware emissions measurement and power dump attacks to strengthen AES-GCM security. Machine learning algorithms can be used to detect and respond to patterns indicative of side-channel attacks, providing dynamic defence mechanisms against them.
To mitigate the side-channel attacks on UAVs, and to maintain the lightweight cryptographic Function efficiency we can increase the system noise in the electromagnetic waves emitted by the isotopic radiator of the UAV.	The signal sent by the UAV will have to be processed to subtract the spectral content of the noise from the signal which increases the signal processing time and uses more power on the power-limited UAV.	Future works could include working on a cloud-based noise-processing environment so that the UAV does not have to subtract the spectral content of the signal and can be handled by the cloud-based environment, removing the extra power usage on the Drones.
This study provides a clear guide in using a Linux-based Ground station dedicated computer system to avoid other preinstalled software having vulnerabilities that could potentially risk the UAV's security.	Linux systems require specialized knowledge in setting up and maintenance. If managed correctly could result in challenges for users who are unfamiliar with Linux, potentially leading to misconfigurations, security oversights, or operational issues.	The focus of future works would be on creating and developing automated configuration scripts to avoid unfamiliar users to Linux from harbouring misconfigurations and security oversights. Further reducing operational issues.
We have used SNORT NIDS to identify and categorize malicious network activity and inorganic traffic based on signature-based anomaly detection. SNORT generates alerts to notify the network administrator of potential security threats before an attacker can cause damage to the network.	A major drawback of this advantage is SNORT NIDS identifies inorganic traffic by establishing a normal behaviour baseline for an entire UAV-Ecosystem's network activity if it incorrectly identifies a false negative then the normal baseline will be disrupted resulting in allowing future attacks of a similar kind to disrupt the services of the network.	To minimize such false negatives in future works we can use adaptive algorithms that dynamically adjust the normal behaviour baseline of SNORT NIDS based on real-time changes done by the maintenance engineers.
Obfuscating SSID from network broadcast scans so that network scans from Wireshark and Airodump-ng become more difficult for the Attacker. This method can protect the UAV Ecosystem against passive scans, the SSID will still be visible in the beacon frames which could result in gaining system access via frag attack.	This overly aggressive dynamic adaptation can lead to an increase in false positives, and it could also prevent legitimate users from connecting to the UAV Ecosystems network.	The focus of our future work would be to develop more effective methods for hiding the SSID from broadcast scans. And reducing overly excessive dynamic adaptation.
To protect our web applications and APIs from malicious input parameters, Filter scrubs and dynamic whitelisting are techniques used to protect the UAV and its ecosystem's Web Applications Interface and Application Program Interface from malicious activities. We also propose to use Rate-limiting and request-throttling measures to serve as effective safeguards by restricting the number of incoming requests within a specified timeframe and enabling us to detect any suspicious activities and patterns that may emerge from potential attacks.	As mentioned before this method has added an overly aggressive dynamic adaptation that can lead to an increase in false positives.	The focus of our future work would be to reduce excessive dynamic adaptation to eliminate the number of false positives.

and Entropy calculation (in Nat) under multiple security matrices.

- Balancing security with resource constraints, intrusion detection systems (IDS) like SNORT may generate false positives, flagging legitimate activities as security threats. While SNORT provides signature-based and anomaly-based detection, dynamic whitelisting can help reduce false positives.
- Additional signal processing time is required when enhancing UAV security through the addition of signal noise.
- An increase in power consumption can be observed as a consequence of implementing certain security measures for UAV protection against side-channel attacks.

We have also listed the advantages, shortcomings and a brief explanation about future works that can be achieved with the obtained results as shown in Table 2.

By carefully balancing these security measures and addressing their interactions with UAV Ecosystems, drone operators can enhance security while minimizing the disruptions to UAV operations in real time.

CONCLUSION

This research paper has proposed a set of security measures to enhance the security, efficiency, functionality, and lifespan of the IoD by applying the firmware encryption, disabling non-utilized ports, blocking inorganic traffic using IDS SNORT, intrusion detection systems, obfuscating SSID from network broadcast scans, and implementing filter scrubs and dynamic whitelisting for web application interfaces. The underlying model for preventing malicious requests can be rigorously quantified and modelled, taking into account elements like attack frequency, severity, and cumulative impact, in order to extend the lifespan of a UAV and implement security measures. For this, we can first start with the groundwork of gathering historical data on attack frequency, severity, and their cumulative impact on UAV operations. This data will include successful and attempted attacks to create a proper threat model that identifies potential attack vectors, and their likelihood of occurrence. With this enough data, it can be possible to calculate the risk associated with each attack vector and prepare a countermeasure beforehand, such as firmware encryption to evade hardware-based attacks and glitches. A holistic risk assessment can evaluate the cost-effectiveness of the provided security measures and can be explored in future works. By following these protocols, we can identify the most cost-effective security measures to prolong UAV lifespan and enhance the security of the overall UAV-Ecosystem.

While these measures can provide a strong level of security, there is still room for further research to explore how 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 the security and integrity of the UAV ecosystem.

AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

DATA AVAILABILITY STATEMENT

The authors confirm that the data that supports the findings of this study are available within the article. Raw data that support the finding of this study are available from the corresponding author, upon reasonable request.

CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ETHICS

There are no ethical issues with the publication of this manuscript.

REFERENCES

[1] Abualigah L, Diabat A, Sumari P, Gandomi AH. Applications, deployments, and integration of internet of drones (IOD): a review. IEEE Sensors J 2021;21:25532–25546. [CrossRef]

[2] Yaacoub JP, Noura H, Salman O, Chehab A. Security analysis of drones systems: Attacks, limitations, and recommendations. Internet Things (Amst) 2020;11:100218. [CrossRef]

[3] Sinha H, Malik N, Dahiya M. Drone Ecosystem: Architecture for Configuring and Securing UAVs. Proceedings of Fourth International Conference on Computing, Communications, and Cyber-Security (IC4S 2022). Singapore: Springer; 2023. [CrossRef]

[4] Mekdad Y, Aris A, Babun L, Fergougui AE, Conti M, Lazzeretti, R, et al. A Survey on Security and Privacy Issues of UAVs. arXiv preprint arXiv: 2021:2109.14442.

[5] Tiwari M, Kumar R, Bharti A, Kishan J. Intrusion detection system. Int J Tech Res Appl 2017;5:38-44.

[6] Keleman L, Matić D, Popović M, Kaštelan I. Secure firmware update in embedded systems. In 2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin) (pp. 16-19). IEEE, 2019. [CrossRef]

[7] Wu Y, Noonan JP, Agaian S. Shannon entropy based randomness measurement and test for image encryption. arXiv preprint arXiv:2021:1103.5520.

[8] Haider SK, Nauman A, Jamshed MA, Jiang A, Batool S, Kim SW. Internet of drones: Routing algorithms, techniques, and challenges. Mathematics 2022;10:1488. [CrossRef]

[9] Lin C, He, D, Kumar N, Choo KKR, Vinel A, Huang X. Security and privacy for the internet of drones: Challenges and solutions. IEEE Commun Mag 2018;56:64–69. [CrossRef]

[10] Dabrowski A, Pianta N, Klepp, T, Mulazzani M, Weippl E. IMSI-catch me if you can: IMSI-catcher-catchers. In Proceedings of the 30th annual computer security applications Conference (pp. 246–255), 2014. [CrossRef]

[11] Langley A, Riddoch A, Wilk A, Vicente A, Krasic C, Zhang D, 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 (pp. 183–196), 2017. [CrossRef]

[12] Roy M, Ahsan S, Kumar G, Vimal, A. Implementation of quick UDP internet connections (QUIC) protocol. Int J Eng Comput Sci 2020;9:24921–24924. [CrossRef]

[13] Miller I. Protection against a variant of the tiny fragment attack (RFC 1858) (No. rfc3128), 2001. [CrossRef]

[14] Xu F, Ahmad S, Ahmed M, Raza S, Khan F, Ma Y, et al. Beyond Encryption: Exploring the Potential of Physical Layer Security in UAV Networks. J King Saud Univ Comput Inform Sci 2023;35:101717. [CrossRef]

[15] Mekdad Y, Aris A, Babun L, El Fergougui A, Conti M, et al. A survey on security and privacy issues of UAVs. Comput Netw 2023;224:109626. [CrossRef]