Detection of Targeted Attacks Using Medium-Interaction Honeypot for Unmanned Aerial Vehicle

Abdul Majid Jamil[1(✉)], Hassan Jalil Hadi[1], Sifan Li[1], Yue Cao[1], Naveed Ahmed[2], Faisal Bashir Hussain[3], Chakkaphong Suthaputchakun[4], and Xinyuan Wang[5]

[1] School of Cyber Science and Engineering, Wuhan University, Wuhan, China
{majidjamil,yue.cao}@whu.edu.cn
[2] Prince Sultan University, Riyadh, Saudi Arabia
nahmed@psu.edu.sa
[3] Bahria University, Islamabad, Pakistan
fbashir.buic@bahria.edu.pk
[4] Bangkok University, Bangkok, Thailand
chakkaphong.s@bu.ac.th
[5] Zhejiang Scientific Research Institute of Transport, Hangzhou, China

Abstract. Over the last two decades, there has been significant growth in the drone industry with the emergence of Unmanned Aerial Vehicles (UAVs). Despite their affordability, the lack of security measures in commercial UAVs has led to numerous threats and vulnerabilities. In addition, software, and hardware complexity in UAVs also trigger privacy and security issues as well as cause critical challenges for government, industry and academia. Meanwhile, malicious activities have increased, including stealing confidential data from UAVs and hijacking UAVs. These attacks are not only illegitimate but also appear to be increasing in frequency and sophistication. In addition, the current defence mechanisms for counterattacks are not sustainable for two reasons: they either demand strict firmware updates for all of the system's devices, or they demand the deployment of a variety of advanced hardware and software. This paper proposes a Medium Interaction Honeypot-Based Intrusion Detection System (MIHIDS) to protect UAVs. Our system assists in detecting active intruders in a specific range (radio frequency) and provides details of attacking technologies to exploit UAVs. Our system is a passive lightweight, signature-based MIHIDS that is simple to integrate into UAV without requiring changes in network configuration or replacement of current hardware or software. The performance assessment demonstrates that in a typical network situation, our proposed framework can identify MitM, Brute-force, and DE-authentication attacks with a maximum detection time of 60 s. Under normal network scenarios, a minimum True Positive Rate (TPR) and performance efficiency is 93% to 95% during a short-distance detector.

Keywords: Unmanned Aerial Vehicle · Medium Interaction Honeypot · Intrusion Detection System

S. Goel and P. R. Nunes de Souza (Eds.): ICDF2C 2023, LNICST 571, pp. 164–185, 2024.
https://doi.org/10.1007/978-3-031-56583-0_11

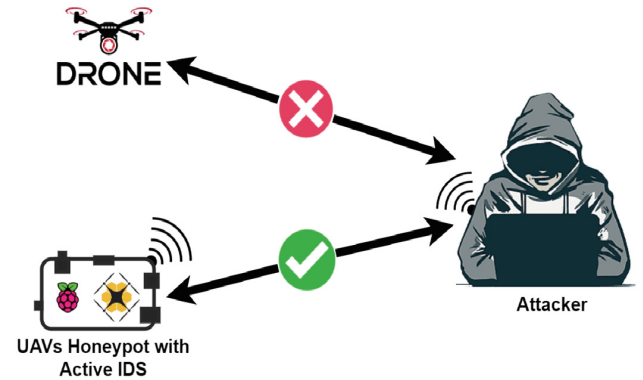
1 Introduction

Drones, also known as Unmanned Aerial Vehicles (UAVs), have become highly popular in recent years. UAVs are used for commercial and domestic purposes, including surveying agriculture, delivering packages, shooting pictures and films. Aside from that, UAVs are used for even mission-critical tasks, such as pharmaceutical distribution, health and safety monitoring [1,2]. However, these might involve data theft, mission disruption, or stealing and illegally using UAVs.

Attacks on UAVs are becoming more widespread as these devices are computer-controlled and have radio or wireless communication. A system that only serves as a target for attacks, reconnaissance, and compromise is known as a honeypot [3]. Typical uses of honeypots include early warning defence mechanisms, methods for investigating attackers and their approaches, and minimising a monitored network's attack surface.

In addition to the honeypot functions mentioned above, we contend that a UAV honeypot brings further benefits, by exploiting the specific characteristics of UAVs (especially the signal strength quality and the capability to move quickly through an area). Using the UAV's signal strength rather than maintaining visual contact with the target, a UAV attack scenario, in particular, enables the attacker to reap the benefits and control the UAV (e.g., by using a strong-signal antenna). Therefore, we claim that a UAV honeypot-based IDS is capable of detecting a UAV attack as long as (i) it transmits a signal that is stronger than UAVs themselves (done with a suitable antenna), and (ii) it is positioned in a desirable range (radio frequency).

High-interaction honeypots are real systems that highlight specific vulnerabilities and they are monitored closely. More specifically, this system creates nothing but a perception of vulnerability, encouraging adversaries to attack it. It may be low, medium, or high interaction, depending on how much interaction the attacker has with the honeypot. Since maintaining high-interaction honeypots is quite expensive, there is a chance that they could be compromised. Meanwhile, low- and medium-interaction honeypots are easier and simpler to monitor and configure because they emulate protocols. Figure 1 depicts a UAV honeypot overview.



High-Interaction Honeypot	Medium-Interaction Honeypot	Low-Interaction Honeypot
• Simulates all aspects of the OS	• Limited access to the OS	• Limited access to the OS
• Real System, Emulate all Service	• Emulate internet protocols	• Emulate limited internet protocols
• Virtual Machine, High Risk	• Emulate network services	• Emulate limited network services

Fig. 1. UAV Honeypot Overview

This paper proposes a Medium Interaction Honeypot Intrusion Detection System (MIHIDS), a model to gather the most helpful information (attack methods) about the attacker. Our system can help identify active attackers in a specific range (radio frequency), and enlighten on attack methods to compromise UAVs. We evaluate system performance, and accuracy to demonstrate how the honeypot can respond to various realistic attack scenarios. Moreover, our system also detects and generates the logs of each action performed by attackers, and keeps a record of these logs in the MySQL DB for further analysis of these malicious activities. The main contribution of this paper are summarized as follows:

- Provide medium-interaction honeypot for UAV specified and tailored protocols (FTP, SSH, TELNET, and MAVLINK), record as well as analyze malicious activity and distract attackers while reducing the attack surface.
- Experimental analysis of attack traffic and creation of potential attack signatures for MitM, Brute force, and De-Authentication attacks.
- Develop a MIHIDS based on a behavioural rule specification by combining signature-based algorithms. It uses minimal memory, while maximizing detection accuracy by checking for observable behavioural anomalies. An attack event is identified by anomaly detection using a threshold-based approach, which is based on a network traffic profile of attack behaviour.

2 Related Work

Honeypot systems with value lying only in probes are attacked and compromised. Honeypots are of three types, low-interaction, medium-interaction and high-interaction, based on the interaction level they provide to the adversary.

Without real production value, any interaction or communication with the system is thought of as an attack. On the contrary, real systems like Virtual Machines (VM) lie in the class of high-interaction honeypots. High-interaction honeypot systems are keenly monitored and can exhibit specific vulnerabilities. However, these systems are expensive and the risk of being compromised is high. Conversely, low-interaction as well as medium-interaction systems, only emulate protocols. This system has become a priority compared to high-interaction systems for multiple factors [6]. The first factor is low cost and easy maintenance because they can provide detailed and explicit logging along with monitoring functionalities. The second factor is the ease with which these systems can be developed, contained, and secured.

Table 1. Evaluating the Capabilities of Available Honeypots

Reference	Simulated Service	MAVLINK	Level of Interaction	Resource Level	Role
HoneyCloud [16]	SSH,Telnet,HTTP,MySql	No	Physical/Virtual	High	Server
HosTaGe [7]	FTP, SSH, Telnet	No	Virtual	Medium	Server
Heralding [11]	FTP, SSH, Telnet	No	Virtual	Low	Server
Bluepot [10]	L2CAP BT, RFCOMM, OBEX	No	Virtual	Medium	Server
HoneyWRT [13]	Telnet	No	Virtual	Medium	Server
HoneyPy [12]	TCP,UDP	No	Virtual	Medium	Server
Cowrie [14]	SSH, Telnet	No	Virtual	Low	Server
Kojoney2 [15]	SSH	No	Virtual	low	Server

As listed in Table 1, numerous honeypots are capable of monitoring and simulating a variety of general-purpose protocols, including SSH, FTP, and Telnet [3–5,9]. Moreover, the investigation of honeypots in handling the MAVLink protocol for UAVs has not been adequate. Similarly, none of the honeypots can imitate extracted File System (FS) and record all unauthorized modifications of the UAV. Furthermore, most honeypots cannot be configured to impersonate other devices, since they are developed for a single set of use cases. Only a few honeypots have a portability feature to fit a UAV operation or are even directly attached to one [7]. A honeypot must be able to simulate UAV radio interfaces although none of them is designed to do so. Only Bluepot can simulate the rarely used Bluetooth radio [10]. Moreover, the classification of honeypots based on the level of interaction in the above table is the most flexible. Furthermore, the author states that game theory has been used in the past to model security scenarios but has not been applied to the problem of UAV network security [17]. The paper attempts to bridge this gap by proposing a game-theoretic framework for collaborative honeypot defence in UAV networks [18].

3 Proposed MIHIDS

The primary security technologies for UAVs are detection and defence. The system can identify significant attacks from time series using behaviour-based anomaly detection, which can detect attacks at an accurate and consistent stage, also supporting timely early warning. These systems work together to quickly implement the necessary security protections, reducing the probability of an attack.

3.1 Framework Overview

The infrastructure, security protection, application, and user interfaces layers are the four layers that build up the security framework, as depicted in Fig. 2.

The security framework is built on the bottom of the **Infrastructure Layer**. Its purpose is to provide various data and computational resources for different applications. It includes database management, activity logging, and a testbed environment. Databases are used to keep a record of logs and alerts. The testbed environment serves as the interface between the detection and active defence modules. Meanwhile, it is also utilized to carry out various computing operations.

The **Security Protection Layer** serves as the foundation of the security framework, and also performs functions for the defence and detection of all types of security threats. The active defence module is used to protect against attacks, while the detection module is used to identify any unusual intrusions into the system. The two sub-modules of the security layer are described in depth in Sects. 3:3.2 and 3.3.

The **Application Layer** offers different application services for each UAV function, including GPS, UAV flight controller, authentication APP and data collection. Information about users and data from UAVs is collected by using a data collector. The UAV flight controller is used to direct the movement and data transmission of the UAV. The authentication APP can be used for UAV-Server or UAV (client-honeypot) authentication. The GPS is a fundamental component of the UAV flight process, as it provides flight paths and allows the user to track the location of the UAV in real-time.

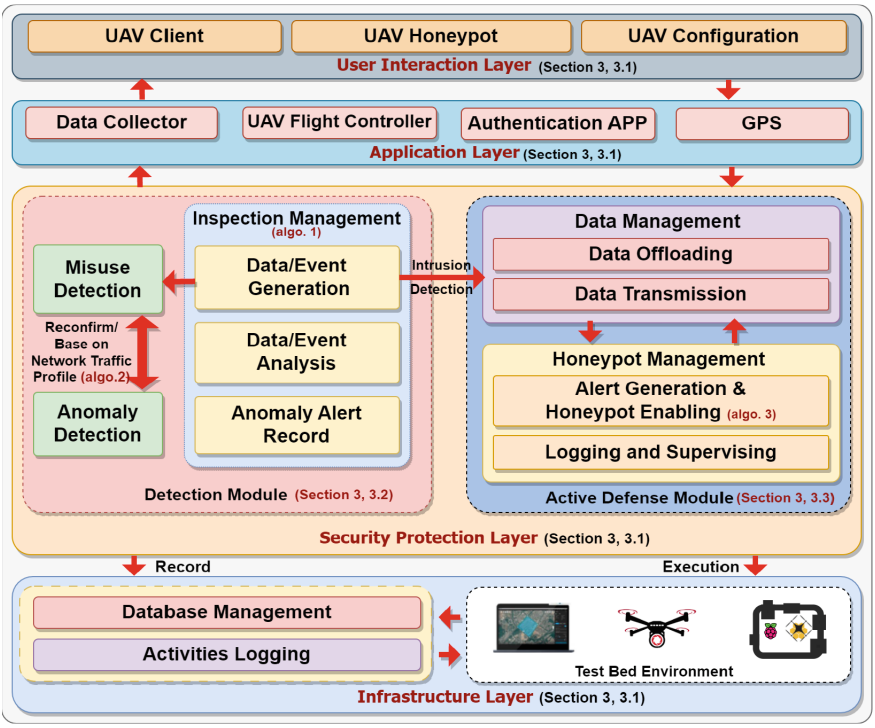


Fig. 2. MIHIDS Framework Overview

The **User Interaction Layer** provides an interactive interface for users that includes the UAV client, UAV honeypot, and UAV configuration.

3.2 Detection Module

Real-time analysis of network traffic is essential for identifying potential intrusions against UAVs while they are in flight. One effective approach is to deploy an intrusion detection system (IDS) that can detect a range of intrusion classes, including signal alteration, routing attacks, malware, and message forging attacks [19,20]. It's also important to develop anomaly detection frameworks that can identify unusual patterns of activity that could indicate malicious behavior. Furthermore, to provide additional protection against hostile actors, the use of honeypots and honeynets in combination with an IDS is recommended. In order to create intelligent devices, it's necessary to provide them with instructions and define rules that govern their behavior, as shown in Algorithm 1. In the case of UAVs, rules are stored in memory (database) and levels of rule acceptance are established. To minimize false-positive predictions, a new IDS has been proposed that is based on rules of behavior.

Misuse detection, detection management and anomaly detection technologies are mainly included in the detection module of MIHIDS. The misuse detection technique is used to detect cyber-attacks. The detection stage, which uses the collected data to compare with the threshold value or signature rule to judge whether it conforms to the rules. If the detection result is "conforming", there will be no abnormal intrusion; otherwise, there will be an abnormal intrusion, as shown in Algorithm 2. This method has high accuracy and relatively mature technology, which is convenient for system maintenance. However, it is very difficult to update and maintain the signature database.

Anomaly detection determines the attack event by using the rule-based matching method based on the rule database of the attack behaviour. It is necessary to establish the "network traffic profile (Device Behaviour)" of the subject normal activities, and compare the current activity status of the subject with the "network traffic profile (Device Behaviour)". Also, when the subject violates its signature-based rules that activity is considered an "invasion". This method can detect intrusions that have never occurred or the type of abuse of authority, and has little dependence on the operating system. However, this method has a high alarm rate. Besides, it is difficult to establish an "network traffic profile" and design a signature-based algorithm. It does not detect normal operations as "intrusion" or ignore the real "intrusion" behaviour. The primary purpose of **Inspection Management** is to control the generating, analyzing, and handling of data events and alerts related to suspicious records.

3.2.1 Signature Development for Intrusion Traffic: By developing signatures of intrusion traffic and utilizing analytics based on an empirical analysis of intrusion and normal traffic, we can efficiently detect various intrusion signatures using a threshold-based technique, which enables us to identify Brute-Force, MitM, and De-Authentication attacks based on network traffic characteristics. Furthermore, we conduct a continuous monitoring of the legitimate client UAV and its specific ports for a probe period of 5 min, followed by the implementation of 4 phases of Brute-Force, MitM, and DE-Authentication attacks lasting for 60–100 seconds.

3.2.1.1 Concurrent Probe or Beacon Response Traffic: Figure 3 describes a network attack record with the concurrent probe or beacon responses on two separate channels with the same BSSID and SSID. Moreover, benign traffic scenarios were examined, but could no concurrent probe response traffic found on different channels in the same frequency range with the same BSSID and SSID.

On the other hand, when they use dual-band frequencies, there may be concurrent beacons if the Server (GCS) broadcasts the same SSID. Since the channels utilised in the 2.4 GHz and 5 GHz bands are different, such concurrent beacons can be clearly differentiated as benign traffic.

In the event of a sudden detection of a large volume of the concurrent probe or beacon response traffic on two different channels with similar BSSID and SSID, the UAV GCS Server can generate traffic analysis warnings, as it may

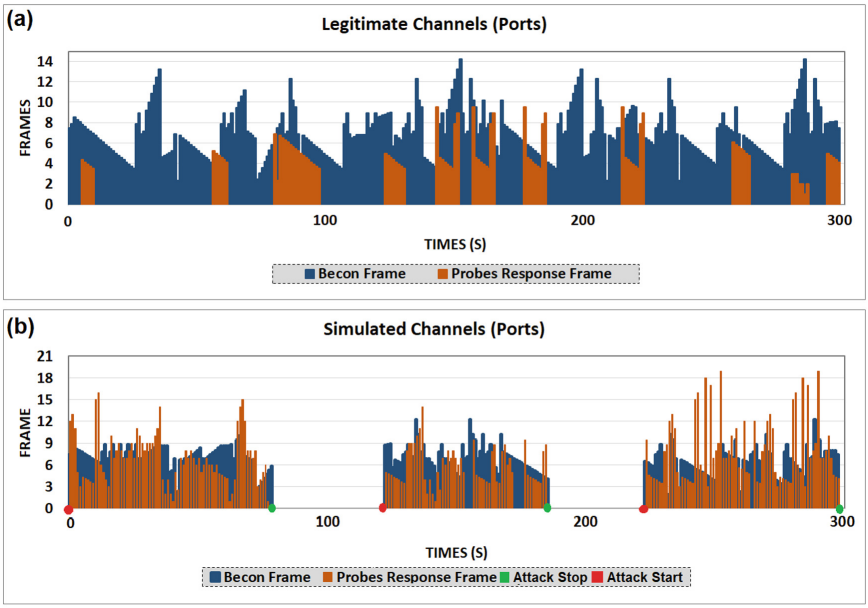


Fig. 3. Network intrusion trace during probe and beacon responses (a) legitimate channel (b) simulated channel

indicate the start of an intrusion. To efficiently detect concurrent probe or beacon response traffic that coincides with attacks, a threshold (TH1) of the beacon or probe response frame is set to 1, allowing for rapid identification during a probe interval. Additionally, the presence of intrusion is validated by examining concurrent traffic in the UAV network.

3.2.1.2 Concurrent Connection Traffic: A network attack trace is shown in Fig. 4 with two channels carrying concurrent authentication frames on the same SSID and BSSID.

A network attack trace is shown in Fig. 5 with simultaneous association requests and response frames originating on 2 different channels with similar BSSID and SSID.

Figures 4 and 5. depict the trace of a network attack, along with concurrent frames on two different channels with identical BSSID and SSID. From these observations, it has been confirmed that concurrent connection traffic can be utilized as an attack signature for the identification of MitM, Brute-force, and DE-Authentication intrusions in a UAV client. When a UAV client is targeted by one of these attacks, two concurrent association frames (request and response) and two concurrent authentication frames (request and response) are captured on both channels, all with similar BSSID and SSID. This is due to the transfer of association and authentication frames between real and simulated channels (ports) by the attacker. To ensure swift detection of concurrent connection traffic containing a Brute-force, MitM, or DE-Authentication intrusion, the threshold (TH2) for association and authentication frames has been set to 1 frame. Setting

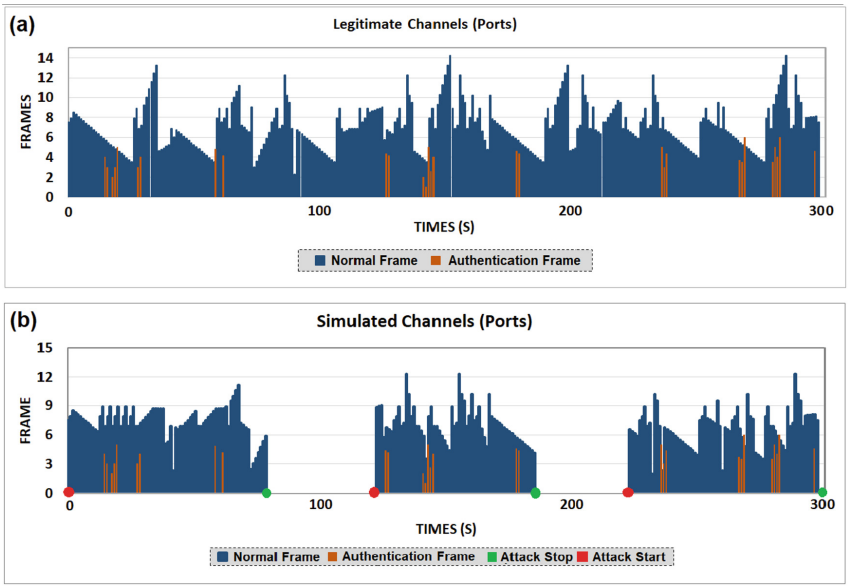


Fig. 4. Network intrusion trace during authentication responses (a) legitimate channel (b) simulated channel

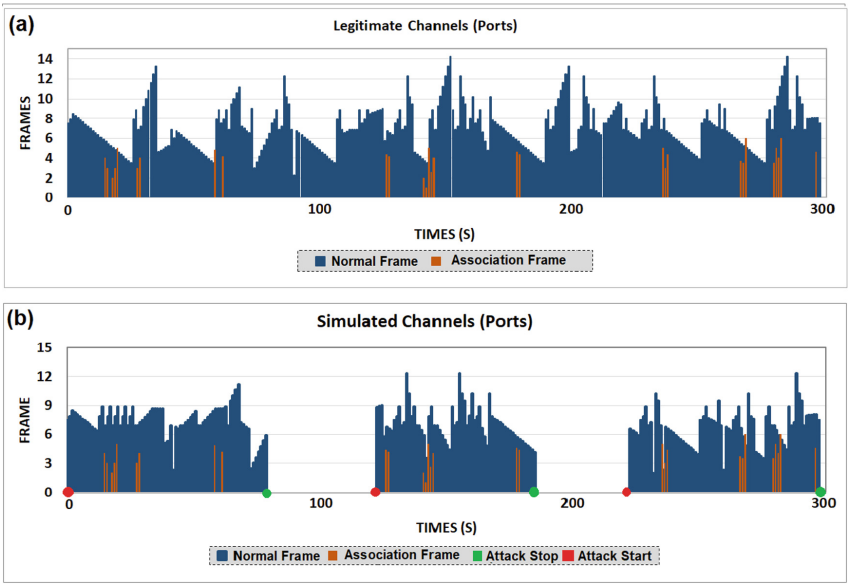


Fig. 5. Network intrusion trace during association responses (a) legitimate channel (b) simulated channel

the threshold value for association and authentication frames to 1 helps to detect and respond quickly to brute-force, MitM, or DE-Authentication intrusions. This is because these attacks are typically automated and fast-paced, with multiple frames being sent in quick succession. A threshold value of 1 helps catch the attack as soon as it starts while minimizing false positives.

Several intrusion signatures are compiled in Table 2 to help identify MitM, Brute-force, and DE-Authentication attempts during a probe interval.

Algorithm 1: Traffic analysis while establishing simultaneous connections. The GCS (UAV-Server) and a specific UAV-Client are monitored in real-time by this method as they establish a connection. This algorithm monitors synchronous communication for association and authentication. Additionally, this algorithm also detects traffic using the source and destination MAC address of the device running in parallel.

Table 2. Several Intrusion Signatures

Intrusion Signature	Statistic Used	Threshold
Concurrent Beacon Traffic	No. of the probe or beacon response	TH1 $\geq$1
Concurrent Connection Traffic	No. of Authentication Frames	TH2 $\geq$1
Concurrent Connection Traffic	No. of Association Frames	TH3 $\geq$1

Algorithm 1: Concurrent UAV Traffic Analysis

```
Input: UAV Traffic
Output: Number of synchronous frames (AUTHCC and AUTHRC) or (ASSOCC and
        ASSORC), GCS-MAC = MAC ID of the GCS, C-CHANNEL = Current
        channel(port) of the GCS
while probe-interval do
    Extract SMAC, DMAC, and Channel of the frames;
    if frame[Dot11].type == 0 and frame[Dot11].subtype == 11 then
        while UAV-Client-MAC in device database do
            if (smac == GCS-MAC and dmac == UAV-Client-MAC) or (smac ==
              UAV-Client-MAC and dmac GCS-MAC) and channel == C-CHANNEL
              then
                Count authentication-current-channel(AUTHCC);
            if (smac == GCS-MAC and dmac == UAV-Client-MAC) or (smac ==
              UAV-Client-MAC and dmac GCS-MAC) and channel != C-CHANNEL
              then
                Count authentication-rouge-channel(AUTHRC);
    if frame[Dot11].type == 0 and frame[Dot11].subtype == 1 then
        while UAV-Client-MAC in device database do
            if (smac == GCS-MAC and dmac == UAV-Client-MAC) or (smac ==
              UAV-Client-MAC and dmac GCS-MAC) and channel == C-CHANNEL
              then
                Count association-current-channel(ASSOCC);
            if (smac == GCS-MAC and dmac == UAV-Client-MAC) or (smac ==
              UAV-Client-MAC and dmac GCS-MAC) and channel != C-CHANNEL
              then
                Count association-rouge-channel(ASSORC);
```

Algorithm 2: Brute-force, MitM, DE-Authentication attack traffic analyzer: Based on threshold values, this algorithm determines the status of the attack traffic at the completion of each probe interval.

Algorithm 2: Intrusion Traffic Collator

Input: Output of Algorithms 1
Output: Status of Intrusion Traffic
if *(Intrusion-Traffic = True)* **then**
  if *(AUTHCC ≥ TH1 and AUTHRC ≥ TH1 or AOSSCC ≥ TH2 and ASSORC ≥ TH2)* **then**
    CON-Brute-force-Intrusion = True;
  if *(AUTHCC ≥ TH2 and AUTHRC ≥ TH2 or AOSSCC ≥ TH1 and ASSORC ≥ TH1)* **then**
    CON-MitM-Intrusion = True;
  if *(AUTHCC ≥ TH3 and AUTHRC ≥ TH2 or AOSSCC ≥ TH3 and ASSORC ≥ TH1)* **then**
    CON-Credential-Reuse-Intrusion = True;
  if *(CON-Brute-force-Intrusion = True) or (CON-MitM-Intrusion = True) or (CON-Credential-Reuse-Intrusion = True)* **then**
    Intrusion-Traffic = True;
  else
    Intrusion-Traffic = False;

3.3 Active Defence Module

There are mainly two parts of active defence in the MIHIDS security framework including honeypot management and data management. This module uses system-generated defence mechanisms (like control, evasion, deception and detection, etc.) to reach against cyber-attacks. Active defence of decoy technology (Honeypot technology) was primarily studied. The technology is efficient for protecting system security as it deploys vulnerable network services or hosts as bait for deceiving attackers. The data diversion phase of **Data Management**, merges with numerous network protection technologies for the coordination of data collected by the system in the collection, analysis, distribution, firewall and transmission process to make sure safe transmission of data.

Additionally, **Honeypot Management** consists of creation monitoring and logging. It gives administrators a system parameter for managing and configuring user interfaces. In the creation phase, a honeypot is created based on the characteristics of data received from the analysis phase of data management. All events happening in the system are recorded daily with the help of logging and monitoring. The intrusion and detection operations help the system administrator to check the reason of error as well as traces left by attackers. The user identity is authenticated by the authentication APP of the MIHIDS when there is no attack. In this process, numerous operators have data information cooperate for the authentication of user identity. In case of an attack, abnormal behaviour or intrusion is detected by the detection module and the alarm is generated simultaneously, as shown in Algorithm 3. Active defence sets Honeypot for capturing attack behaviour and planning protection measures accordingly.

Algorithm 3: Alert Generation based on the status of attack traffic provided by algorithm 2 predicts the presence of Brute-force, MitM, and DE-Authentication intrusion.

Algorithm 3: Alert Generation

```
Input: Output of Algorithm 2
Output: Alert Generation and Honeypot Activation
if (Intrusion-Traffic = False) then
    LOG as "No Intrusion Found"
if (Intrusion-Traffic = True) then
    if (CON-Brute-force-Intrusion = True and Intrusion-Traffic = True) then
        Alert Generate;
        LOG as "Bruteforce"
    if (CON-MitM-Intrusion = True and Intrusion-Traffic = True) then
        Alert Generate;
        LOG as "MitM"
    if (CON-Credential-Reuse-Intrusion = True and Intrusion-Traffic = True) then
        Alert Generate;
        LOG as "Credential-Reuse"
else
    if (CON-Brute-force-Intrusion = False or CON-MitM-Intrusion = False or
      CON-Credential-Reuse-Intrusion = False) and Intrusion-Traffic = True) then
        Alert Generation;
        LOG as "Bait Session"
```

4 Deployment

In order to have a real attack model, the implementation is done with a variation of Honeypot Server, Honeypot UAV, Client UAV, and Attacker Machine. The Ubuntu 18.04 virtual machine was used as a testbed. The implementation steps of framework are as follows:

- **Step 1:** Ubuntu VM is used for the deployment and configuration of the MIHIDS server.
- **Step 2:** When the Honeypot server is successfully deployed and configured, prerequisite configurations, such as database installation and SSL certificate generation, are carried out to securely access the Honeypot server dashboard.
- **Step 3:** When the configuration is completed, the MIHIDS server generates a secure, and unique link to establish a connection with the client UAV and honeypot UAV.
- **Step 4:** After that, the client UAV and honeypot UAV are configured on two separate Ubuntu VMs and connected to each other using a unique link generated by the MIHIDS server, thereby establishing a connection between the two VMs.

Figure 6 shows the deployment and working of the proposed system. The workflow of the system is as follows: (1) The server is started successfully, (2) The server finds the already connected devices, (3) If the devices (Client UAV and Honeypot UAV) are not found, it will generate a unique link, (4) The UAV

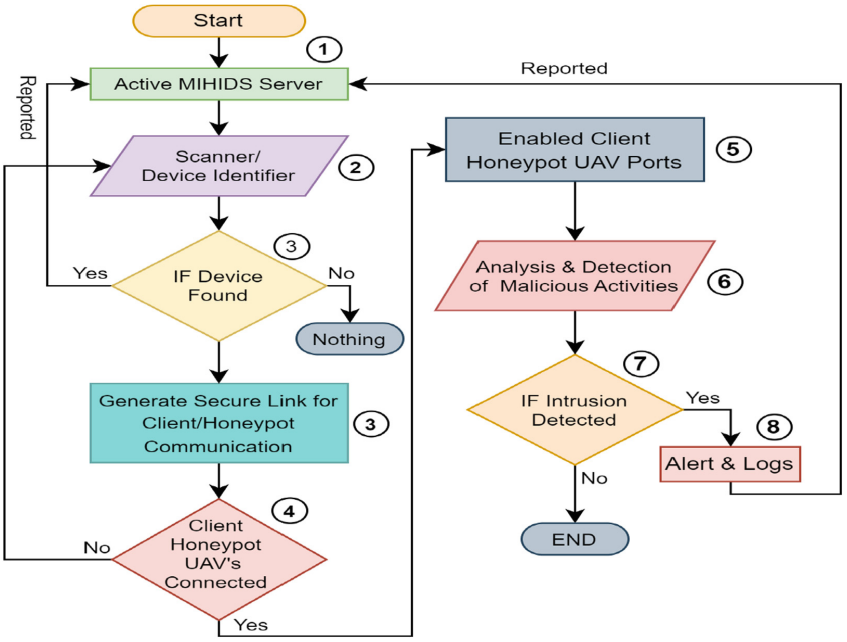


Fig. 6. WorkFlow Diagram

client and UAV honeypot are connected by using the link and reporting to the server, (5) Both devices are configured after connection with the server, (6) The UAV honeypot is activated automatically when the UAV client reports intrusion to the server, (7) The server analyzes the reported traffic, and (8) The server generates logs and alerts against intrusion traffic on the server GUI.

For the connection among UAV, honeypot UAV and honeypot server, we use the WiFi dongle to give the concept of a real-world scenario. When our testbed is configured, we use the Kali Linux VM as an attacker and generate the attack on the client UAV. The honeypot UAV detects the attacks using the IDS set up in our system. Next, the Honeypot server marks this attack as a bait session, activates the honeypot, and transfers all traffic to the Honeypot UAV. Then, all activities are recorded by generating the logs on the honeypot server for further investigation and analysis. Figure 7. is shown the flow of testbed as follows: (1) The Server is run successfully, (2) The UAV client is activated by using the link and reporting to the server, (3) The attacker performs an attack on the UAV client, (4) The UAV honeypot is activated automatically when the UAV client reports intrusion to the server, (5) The server analyzes of the reported traffic, (6) The server generates logs against intrusion traffic on Server GUI.

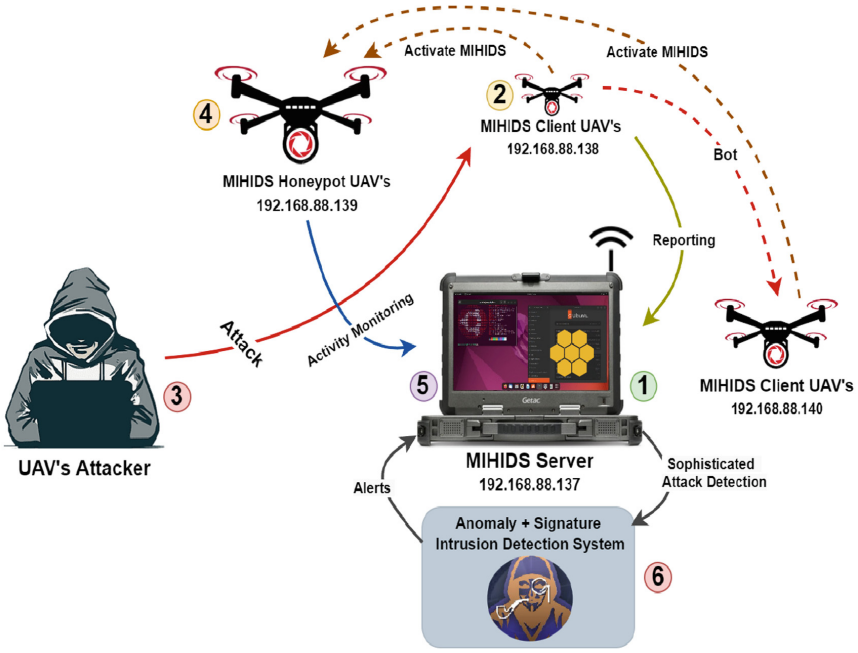


Fig. 7. Testbed Flow Diagram

5 Implementation Result and Analysis

MIHIDS is comprised of several key components, including the MIHIDS Server, MIHIDS Honeypot UAV, MIHIDS Client UAV, Attacker Model, Detection of Attacks, and Alert Generation, Real-Time Monitoring and Attacks Classification.

5.1 MIHIDS Server

The first step is to install the prerequisite libraries: Python 2.7, Flask library, SSL library, and MySQL DB for configuration. The MIHIDS server set-up is installed after the required libraries was installed. After the installation and configuration were completed, the MIHIDS server is activated by executing the script. The server is successfully running at IP address 192.168.88.137 on port 5000. After that, by adding an SSL certificate for the secure server dashboard, the server will generate a unique encrypted password to access the dashboard, as shown in Fig. 8.

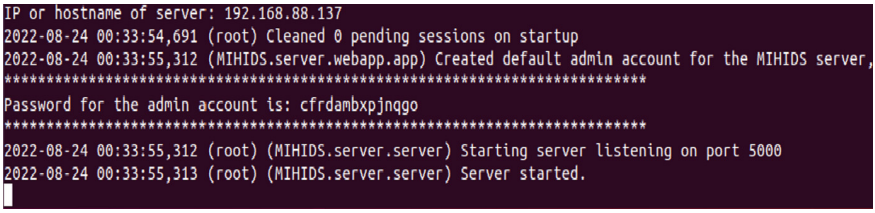


Fig. 8. Server Activated

When the MIHIDS server dashboard is configured successfully, it will generate the unique link for establishing the connection between the honeypot UAV and the client UAV, as shown in Fig. 9. Furthermore, the MIHIDS server will start monitoring the client UAV. If an intrusion is detected, it will generate an alert on the server dashboard.

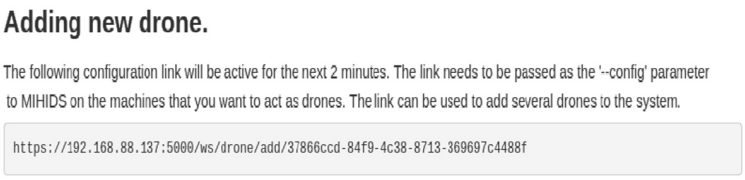


Fig. 9. Server Generates a Unique Link

5.2 MIHIDS Client UAV

The second step is to set up the MIHIDS client on the Ubuntu VM. When the configuration and installation was completed, the client UAV establishes a connection with the server using the unique link generated by the MIHIDS server. When the connection is established, the client UAV (IP address 192.168.88.138) is added to the MIHIDS server for monitoring the client UAV network traffic. After that, the client UAV forwards all traffic to the MIHIDS server to monitor UAV intrusions and attacks. Figure 10 shows that the client UAV is configured, and ready for listing network traffic on the specified configured port.

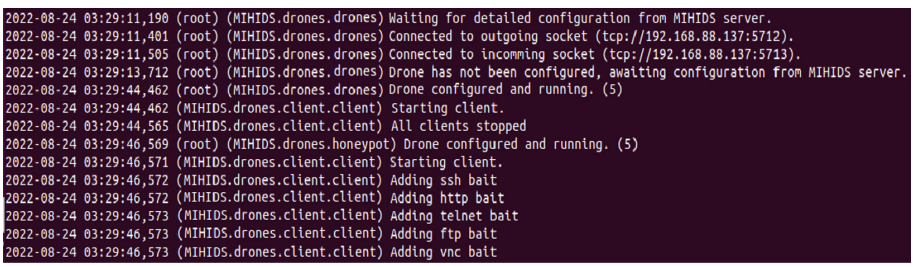


Fig. 10. Client UAV Started

5.3 MIHIDS Honeypot UAV

The third step is to set up the MIHIDS honeypot on the Ubuntu VM. When the configuration and installation is completed, the honeypot UAV establishes a connection with the server using the unique link generated by the MIHIDS server. When the connection is established, honeypot configure the required ports (HTTP, MAVLINK, SSH, TELNET, VNC, and FTP). The SSL certificate is added to the MIHIDS honeypot UAV (IP address 192.168.88.139). After that, they can start recording the attacker's activities and lure the attacker. Figure 11 shows that the honeypot UAV is configured, and ready for listing network traffic on the specified configured port.

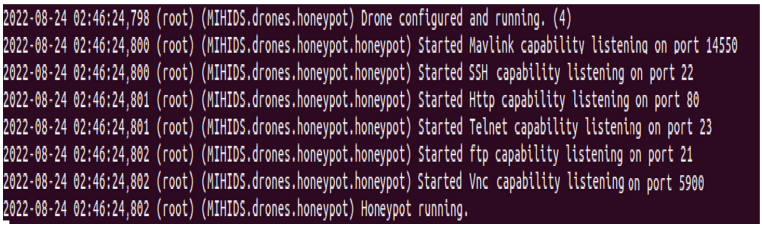


Fig. 11. Honeypot Start Listening

5.4 Attacker Model

The Kali Linux VM is used as the attacker model, and the tools used for the attacks are Hydra and Aircrack-ng. Using Kali Linux's Hydra tool, users can brute-force usernames and passwords for various services, including FTP, SSH, TELNET, and MS-SQL. Figure 12 shows that the attacker performed the attack on UAV.

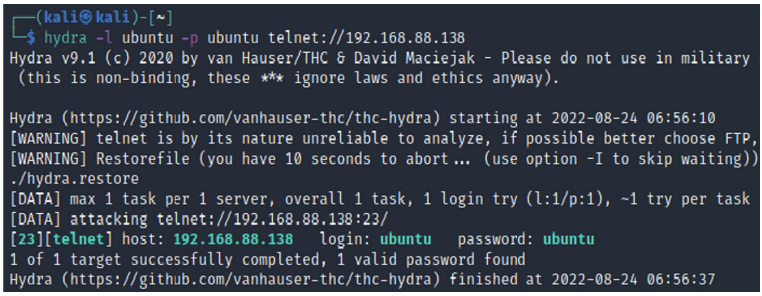


Fig. 12. Aircrack-ng Launch DE-Authentication Attack

Aircrack-ng is used for DE-Authentication attacks. DE-Authentication attacks are a kind of DoS attack that targets communication between a client

UAV and a MIHIDS server. This attack sends disassociated packets to one or more client UAVs currently connected to the access points. Figure 13 shows that the attacker performed the DE-Authentication attack on UAVs.

```
root@kali:~# aireplay-ng --deauth 0 -c 98:5F:D3:4A:B1:31 -a C4:E9:84:3F:26:04
wlan0mon
21:36:31  Waiting for beacon frame (BSSID: C4:E9:84:3F:26:04) on channel 1
21:36:31  Sending 64 directed DeAuth. STMAC: [98:5F:D3:4A:B1:31] [ 1|51 ACKs]
21:36:32  Sending 64 directed DeAuth. STMAC: [98:5F:D3:4A:B1:31] [ 0|52 ACKs]
21:36:32  Sending 64 directed DeAuth. STMAC: [98:5F:D3:4A:B1:31] [ 0|47 ACKs]
21:36:33  Sending 64 directed DeAuth. STMAC: [98:5F:D3:4A:B1:31] [20|49 ACKs]
21:36:33  Sending 64 directed DeAuth. STMAC: [98:5F:D3:4A:B1:31] [24|48 ACKs]
21:36:34  Sending 64 directed DeAuth. STMAC: [98:5F:D3:4A:B1:31] [ 1|52 ACKs]
21:36:34  Sending 64 directed DeAuth. STMAC: [98:5F:D3:4A:B1:31] [ 0|53 ACKs]
21:36:35  Sending 64 directed DeAuth. STMAC: [98:5F:D3:4A:B1:31] [ 1|53 ACKs]
21:36:36  Sending 64 directed DeAuth. STMAC: [98:5F:D3:4A:B1:31] [ 6|48 ACKs]
```

Fig. 13. Aircrack-ng Launch DE-Authentication Attack

5.5 Detection of Attacks

Attacks are detected by IDS and displayed on the MIHIDS server dashboard. Attacks were detected on various enabled ports (HTTP, MAVLINK, SSH, TELNET, VNC, and FTP). There are 3 columns in the attacks recorded table. (i) Capability indicates the ports exploited during attacks, (ii) Attacks indicates the number of attacks detected, and (iii) % indicates the percentage of attacks detected on enabled ports. The results of attack detection shown below are obtained in Fig. 14.

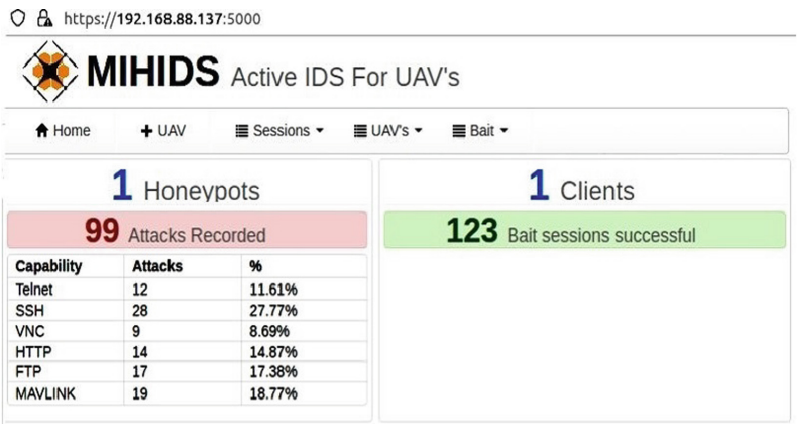
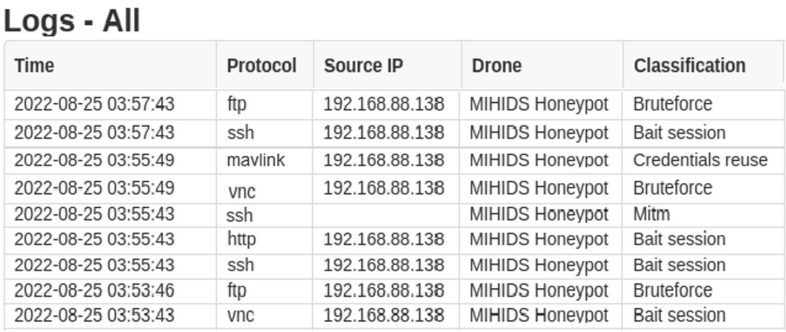


Fig. 14. Server Detect the Attacks

5.6 Alert Generation, Real-Time Monitoring and Attacks Classification

The MIHIDS server generates several distinct attack alerts, including brute force, credential reuse, and MitM attacks on various ports. Our proposed solution also includes a real-time monitoring system and the classification of attacks. The MIHIDS server is used to analyze the attack logs, and monitor the traffic with the help of real-time monitoring on the GUI. Figure 15 depicts the overall logs of real-time monitoring and attack classification. (i) Protocol denotes the exploited protocol during attacks, (ii) Source IP denotes the IP address of the compromised UAV client, (iii) Drone denotes which UAV detected the attacks, and (iv) Classification denotes the types of attacks carried out.



Logs - All

Time	Protocol	Source IP	Drone	Classification
2022-08-25 03:57:43	ftp	192.168.88.138	MIHIDS Honeypot	Bruteforce
2022-08-25 03:57:43	ssh	192.168.88.138	MIHIDS Honeypot	Bait session
2022-08-25 03:55:49	mavlink	192.168.88.138	MIHIDS Honeypot	Credentials reuse
2022-08-25 03:55:49	vnc	192.168.88.138	MIHIDS Honeypot	Bruteforce
2022-08-25 03:55:43	ssh		MIHIDS Honeypot	Mitm
2022-08-25 03:55:43	http	192.168.88.138	MIHIDS Honeypot	Bait session
2022-08-25 03:55:43	ssh	192.168.88.138	MIHIDS Honeypot	Bait session
2022-08-25 03:53:46	ftp	192.168.88.138	MIHIDS Honeypot	Bruteforce
2022-08-25 03:53:43	vnc	192.168.88.138	MIHIDS Honeypot	Bait session

Fig. 15. Real Time Logs and Attacks Classification

6 Results, Discussion and Performance Analysis

The MIHIDS is made to be set up close to real UAV locations and potential attackers. Therefore, it is not required to handle honeypots operating on a large internet scale, but only for attackers within physical or wireless range. We still strive for high efficiency within a low power consumption to run the MIHIDS on low-power or even battery-powered devices (e.g., our system uses limited resources to run on the real device). The ultimate goal is to use real UAVs to carry the MIHIDS or to integrate the MIHIDS into a flight controller for a UAV. To determine this, we measure the CPU usage on an Ubuntu 18.04 system with 2GB of RAM. After successful deployment and implementation, the system is tested to evaluate its performance. Four significant factors were used for the performance analysis of our system. When an attack occurs, the alert is generated, and each prediction outcome of our framework has been classed as a true positive (TP); a true negative (TN), when there is no attack and no alert is generated; a false positive (FP), occurs when an attack is not detected but the alert is generated; a false negative (FN), occurs when there is no attack but no alarm is generated. These factors can be used for finding statistical parameters like accuracy. The accuracy of a model is the number of correct predictions made

by the model, and correct predictions indicate higher accuracy. The following equation can be used to calculate accuracy:

$$Accuracy = \frac{Correct\ Predication}{Total\ Cases} * 100 \tag{1}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100 \tag{2}$$

In our case;

$$Accuracy = \frac{Total\ Attacks\ Detected}{Total\ Attacks\ Performed} * 100 \tag{3}$$

In general, a UAV honeypot-based IDS with high attack detection efficiency and alert generation efficiency, can effectively detect and respond to cyber attacks against UAVs, helping to improve the security and reliability of these systems.

6.1 Average Detection Rate

The average detection rate for various attacks is shown in Fig. 16 with regard to the length of the probe interval. According to Fig. 16, our framework displays an average TPR of more than 93% when the probe period is 60 s. This is due to the fact that with a probe period of 60 s, our framework is able to gather enough attack frames, or more attack data with the use of several probe intervals, to possibly differentiate between various MitM, Brute-force, and DE-Authentication attacks and their variations. This shows that our algorithm performs more effectively the observation period is more prolonged.

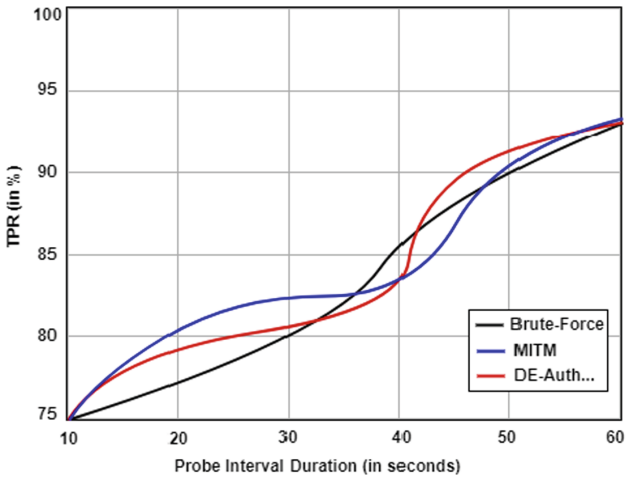


Fig. 16. Average Detection Rate at different distances with various probe interval lengths

6.2 Attacks Detection Efficiency

The attack detection efficiency of the MIHIDS refers to the ability of systems to accurately detect and identify cyber attacks against the UAV. This can be measured by the number of attacks that were successfully detected and identified by the MIHIDS system, as well as the false positive and false negative rates.

Figure 17 demonstrated the attacks detection efficiency of the system. A total 104 attacks were conducted in 4 phases, out of which 99 attacks (MitM, Brute-force, and Credential Reuse) were detected, and the MIHIDS server generated 123 bait sessions. The number of attacks increased in each subsequent phase. The bait sessions also increased according to the number of attacks. (i) In phase 1, a total 13 attacks were performed, and 10 attacks were detected. (ii) In phase 2, a total 21 attacks were performed, and 19 attacks were detected. (iii) In phase 3, a total 30 attacks were performed, and all attacks were detected. (iv) In phase 4, a total 40 attacks were performed, and all attacks were detected. Using the above-mentioned Eq. 3, we determine that the system performance accuracy is 95.19%. A performance percentage indicates that each part is qualified to address the identified issue. This shows that the proposed system is efficient enough to detect targeted attacks on specific ports.

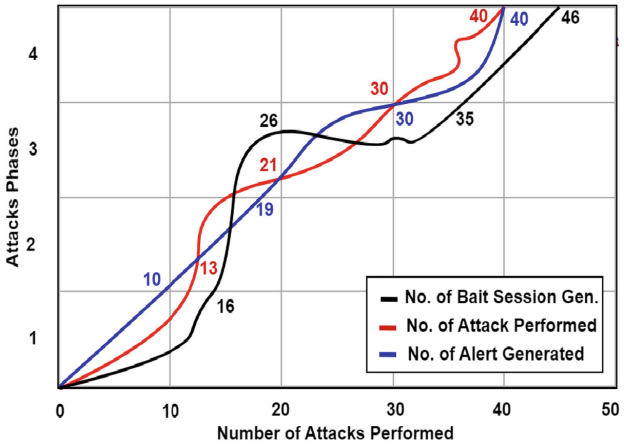


Fig. 17. Attacks Detection Efficiency

6.3 Alerts Generation Efficiency

The alert generation efficiency of the MIHIDS refers to the ability of systems to generate alerts in a timely and effective manner. This can be measured by the speed at which alerts are generated, the alerts accuracy, and the alerts effectiveness in informing relevant parties about the attack.

Figure 18 depicts the alert-generating efficiency of the system. The MIHIDS generates 99 alerts against 104 attacks. Only five alerts are not sent in the case

of intrusion detection. The five missed alerts belong to the info category. In the MIHIDS, alert has three categories: critical, medium, and info. The system can't ignore critical and medium alerts, but the info alerts system can ignore them due to false positives.

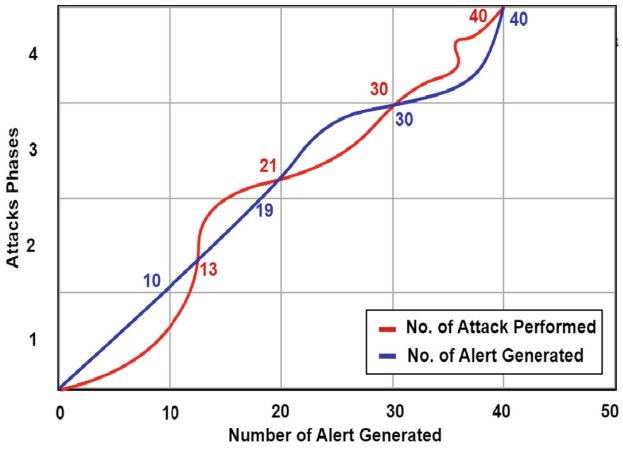


Fig. 18. Alert Generation Efficiency

7 Conclusion

This research proposes the MIHIDS, a honeypot for identifying cyberattacks in the UAV ecosystem. The MIHIDS honeypot-based IDS focuses exclusively on UAV protocols (FTP, SSH, TELNET, MAVLINK). It is also lightweight, so it can be used in small, inexpensive devices and is therefore simple to deploy in various devices (e.g., attached to a UAV, Raspberry Pi). We have examined the effectiveness of the honeypot, and performed several experiments demonstrating the MIHIDS capacity to respond to attack scenarios as close to reality as possible. Additionally, this research aims to evaluate the effectiveness of the honeypot-based IDS, and obtain the most accurate information/alerts about the cyber-attacks. Finally, we intend to examine the performance of the honeypot and its capacity for managing numerous connections simultaneously. In terms of future work, we intend to further enhance the MIHIDS by concentrating on enhancing protocol emulation and preventing attacks.

Acknowledgement. The work is supported in part by the Wuhan Knowledge Innovation Program (2022010801010117) and Wuhan AI Innovation Program (2023010402040020) and Major Science and Technology Project of Zhejiang Province - Bilateral Industry Joint R&D Program Project (2021C04007).

References

1. Rodday, N.M., Schmidt, R.D.O., Pras, A.: Exploring security vulnerabilities of unmanned aerial vehicles. In: NOMS 2016–2016 IEEE/IFIP Network Operations and Management Symposium, pp. 993–994 (2016)
2. Pleban, J.-S., Band, R., Creutzburg, R.: Hacking and securing the AR. Drone 2.0 quadcopter: investigations for improving the security of a toy. In: Enabling Technologies, Algorithms, and Applications, Mobile Devices and Multimedia (2014)
3. Nawrocki, M., Wählisch, M., Schmidt, T.C., Keil, C., Schönfelder, J.: A survey on honeypot software and data analysis. arXiv preprint: arXiv:1608.06249 (2016)
4. Pa, Y.M.P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T., Rossow, C.: IoTPOT: analysing the rise of IoT compromises. In: 9th USENIX Workshop on Offensive Technologies (WOOT 15) (2015)
5. Rist, L., Haslinger, D., Smith, J., Vestergaard, J., Pasquale, A.: Conpot honeypot (2013)
6. Provos, N., Holz, T.: Virtual Honeypots: From Botnet Tracking to Intrusion Detection. Pearson Education, London (2007)
7. Vasilomanolakis, E.: This network is infected: Hostage-a low-interaction honeypot for mobile devices. In: Proceedings of the Third ACM Workshop on Security and Privacy in Smartphones & Mobile Devices (2013)
8. Vasilomanolakis, E., Srinivasa, S., Cordero, C.G., Mühlhäuser, M.: Multi-stage attack detection and signature generation with ICS honeypots. In: NOMS 2016–2016 IEEE/IFIP Network Operations and Management Symposium (2016)
9. Hadi, H.J., Sajjad, S.M., un Nisa, K.: BoDMitM: Botnet detection and mitigation system for home router base on MUD. In: 2019 International Conference on Frontiers of Information Technology (FIT) (2019)
10. Smith, A.: Bluepot: Bluetooth honeypot (2013). https://github.com/andrewmichaelsmith/bluepot
11. johnnykv/heralding: Credentials catching honeypot. https://github.com/johnnykv/heralding
12. foospidy/HoneyPy: A low to medium interaction honeypot. https://github.com/foospidy/HoneyPy
13. CanadianJeff/honeywrt. https://github.com/CanadianJeff/honeywrt
14. Michel Oosterhof. Cowrie honeypot. https://github.com/micheloosterhof/cowrie (2014)
15. Klein, J.C.: Kojoney2 honeypot. https://github.com/madirish/kojoney2
16. Dang, F.: Understanding fileless attacks on Linux-based IoT devices with HoneyCloud. In: Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services (2019)
17. Wang, Y., Su, Z., Benslimane, A., Xu, Q., Dai, M., Li, R.: Collaborative Honeypot defence in UAV networks: a learning-based game approach. arXiv preprint: arXiv:2211 (2022). 01772
18. Su, Z., et al.: Collaborative Honeypot defence in UAV Networks: a learning-based game approach (2022)
19. Hadi, H.J., Cao, Y., Nisa, K.U., Jamil, A.M., Ni, Q.: A comprehensive survey on security, privacy issues and emerging defence technologies for UAVs. J. Netw. Comput. Appl. **213**, 103607 (2023). https://doi.org/10.1016/j.jnca.2023.103607
20. Hadi, H.J., Cao, Y.: Cyber attacks and vulnerabilities assessment for unmanned aerial vehicles communication systems. In: 2022 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, pp. 213–218 (2022). https://doi.org/10.1109/FIT57066.2022.00047