Security Threats Analysis of the Unmanned Aerial Vehicle System

Rune Hylsberg Jacobsen
Department of Electrical and Computer Engineering
Aarhus University, Aarhus, Denmark
ORCID: 0000-0001-9128-574X

Ali Marandi
Department of Electrical and Computer Engineering
Aarhus University, Aarhus, Denmark
ORCID: 0000-0002-2290-4495

***Abstract*—Autonomous Unmanned Aerial Vehicles (UAVs) find increasing use in the civil airspace where multi-UAV systems are deployed to perform operations supervised by remote control facilities. Since these multi-UAV systems operate in public locations and connect using open communication standards, it raises significant security concerns. While most UAV designers befittingly deal with the complexity of autonomy, communication, and control, they often ignore to consider system security aspects in the early design phase. To support the security design process of multi-UAV systems, this paper provides an analysis of cybersecurity threats of multi-UAV systems based on the STRIDE model. The identified threats are subsequently linked with a risk assessment to be prioritized according to severity. We conclude by establishing a set of security design recommendations for connected inspection UAVs to contribute with guidance for the design of future multi-UAV systems.**

***Index Terms*—Unmanned aerial vehicle, drone, threat model, risk assessment, security management.**

I. Introduction

Breakthrough in research and innovation in Unmanned Aircraft Systems (UASs) has led to a significant increase in the deployment of autonomous Unmanned Aerial Vehicles (UAVs) in the civil airspace. UAVs also referred to as drones, are unmanned aircraft that perform tasks such as surveillance, monitoring, search and rescue, military use, and infrastructure inspections [1]. Security threats to UAVs are typically targeted at the system level, which comprises all functions employed to allow the UAV to operate. This may include the hardware and software running on the UAV, the Ground Control Station (GCS), the support Cloud Services (CS), and the connections between these entities. Cyberthreats are further accentuated by UAVs connecting to open, wireless networks with access to the Internet [2], [3].

The urgency of addressing the security of connected UAVs can hardly be exaggerated. In 2009, the media reported that a US military UAV had been hacked by Iraqi insurgents that intercepted live video feeds from a UAV and reveal potential targets [4]. Other known attacks to the UAS include computer virus infection of the mission command center of a US air force base controlling military UAVs [5]. Best et al. [3] introduced a framework for understanding and documenting vulnerabilities and attack opportunities of the UAS. It was found that most of the cyberattacks documented across different types of sources use either Denial of Service (DoS) or spoofing attacks. These attacks target open networks of the UAS and use radio frequencies to overpower the original owner's signals.

There is a good knowledge base concerning threats related to TCP/IP communication and vulnerabilities on the Internet are continuously monitored by Computer Security Incident Response Team (CSIRT) [6]. The increasing use of Robot Operating System (ROS) for autonomous systems exposes a new set of communication interfaces [7]. A software bug may result in a vulnerability that can be exploited by an adversary. Furthermore, "backdoors" likely exist in the software when executing unknown code may impact several security implications among others the risk of executable software having some undocumented, unwanted, or hidden functionality [8].

In an ideal world, the goal would be to eliminate all security risks of a system. However, designing for full security is costly. Therefore, system threats need to be prioritized according to the severity of the risks they pose. The assessment of security risks allows an organization to align system design with the security objectives and to focus on threats that pose the highest severity. This paper aims to provide a security threat analysis of an Internet-connected multi-UAV system. The analysis will help UAV manufacturers and system integrators to address important security concerns for their designs. We provide an approach to assess cybersecurity threats that bases on a taxonomy of the UAS accompanied by a data flow model to clarify behavioral aspects. We apply the STRIDE model to reveal the most important threats to the system and point to technologies suitable for mitigating these threats.

II. Related Works

The European Union Agency for Cybersecurity (ENISA) is the Union's agency dedicated to achieving a high common level of cybersecurity across Europe. Among others, the agency contributes to EU cyber policy, by identifying and evaluating the top cyber threats regularly to enhance the trustworthiness of Information and Communication Technology (ICT) products, services, and processes [2]. To this end, ENISA has addressed thematic threat landscapes related to Internet infrastructure, smart grids, and 5G networks. However, a threat

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 861111.

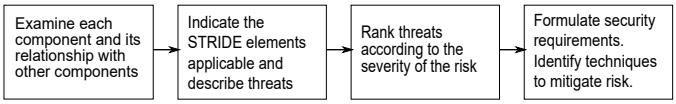

Fig. 1. Method of the overall security assessment.

landscape specifically targeted UAS and multi-UAV systems seems to be missing.

In [9], Whelan et al. analyzed security threats to an autonomous UAS. The work focused on a vulnerability assessment of the non-autonomous and long-range autonomous UAS. It was stated that the in-field placement of GCS opens a new range of attacks that are not typical in most UAS deployments [9]. Furthermore, Ly and Ly [10] offered a review of the cyberthreats of UAVs. Their study used the STRIDE threat model to classify cyberattacks. It was concluded that the most common and effective forms of a cyberattack on UAVs are spoofing and DoS attacks.

Javaid et al. [11] discussed various security threats to a UAV system. Threats were analyzed and a threat model showing possible attack paths was presented. In their work, the security triad, with its three facets: confidentiality, integrity, and availability was used as the overarching security model. It was further argued that from a security and threat analysis perspective, it is necessary to understand that a typical UAV network is not similar to the traditional wireless networks such as Wireless Sensor Networks (WSNs) and Mobile Ad-hoc Networks (MANETs) because of similar power and data rate requirements.

In this paper, we provide a review of the security threats of the UAS. We link our analysis to a system taxonomy to model structural parts of the UAS and combine this description with a data flow model to identify interactions between system parts. Our analysis furthermore showcases the applicability of the STRIDE methodology for multi-drone threat analysis.

III. Methodology

Threat modeling is the process of identifying and risk-rating adversaries and their attacks [12]. Essentially, it divides into two equally important tasks: threat identification and risk management. Our security assessment follows four steps (Fig. 1). First, we analyze the multi-UAV system to derive a taxonomy describing its structure and its data flow to examine the relationship between subsystems by following the data in the system. Second, we analyze system threats using STRIDE threat modeling, which is a known technique for analysis of computer security threats at first [13]. Third, we apply a risk assessment to prioritized threats. Finally, we outline the most central security objectives for the system and briefly point to technologies suitable to mitigate the identified threats.

For risk assessment, threats are analyzed concerning their likelihood of occurrence, their possible impact on individual users and the system, and the global risk they represent adhering to a standard evaluation methodology [11]. The evaluation is conducted according to three criteria: *likelihood*, *impact*, and *severity*. The *likelihood* score evaluates the possibility of attacks being initiated. The *impact* score ranks the resulting state of the system after an attack. The risk *severity* is calculated as the product of the *impact* and *likelihood* values for a given threat. In summary, Table I shows the used mapping of the qualitative assessment of risks to the corresponding rank of the risk levels.

TABLE I
Risk evaluation grid. Adapted from [11].

Criteria	Cases	Rationale		Rank
Likelihood	Unlikely	Strong	Low	1
	Possible	Solvable	Reasonable	2
	Likely	None	High	3
Impact	Low	Annoyance	Very limited outage	1
	Medium	Loss of service	Limited outage	2
	High	Long time loss of service	Long term outage	3
Severity	Minor	No need for countermeasures		1-2
	Major	The threat needs to be handled		3-4
	Critical	High priority		6-9

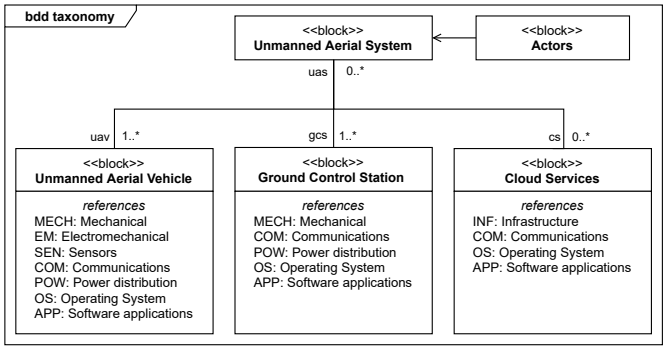

Fig. 2. Block definition diagram for a UAS taxonomy.

IV. System Model

To identify essential subsystems and their interactions, we taxonomize the UAS and present a data flow model. This leads to an identification of the attack surface of the system [13].

A. System Taxonomy

The UAS for inspection will be deployed as a set of collaborative UAVs. During operation, each UAV is assigned a sequence of tasks, e.g., inspecting, traveling, or charging. UAVs will be monitored by using one or more GCSs. The inspection mission is supported by digital services deployed in a cloud infrastructure. Services include mission planning and supervision as well as storage of mission data. Fig. 2 presents a taxonomy of the UAS and models its main subsystems. The key actors of the system are pilots, supervisors, field engineers, the surrounding environment, and external service providers.

We consider rotary-wing UAVs for inspection due to their high maneuverability. A UAV is composed of mechanical (MECH) parts such as a body frame, electrical motors, propellers, and antennas. It is further equipped with sensors (SEN), e.g., Inertial Measurement Units (IMUs), a Global Navigation Satellite System (GNSS) receiver, magnetometers,

and cameras to support navigation and inspection. An Operating System (OS) abstraction including flight controllers and an Onboard Computer (OBC) platform support the running of software applications (APP). The OS abstraction features a ROS middleware supporting the control functions [14]. Furthermore, a power distribution system (POW) ensures power to be delivered to relevant system parts. It is important to note that UAVs are constrained concerning memory, computation, and battery capacity [15].

The multi-UAV system forms a wireless mesh network to support collaboration between UAVs [16]. The UAV mesh network also acts to extend the coverage of the inspection area. Each UAV is connected to a GCS for supervisory control through the Command and Control (C2) channel using long-range wireless communication [16]. Although always wireless, different mediums such as Radio Frequency (RF), mobile network, and wireless mesh networking are used to support a variety of communication interfaces (COM). The network setup allows telemetry and payload data to be routed over multiple hops in the network. Furthermore, the GCS is an IP node that allows a multi-UAV system to connect to the Internet. Like the UAVs, the GCS is composed of MECH parts such as a ruggedized computer platform, antennas, and possibly also a display. Also, the Operating System (OS) supports the running of applications (APP) on the GCS. The GCS will typically be battery-powered (POW) to support field operation.

The continuous collection of high-resolution inspection images requires a large and resilient data storage capacity. Furthermore, running data analysis algorithms requires an up-scaling computing infrastructure. To enable this, a cloud infrastructure (INF) is used for data storage and processing. It enables the display of mission data during the inspection through the deployment of applications. Access to the CS is provided over an IP network infrastructure (COM). This allows us to deploy cybersecurity protection schemes from the Internet society such as firewalls, intrusion detection systems as well as ciphers and protocols for protecting data. The cloud infrastructure stores sensitive information about the customer, their missions, and the payload results. For this reason, the security and privacy of the cloud is an important aspect of the UAS threat analysis.

B. Data Flow Model

Data Flow Diagrams (DFDs) have gained wide acceptance in threat modeling to describe how data flows through a system despite its limitations [17]. The DFD specifies how processes act upon data and where data is stored. The trust boundaries have been indicated with red dashed lines (Fig. 3). An outer-trust boundary marks interfaces to external actors (shown with rectangles), while inner-trust boundaries represent different privilege levels in the system. The UAV will receive command input from the *pilot* actor and through sensing of the *environment* context including input for navigation. Flight control and sensory information are stored locally. The inspection application controls the recording of images and sends these to the CS for storage via a communication relay function in the GCS. The UAV, which may be guided by a *Supervisor* actor, also receives control information from the GCS and returns telemetry information. Mission control data is received from CS running the mission control function. Data relevant to the mission are cached in the GCS. Inspection images are stored in the cloud storage alongside relevant telemetry and mission data. To support the inspection mission, the CS will make use of data from *External Services* through an Application Programming Interface (API).

C. Attack Surface

The attack surface represents data interfaces that cannot restrict data access to a sufficient degree. These interfaces may be subject to cyberattacks [13]. Keeping the attack surface small is a basic security measure. The attack surface of the UAS is defined by attacks made possible by 1) gaining physical access to a UAV or a GCS, 2) intercepting or otherwise exploiting a communication interface, and 3) manipulation with the software installed. Cyberattacks on communication interfaces can take place at different layers of the protocol stack. As the multi-UAV system connects to the Internet, among others to get access to CSs, the system is exposed to vulnerabilities associated with communication over IP networks [6].

V. Threat analysis

In the following, we discuss the outcome of our STRIDE analysis.

A. Spoofing

The multi-UAV system implements a swarm where UAVs dynamically join and leave the swarm. A malicious UAV could make a spoofing attack and join the swarm. This would allow the malicious UAV to eavesdrop on the UAV-to-UAV communication, making a man-in-the-middle attack. Furthermore, a malicious UAV, which has gained access to the multi-UAV network, will be able to inject false/corrupted data that could disturb coordination functions of the swarm. For instance, a formation-flying maneuver could be destroyed, or the swarm could be tricked into believing that other UAVs are in false positions. Unfortunately, detecting a malicious UAV in an autonomous swarm is a complicated task that may implement sophisticated learning and clustering algorithms [18].

Another spoofing threat arises from a malicious GCS pretending that it is a legitimate GCS. The malicious GCS could eavesdrop on telemetry data or intercept the C2 channel to take control over a UAV. Moreover, the malicious GCS would potentially be able to inject false telemetry data into the system tricking the mission supervision function with false status and position data.

Spoofing of the GNSS signal may occur when an adversary emits a stronger GNSS signal with false location information. In [19], the threat to unmanned vehicles, guided by Global Positioning System (GPS) receivers to spoofing threats, was analyzed. It was noted that the spoofing of the GPS-based

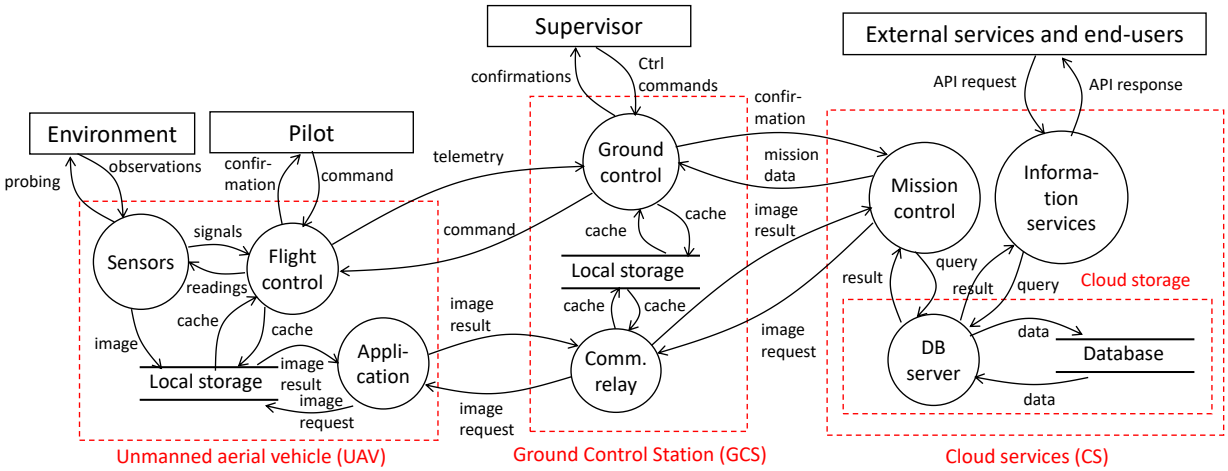

Fig. 3. Data Flow Diagram (DFD) for the UAV. The DFD is simplified to show only one UAV and one GCS.

navigation is more threatening than intentional jamming since the receiver cannot detect an attack [19]. Kerns et al. [20] demonstrated that an attacker who controls critical sensor measurements made by a UAV has great authority over the system. The study presented requirements for overt and covert capture of the navigation system. As a further demonstration of the GPS spoofing threat, Shijith et al. [21] used a manipulated GPS signal, which resembles the original signal, to guide and land a UAV at the desired location of an adversary. This provided a relatively easy attack using a GPS signal-simulator based on a software-defined GPS receiver.

The distributed network of entities running ROS nodes exposes a middleware-based communication infrastructure susceptible to cyberattacks. ROS coordinates communications between the hardware and operational software on the UAVs using message-passing. ROS-based systems comprise "nodes" that communicate by publishing "messages" to different "topics," to which other nodes may listen. A recent analysis shows that ROS lacks several security enhancements to make it suitable for industrial use [22]. It is noted that under the "restriction of not touching the OS system internals, we cannot easily exclude a publisher or subscriber [in the network], nor can we preclude a new publisher being started and replacing the existing one" [22]. If a malicious ROS node (software) can take part in the ROS network, it may eavesdrop or inject false information into the network. False information could fake the control system of the UAV to believe that the battery level is critically low, which would trigger a safety protocol. In [23], the authors re-purposed the mobile sensor node to serve as a surrogate for a car-like robotic system to emulate the cyber-physical challenges associated with deployed mobile robotic systems based on ROS. It was demonstrated that spoofing attacks were possible using low-cost, low overhead, cyberattacks on a robot implementation.

Finally, since access to CS is obtained over the Internet, the system is vulnerable to spoofing attacks addressing layers 2 to 4 of the Open Systems Interconnection (OSI) model. An adversary could run malware over the Internet and eavesdrop on communications, inject false information in the network, or otherwise attack the availability of the system.

B. Tampering

As UAVs and GCSs are out in the inspection area with no or limited physical security, they are vulnerable to physical capture and tampering attacks [15]. Tampering attacks could damage the mechanical parts, sensors, power systems, etc. For instance, destroying a camera or breaking an antenna would significantly inhibit the UAVs' ability to operate autonomously. Since UAVs and GCSs are valuable assets, there is furthermore a risk of theft.

Suppose an attacker has got access to the communication network of the UAS, there is a risk that data is compromised. Inspection data, i.e., images, telemetry, etc., and log files can be faked. While corrupting mission data would sabotage the inspection operation it will be relatively easy to detect. In contrast, the corruption of log files could hinder the auditing of an inspection operation. An adversary would potentially be able to erase evidence of an intrusion made. Since data is spread across multiple subsystems, an adversary would need to gain access to all relevant subsystems to make the attack hard to detect. Moreover, an adversary would be able to tamper with the software running on the UAVs to install malware, i.e., viruses, Trojans, key-loggers, and botnets, or simply delete the software.

For all data sources, there is a risk of data being corrupted in the data storage from a side channels attack by an adversary, e.g., by using a source of electromagnetic radiation, excessive heat, or accidentally by a surge of electricity from a power cable. Since the UAV and the GCS to a large extend are built from 3rd party hardware and software components, a vulnerability exists from badly made software and hardware introducing ways to exploit the system. In addition, a malicious manufacturer could implement interception interfaces to eavesdrop on the communication or to allow malware to be installed.

A tampering attack may arise from the modification of data passed by a malicious UAV when a packet is transmitted along

the routing path in the UAV swarm [18]. A change in the routing information would possibly lead to UAVs getting their traffic hijacked whereas dropping of routing information could inhibit data from getting to the GCS. Furthermore, replay attacks arising from copying and resending routing information would consume communication bandwidth, waste the power of the UAV to make a DoS attack on the network [18].

Another class of tampering attacks comes from changes made to the environment. Some algorithms, running in the UAV, support navigation from recognition of specific features in the environment. For instance, visual odometry can determine the position and orientation of the UAV by analyzing images from onboard cameras [24]. Tampering with the physical environment could potentially disturb the navigation of the UAV leading in a wrong direction.

C. Repudiation

Although inspection operations aim to be autonomous, there will be a mission supervisor and possibly also a pilot actor to oversee the operation. A repudiation attack occurs if actors deny having performed certain interventions or deny having configured the system in a specific way possibly hiding misconduct. For instance, a pilot may attempt to deny taking control over a UAV that caused an accident or the pilot may deny that certain safety protocols were ignored.

A UAV engineer may attempt to deny having installed a specific software or made a specific configuration of a UAV and in the GCS. This could for instance be to deny not having run the system with a sufficient degree of security. A UAS organization, represented by a supervisor actor, could deny a certain sequence of events related to the autonomous mission to inhibit investigators being able to determine the cause of an accident, e.g., a fatal crash.

Concerning mitigation of repudiation attacks, the use of blockchain technology seems promising [25].

D. Information Disclosure

An adversary intercepting communications in the UAS will be able to eavesdrop on data exchanges and thereby obtaining knowledge on how the system is operating. Eavesdrop attacks could also give the adversary access to telemetry data and thereby information about the whereabouts of the UAVs and their recordings. In particular, it is difficult to detect a passive eavesdrop attack as was the case for the US military in Iraq [4]. Passive eavesdrop attacks could be launched from the deployment of malicious UAVs or GCSs. Since the multi-UAV system is connected to an IP infrastructure, the UAS is furthermore vulnerable to information disclosure from malicious remote users intercepting IP traffic.

Another threat to information disclosure exists for the data storage. In particular, the CS will hold information related to the inspection missions and their progress as well as the inspection images themselves. Part of this data may be cached on the UAVs or GCSs. Software on a hijacked UAV can be copied and reverse engineered. This allows an adversary to disclose how the system is being built.

E. Denial of Service

Jamming of communications and scrambling/distortion of signals are threats to most wireless control systems. Jamming is achieved by introducing a source of noise strong enough to significantly reduce the capacity of the channel. Jamming may be either unintentional or malicious. In addition, we may consider electromagnetic interference from power cables as a potential unintentional source of a jamming attack. Since jamming is easy to detect and address, we believe that it can have a medium impact on both the user and system.

Jamming of the GNSS signal will impact the system navigation possibly leading the system to a wrong location or tricking a safety procedure. Although more difficult than jamming, successful GNSS spoofing can cause a UAV to go off course, crash, or be hijacked by an adversary.

Another type of DoS attack arises from tricking the UAV sensors with false input. For instance, placing obstacles in front of the UAV would trick the obstacle avoidance algorithm resulting in a stop of the UAV.

The UAS may suffer from attacks launched from the Internet such as botnet attacks [26]. Botnet attacks can be used to perform Distributed Denial of Service (DDoS) attacks, steal data, and inject messages into the network that allow the adversary to access the device and its connections.

UAVs may suffer from a lack of energy to continue their operations. Similar to WSNs, that may suffer sleep deprivation attacks, UAVs could be exposed to "denial of charging" attacks. Such attacks may be intentional or unintentional. Intentional attacks could deny the UAV's physical access to the charging point. Unintentional attacks may be a result of bad algorithms not being able to find and attach to the charging point. If a UAV is hindered sufficient charging it may become useless for further operation. Furthermore, DoS attacks based on information flooding can create network congestion, which may result in disrupting the operation of multi-UAV systems. An attacker can send a lot of SYN packets to cause buffer overflow at the UAVs besides creating network congestion [27]. The work in [27] proposes an algorithm that leverages Bayesian inference to detect flooding attacks such as SYN flooding attacks, Route Request flooding attacks, and Hello flooding attacks.

F. Elevation of Privileges

Elevation of privilege is a typical "step-stone" in an attack vector used to leverage other cyberattacks to exploit sensitive data, disrupt operations and construct back-doors for future attacks. The OS deployed as part of the UAS (cf. Fig. 2) will typically be based on a Linux distribution and follow the Unix system security model [28]. Any process under UNIX is executing with a certain user ID and group ID and derives its set of privileges accordingly. Privilege escalation implies that users receive unauthorized privileges that may be exploited to modify files, steal sensitive information, and install malware. Obvious privilege escalation vulnerabilities include weak passwords and stealing of session cookies in the communication. As the UAV is essential "a flying computer",

software security vulnerabilities privilege escalation attacks can apply to a UAV, to leverage malware infections.

An adversary may get access to the cloud storage holding mission data and could potentially take over control and be hijacking the UAVs. Similarly, an adversary getting access to the GCS or access to the flight control process could get control of the system. Web services deployed in the CS are vulnerable to Cross-site Scripting (XSS) attacks. XSS is an elevation of privilege attack exploiting the users' trust in a web server. When successful, it allows malicious code to run with more access rights than a page from the attacker's domain thereby violating the user's origin-based security policy.

Mitigation strategies for privilege escalation attacks include system call monitoring [29] and software execution prevention, i.e., by running applications in a trusted computing environment [30].

VI. Discussion

Table II summarizes the results of our threats analysis and lists the associated risk levels. Risk levels are quantified based on the assumption that proper protection mechanisms (cf. 'Security mitigation' column) have been implemented. Details of these mechanisms are beyond the scope of this paper.

Our analysis confirms the observation that spoofing and denial of service attacks are among the more severe threats to the multi-UAV system [10]. It is further evident that the multi-UAV system needs a mechanism to observe threats, monitor intrusions, and react to cyberattacks following carefully designed safety protocols to be activated when an attack is observed. It is also seen that not only adversaries, in the normal sense of hackers or aggrieved internal users, are important sources of threats but also the manipulation with the surrounding environment plays an important role.

A. Security Design Recommendations

Based on our analysis (Table II), we summarize below the essential security mitigation techniques that must be implemented to counter the threats of the highest risk level.

1) Message origin authentication: Mechanism of authentication of entities, i.e., UAVs, GCSs, and CSs in the network must be implemented.

2) Message authentication and integrity protection: Message exchanges between entities must be authenticated and integrity protected to ensure data can be trusted.

3) Confidentiality protection: Encryption of mission data is required to ensure the privacy of the inspection mission.

4) Secure logging: A method to provide secure and tamper-evident logging of actions including security audit trails is needed to mitigate repudiation threats.

5) Trusted computing: The system should implement a trusted execution environment for software. It is advisable to implement intrusion detection techniques to continuously monitor the UAS.

The commercial viability of a UAS inspection solution will ultimately depend on the level of trust that can be provided to the end-users. The awareness, prioritization of threats, and insight into possible mitigation and protection mechanisms are key to a successful UAS design.

VII. Conclusions

There is a growing need to address the security of the UAS as the uptake of autonomous UAV services is accelerating. This paper has provided a security threat analysis of multi-UAV systems in the context of an autonomous inspections. It was found that the landscape for security threats not only embrace well-known threats from ICT systems but is also hampered by UAVs operating in a hostile environment and replying on constrained resources. We have provided security design recommendations and pointed to a set of security mitigation techniques to counter the most severe threats to the multi-UAV systems. The most severe threats cover spoofing and DoS attacks. However, concerning security in the design process, techniques such as tamper-evident logging, intrusion detection, and establishment of safety protocols are as important.

References

[1] N. Iversen, O. B. Schofield, L. Cousin, N. Ayoub, G. Vom Bögel, and E. Ebeid, "Design, integration and implementation of an intelligent and self-recharging drone system for autonomous power line inspection," in *Proc. of IEEE/RSJ Int. Conf. on Intell. Robot. and Syst. (IROS)*, 2021.

[2] "The European Union Agency for Cybersecurity (ENISA), ENISA Threat Landscape 2020," https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends, accessed: 2021-05-31.

[3] K. L. Best, J. Schmid, S. Tierney, J. Awan, N. M. Beyene, M. A. Holliday, R. Kahn, and K. Lee, *How to Analyze the Cyber Threat from Drones: Background, Analysis Frameworks, and Analysis Tools*. RAND, 2020.

[4] T. Guardian, "US drones hacked by Iraqi insurgents," https://www.theguardian.com/world/2009/dec/17/skygrabber-american-drones-hacked, accessed: 2021-04-22.

[5] ——, "Computer virus infects drone plane command centre in US," https://www.theguardian.com/technology/2011/oct/09/virus-infects-drone-plane-command, accessed: 2021-04-22.

[6] "Cyber security incidence response team," https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends, accessed: 2021-05-31.

[7] R. R. Teixeira, I. P. Maurell, and P. L. Drews, "Security on ROS: analyzing and exploiting vulnerabilities of ROS-based systems," in *Latin American Robot. Symp. (LARS), Brazilian Symp. on Robot. (SBR) and Wksp. on Robot. in Educ. (WRE)*, 2020.

[8] F. Schuster and T. Holz, "Towards reducing the attack surface of software backdoors," in *Proc. of the ACM SIGSAC Conf. on Comput. & Commun. Secur.*, New York, NY, USA, 2013.

[9] J. Whelan, A. Almehmadi, J. Braverman, and K. El-Khatib, "Threat analysis of a long range autonomous unmanned aerial system," in *Int. Conf. on Comput. and Inf. Technol. (ICCIT-1441)*, 2020.

[10] B. Ly and R. Ly, "Cybersecurity in unmanned aerial vehicles (UAVs)," *J. Cyber Secur. Technol.*, vol. 5, no. 2, pp. 120–137, 2021.

[11] A. Y. Javaid, W. Sun, V. K. Devabhaktuni, and M. Alam, "Cyber security threat analysis and modeling of an unmanned aerial vehicle system," in *Proc. of IEEE Conf. on Technol. for Homeland Secur. (HST)*, 2012.

[12] J. Steven, "Threat modeling - perhaps it's time," *IEEE Secur. Priv.*, vol. 8, no. 3, pp. 83–86, 2010.

[13] A. Shostack, *Threat Modeling: Designing for Security*, 1st ed. Wiley Publishing, 2014.

[14] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "ROS: an open-source Robot Operating System," in *Proc. of ICRA Wksp. on open source Softw.*, vol. 3, no. 3.2, 2009.

[15] T. Alladi, Naren, G. Bansal, V. Chamola, and M. Guizani, "SecAuthUAV: A Novel Authentication Scheme for UAV-Ground Station and UAV-UAV Communication," *IEEE Trans. on Veh. Technol.*, vol. 69, no. 12, pp. 15 068–15 077, 2020.

TABLE II
THREAT ANALYSIS SUMMARY FOR THE MULTI-UAV SYSTEM.

Threat description	Adversaries	Attack targets	Security mitigation	Risk levels[a]		
				Likelihood	Impact	Severity
Spoofing	Malicious UAV, GCS, ROS node	Eavesdropping	Message origin authentication, data encryption	3	1	3
	Malicious UAV, GCS, ROS node	Inject fake data	Message authentication, integrity checking	2	3	6
	Malicious actor in the field	Theft of an asset, sabotage	Physical security	3	2	6
	Malicious actor in the field	Inject fake GNSS position data	Message origin authentication, integrity check	2	3	6
Tampering	Malicious actor in the field	Theft, hijacking, or sabotage	Physical security protection, system hardening	3	2	6
	UAV, GCS, bad manufacturer, bad environment	Corrupted mission and payload data	Integrity checking	2	2	4
Repudiation	Pilot, supervisor, bad manufacturer	Deny of specific operational actions, corrupted or missing logs	Tamper-evident logs and security audit trails	3	1	3
	Pilot, engineer	Denying specific configurations and designs	Verifiable and certified software and hardware components	3	2	6
Information disclosure	Malicious actor in the field	Inception of wireless communications, eavesdropping	Data encryption and layer-2 security protection	3	2	6
	Malicious actor on the Internet	Inception of IP or ROS communication, eavesdropping	Data encryption and layer-3 security protection	3	2	6
	Malicious actor on the Internet, bad cloud provider, or malware	Compromise cloud data storage	Perimeter protection (firewall)	3	2	6
Denial of service	Malicious UAV, GCS, or ROS node	False signal injection	Message authentication	2	3	6
	Malicious actor in the field	Jamming RF signal	Safety protocol	3	2	6
	Malicious actor in the field	Jamming GNSS/GPS signal	Safety protocol, invoke secondary navigation system	3	2	6
	Malicious Internet node	Distributed DoS attack over the Internet	System hardening, perimeter protection, network monitoring, safety protocol	3	1	3
	Bad manufacturer, bad environment	Denial of charging, energy deprivation, disconnection/power-off	Safety protocol	2	3	6
Elevation of Privileges	Malicious actor, bad manufacturer	Malware infection, leak of data, disrupt operations and create back-doors	Malware protection, intrusion detection, perimeter protection	3	2	6
	Malicious actor or malware	Leverage other types of cyber-attacks	Enforcement of principle of least privileges, security audit trails, system hardening	3	2	6

[a] Risk levels are ranked as unlikely (1), possible (2), likely (3) for *likelihood* and low (1), medium (2), and high (3) for *impact* cf. Table I.

[16] L. Shi, N. J. H. Marcano, and R. H. Jacobsen, "A Review on Communication Protocols for Autonomous Unmanned Aerial Vehicles for Inspection Application," *Microprocessors and Microsystems*, vol. 86, p. 104340, 2021.

[17] L. Sion, K. Yskout, D. Van Landuyt, A. van den Berghe, and W. Joosen, "Security Threat Modeling: Are Data Flow Diagrams Enough?" in *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, New York, NY, USA, 2020, p. 254–257.

[18] S. Sun, Z. Ma, L. Liu, H. Gao, and J. Peng, "Detection of malicious nodes in drone ad-hoc network based on supervised learning and clustering algorithms," in *16th Int. Conf. on Mobility, Sens. and Netw. (MSN)*, 2020.

[19] T. E. Humphreys, "Assessing the Spoofing Threat: Development of a Portable GPS Civilian Spoofer," in *Proc. of the Inst. of Navig. GNSS (ION GNSS+)*, 2008.

[20] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via gps spoofing," *J. Field Robot.*, vol. 31, no. 4, pp. 617–636, 2014.

[21] N. Shijith, P. Poornachandran, V. G. Sujadevi, and M. M. Dharmana, "Spoofing technique to counterfeit the GPS receiver on a drone," in *Proc. of Int. Conf. on Technological Advancements in Power and Energy (TAP Energy)*, 2017.

[22] B. Dieber, S. Kacianka, S. Rass, and P. Schartner, "Application-level security for ROS-based applications," in *Proc. of IEEE/RSJ Int. Conf. on Intell. Robot. and Syst. (IROS)*, 2016.

[23] J. McClean, C. Stull, C. Farrar, and D. Mascareñas, "A preliminary cyber-physical security assessment of the Robot Operating System (ROS)," in *Unmanned Syst. Technol. XV*, vol. 8741, Int. Soc. for Optics and Photon. SPIE, 2013.

[24] M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy, "Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments," in *Unmanned Systems Technology XI*, vol. 7332. Int. Soc. for Optics and Photon., 2009.

[25] P. Mehta, R. Gupta, and S. Tanwar, "Blockchain envisioned UAV networks: Challenges, solutions, and comparisons," *Comput. Commun.*, vol. 151, pp. 518–538, 2020.

[26] T. Reed, S. Dietrich, and J. Geis, "SkyNET: a 3G-enabled mobile attack drone and stealth botmaster," in *Proc. of 5th Usenix Wksp. on Offensive Technol.*, 2011.

[27] N. Nishanth and A. Mujeeb, "Modeling and detection of flooding-based denial-of-service attack in wireless ad hoc network using bayesian inference," *IEEE Syst. J.*, vol. 15, no. 1, pp. 17–26, 2021.

[28] R. B. Reinhardt, "An architectural overview of UNIX network security," ARINC Research Corporation, Tech. Rep., 1993.

[29] S. Forrest, S. Hofmeyr, and A. Somayaji, "The evolution of system-call monitoring," in *IEEE Annu. Comput. Secur. Appl. Conf. (ACSAC)*, 2008.

[30] Trusted Computing Group TPM Working Group and others, "TCG Specification Architecture Overview," pp. 1–54, 2007, revision 1.4, accessed 2021-07-16. [Online]. Available: https://trustedcomputinggroup.org/resource/tcg-architecture-overview-version-1-4/