American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-7, Issue-10, pp-106-109
www.ajer.org
Research Paper Open Access

Security Of Unmanned Aerial Vehicles: A Literature Review And Research Directions

Caleb, Nanchen Nimyel [1*], Alfred, Nanpak Albert [2] , Odumu, Wesley [3]
[1,2,3]*Computer Engineering Department, Plateau State Polytechnic, Nigeria*
Corresponding Author: Caleb, Nanchen Nimyel

ABSTRACT: *Unmanned Aerial Systems (UAS)- air vehicles equipped with sensors and software that enable the craft to fly without a human pilot on board. This is an emerging technology with a tremendous potential to revolutionize warfare and to enable new civilian applications. Vulnerabilities of UAS being hacked exist, and each of these attacks requires a different set of skills and targets a different set of vulnerabilities. There are claims of the existence of "unhackable" UAS based on the use of formal methods in the kernel of its operating system. On the other hand, AIRFENCE has been designed with over 3 years of military testing with real world tactical scenarios which can automatically detect, locate, track and take over UAV controls all on full auto. From these claims, each piece or part of the whole of nature is always merely an approximation to the complete truth, or the complete truth so far as we know it. Therefore, things must be learned only to be unlearned again or more likely to be corrected.*

KEYWORDS: *Drones, Kernel, control, Operating systems, Vulnerability, Unmanned Aerial Vehicles, Hackable, Formal specification*

Date of Submission:02-10-2018 Date of acceptance: 13-10-2018

I. INTRODUCTION

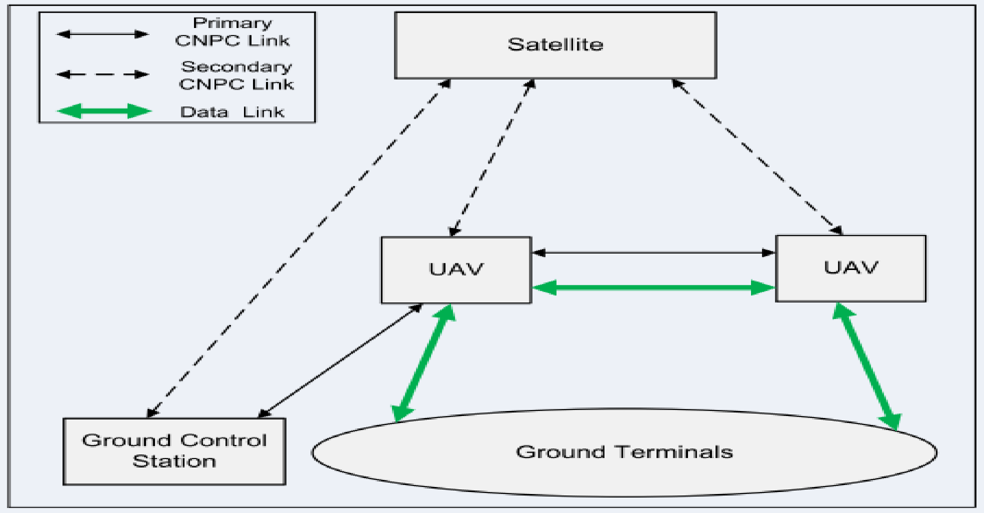
Machines across a range of industries are incorporating increasing intelligence and autonomy, whether in cars, airplanes, stock-trading algorithms, or household appliances [1]. Quite prevailing is the Unmanned Aerial Systems (UAS)- air vehicles equipped with sensors and software that enable the craft to fly without a human pilot on board. This is an emerging technology with a tremendous potential to revolutionize warfare and to enable new civilian applications. It is an integral part of future urban civil and military applications. UAS are considered in the context of a system which includes the air vehicle (Unmanned Aerial Vehicle), the control and communications systems, and the remote human operator(s). In addition, the payload of the UAS is the core element of the aircraft's mission.

Unmanned Aerial Vehicles (UAVs) are better suited for dull, dirty, or dangerous missions than manned aircraft [2]. UAS are mainly used for intelligence, surveillance and reconnaissance (ISR), border security, counterinsurgency, attack and strike, target identification and designation, communications relay, electronic attack, law enforcement and security applications, environmental monitoring and agriculture, remote sensing, aerial mapping and meteorology.

Whatever the UAV is designed for, there is every need to secure it from unauthorized users. Unauthorized users can gain access to either the stored contents of UAS, the processing capabilities (control) of a system, or intercept information being communicated between other systems. Each of these attacks requires a different set of skills and targets a different set of vulnerabilities [3].

Vulnerabilities exist in every system and it seems impossible to eliminate them all. Known vulnerabilities often exist as the result of needed capabilities. Unknown vulnerabilities, which the owner or operator of a system is not aware of, may be the result of poor engineering, or may arise from unintended consequences of some of the needed capabilities. Protecting yourself against attacks is a multistep process, which aims to limit and manage the vulnerabilities of your system [4].

The starkest truth is that wireless transmissions are inherently vulnerable to security breaches. Autonomous systems use wireless technology in their communications with the control terminals. With the broadcast nature of radio propagation, the wireless air interface is open and accessible to legitimate and illegitimate users. Considering the generic networking architecture of wireless communications with UAVs, there are basically two types of communication links, namely the Control and Non-Payload Communications links (CNPC) and the Data Link.



Basic Networking Architecture of UAV Wireless Communications: Source-

The CNPC links are essential to ensure the safe operations of all UASs. The CNPC main information flow can be categorized into three types: i. command and control from Ground Control Station (GCS) to UAVs; ii. aircraft status report from UAVs to ground; iii. sense and avoid information among UAVs. Due to these critical functions, CNPC links should operate in protected spectrum such as L-band (960-977MHz) and C-Band (5030-5091MHz). There is this salient high superior security need for CNPC links to avoid the so-called ghost control scenario, a potentially catastrophic situation in which the UAVs are controlled by unauthorized agents via spoofed control or navigation signals. Therefore, powerful authentication techniques, possibly complemented by the emerging physical layer techniques, should be applied for CNPC links.

The data links aim to support mission-related communications for the ground terminals. Compared to CNPC links, the data links usually have higher tolerance in terms of latency and security requirements [5].

The Claim

According to Heiser, the software that kept an unmanned Boeing Little Bird helicopter's computer secure was at the heart of its operating system, and it could be just what the world needs to make everything [6]. If hackers gain access to the kernel of any computer's operating system, then they can perform actions that are meant to be forbidden. In 2015, Heiser's team proved mathematically that their kernel, seL4, is unhackable. The kernel has a few highly secure properties: it can only do what it is designed to do; its code can't be changed without permission; and its memory and data transfers can't be read without permission. An earlier version of it, called OKL4, is now in millions of smart phones. Heiser says that two features underpin seL4's security, one of which is a new way of isolating data inside the kernel. But the key development was making the code capable of being checked mathematically. Other kernels might have these properties too, but it is impossible to know for sure without mathematical proof [6]. The achievement of the 'unhackable' system is a big deal for cyber-security, but there are always ways to attack hardware, even if the software is impregnable. Hackers might be able to spoof a device's sensors or jam incoming communications or other signals, which could be just as devastating [7].

The aspiration to create formally verified software has existed nearly as long as the field of computer science. In October 1973, Edsger Dijkstra came up with an idea for creating error-free code. While staying in a hotel at a conference, he found himself seized in the middle of the night by the idea of making programming more mathematical [8]. As he explained in a later reflection, "With my brain burning, I left my bed at 2:30 a.m. and wrote for more than an hour". That material served as the starting point for his seminal 1976 book, "A Discipline of Programming," which, together with work by Tony Hoare (who, like Dijkstra, received the Turing Award, computer science's highest honor), established a vision for incorporating proofs of correctness into how computer programs are written. For a long time it seemed hopelessly out of reach, but advances over the past decade in the so-called "formal methods" have inched the approach closer to mainstream practice. Today formal software verification is being explored in well-funded academic collaborations, the U.S. military and technology companies such as Microsoft and Amazon. A formal specification is a way of defining what exactly a computer program does. And a formal verification is a way of proving beyond a doubt that a program's code perfectly achieves that specification [9].

According to [10], about 64% of the 2500+ vulnerabilities in the National Vulnerability Database NVD were due to programming mistakes, and the majority of software vulnerabilities are caused by coding errors that can be introduced during the different phases of the software development lifecycle. The undetected errors can turn into security vulnerabilities at run-time, and can be exploited by intruders who may introduce serious damages to the software critical resources. The undetected flaws can cause cost increase to developers, who will spend more time on maintenance and flaws corrections. Using code analysis tools would avoid such issues and help produce safe and secure software.

The use of formal specification is currently mandated for high assurance security- and safety-critical systems [ITS91, MoD91a, MoD91b]. In these cases, the perceived expense of using formal specification can be justified as there is no other method of performing the project to the satisfaction of the regulatory authorities. Formal specification and verification were found to be practical, cost-effective tools for detecting potential security weaknesses, and helped to significantly strengthen the security of the access control system.

It is "axiomatic" that systems will never be made 100% secure. Formal methods will not break this axiom. Moreover, substitute the word "proven" for "made" in the previous sentence and we have a corollary. It would be foolish for anyone in formal methods to stand up and say "I can prove your system is 100% secure."
The motivation from the formal methods community is clear: security still remains a challenge. The motivation from the security community is strong too. More and more people place their trust in computing systems today for doing everything from casual shopping to medical services, and more and more systems are built out of commercial off-the-shelf components. It is no longer just the government, the military or the universities who are the purchasers, users or conveyors of large complex computing systems. Thus, systems designers and implementers are more willing to pay the price for increasing the assurance that their systems are secure. Formal methods can provide such increased assurance [11]. There are some areas of computer science such as security where exhaustive testing is not possible because security properties are very hard to prove before implementation. Therefore, formal methods can be used to validate security properties before implementation. Theorem provers are normally used to generate formal proofs of software systems while model checkers are used to validate models and remove faults such as deadlocks. The use of formal method specification languages, model checkers and theorem provers depends on the nature of the application being developed [10].

The Disclaim

Focusing on the security of wireless communications, the broadcast nature of radio propagation in the wireless air interface is open and accessible to legitimate and illegitimate users [12]. Broadly speaking, in wireless communications security, the security of the data, the identity of the sender, and the identity of the receiver is put into consideration. The open communications environment makes wireless transmission more vulnerable to malicious attacks than wired communications, including both the passive eavesdropping for data interception, the active jamming for disrupting legitimate transmission, and hijacking of controls. The starkest truth is that wireless transmissions are inherently vulnerable to security breaches. More worrying is the fact that these security issues span through all the processes of the wireless communications: At signal source (transmission point), on transmission in the air (channel or medium), and at the destination point (receiving point). Although there is quite a number of security measures designed to curtail attacks at various points in the communications processes, there are prevailing lapses inherent in almost all these measures [12]. But as wireless systems become more and more ever-present in communications systems, there is every need to constantly seek better and more efficient ways to improve the security of wireless communications. More so, researches on robust and efficient wireless transmission technologies such as the heterogeneous network (HetNet), massive multiple-input multiple-output (MIMO), and millimeter wave (mm Wave) to handle enormous amount of sensitive and confidential information, indicate the need for an unrivalled wireless security service in the design and implementation of wireless communications standardization.

Most information security problems map directly to the logical constructs presented in the OSI Seven Layer Network Model, and so this idea demonstrates the usefulness of the Seven Layer Model in evaluating information security problems and solutions [13]. In the Open System Interconnection (OSI) model, some layers have more impact than others when securing information, but together, they can be used to build a comprehensive solution [14].

To buttress this argument, there are quite a number of devices and associated software that claim to hack into any Unmanned Aerial Vehicle. Most striking amongst these claims is the AIRFENCE. AIRFENCE has been designed with over 3 years of military testing with real world tactical scenarios. At its core, it can automatically detect, locate, track and take over UAV controls all on full auto. In addition, AIRFENCE can locate the operator with pin point accuracy in real time. AIRFENCE is a technology that will detect and disarm consumer UAVs when operated in restricted airspace [15].

According to Russ, the operation principle of AIRFENCE is as follows: UAVs emit electromagnetic waves (radio frequency signals) and have specific RF “finger prints” [15]. These waves generate current in the antenna of AIRFENCE using SDR’s (Software Defined Radios), the current is measured and interpreted with an internal computer, which instantly demodulates locally in the sensor and defines the RF signal as either a characteristic of a UAV based on AIRFENCE proprietary library, cell phone, Wi-Fi router, or unknown. Depending on customer’s specifications, alerts are raised via visual and audio signal from AIRFENCE UI (User Interface) and also by SMS/Email.

II. SUMMARY AND IMPLICATION

It is plausible to develop “unhackable” software in Unmanned Aerial Vehicles (drones) to be used by responsible and humane people; otherwise it could easily be used by terrorists for attacking unsuspecting people and properties. In essence, “Unhackable” drone could be worse than nuclear weapons. Formal methods are rarely used today and are often rejected out-of-hand as being too difficult or expensive. Our experience has convinced us that, at least for small projects, or for small portions of large systems, formal methods are a practical and cost-effective adjunct to traditional software engineering methods.

From history of human knowledge and development, humans do not yet know all the basic laws governing nature, and so there is always an expanding frontier of ignorance. Each piece or part of the whole of nature is always merely an approximation to the complete truth, or the complete truth so far as we know it. Therefore, things must be learned only to be unlearned again or more likely to be corrected [15]. As such, the claim of “unhackable” software may be accepted just for the time being until it is unlearned or proved otherwise. And like the story of the Gazelle in Africa, behind the field of battle, someone has developed a strategy.

REFERENCES

[1]. Andrew, P. W. & Paul, D. S. (2015). Autonomous Systems: Issues for DefensePolicymakers.

[2]. Suraj, G. G., Mangesh, M. G. & Jawandhiya, P. M. (2013). Review of UnmannedAircraft System (UAS) . International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)- Volume 2, Issue 4, April 2013. www.ijarcet.org

[3]. Husrev, T. S., Mahalingam, R. & Ali, N. A. (2004). Data Hiding Fundamentals and Applications: Content Security in Digital Media. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.183.1122&rep=rep1&type=pdf

[4]. Ryan, J.C.H. Julie (2017). How do computer hackers "get inside" a computer? Scientific America. https://www.scientificamerican.com/article/how-do-computer-hackers-g/)

[5]. Yong, Z., Rui Z., & Teng, J. L. (2016). Wireless Communications with Unmanned Aerial Vehicles: Opportunities and Challenges. https://arxiv.org/pdf/1602.03602.pdf

[6]. Michael, Slezak (2015). Unhackable kernel could keep all computers safe from cyber- attacks. NewScientist. https://www.newscientist.com/article/mg22730392-600-unhackable-kernel-could-keep-all-computers-safe-from-cyberattack-2/

[7]. Shames, Iman (2015). Unhackable kernel could keep all computers safe from cyber-attacks. University of Melbourne in Australia. 16 September 2015 https://electrical.eng.unimelb.edu.au/people/iman-shames

[8]. Mike, James (2013). The Poetry of Programming. I-Programmer. https://www.i-programmer.info/history/people/144-dijkstra.html

[9]. Kevin, Hartnett (2016). Hacker-Proof Code Confirmed. Quanta Magazine. https://www.quantamagazine.org/formal-verification-creates-hacker-proof-code-20160920/).

[10]. Zhioua, Z., Short, S., & Roudier, Y. (2014). Towards the verification and validation of software security properties using static code analysis. International Journal of Computer Science: Theory and Application

[11]. Zeineb, Yves, R. Z. & Rabea B. A.(2017). Formal Specification and Verification of Security Guidelines: 2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing.

[12]. Yulong, Z., Jia, Z., Xianbin, W., & Lajos, H. (2016). A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends. Creative Commons https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7467419

[13]. Damon, Reed (2003). Applying the OSI Seven Layer Network Model To Information Security: SANS Institute 2003, As part of the Information Security Reading Room. https://www.sans.org/reading-room/whitepapers/protocols/applying-osi-layer-network-model-information-security-1309

[14]. Kari, A. Pace (2004). A Layered Security Model: OSI and Information Security. Global Information Assurance Certification Paper: SANS Institutehttps://www.giac.org/paper/gsec/3908/layered-security-model-osi-information-security/106272

[15]. Russ Curry (2016). AIRFENCE, A New Counter-Drone Defense Capability. https://www.uasvision.com/2016/10/26/airfence-a-new-counter-drone-defense-capability/

[16]. Feynman, R. P., Leighton, R. B. & Sands, M. (1963). Lectures on Physics: Vol.I: Addison-Wesley Publishing Company- London.

Caleb, Nanchen Nimyel "Security Of Unmanned Aerial Vehicles: A Literature Review And Research Directions "American Journal of Engineering Research (AJER), vol. 7, no. 10, 2018, pp. 106-109