JDMS

Journal of Defense Modeling and
Simulation: Applications,
Methodology, Technology
2016, Vol. 13(3) 331–342
© The Author(s) 2015
DOI: 10.1177/1548512915617252
dms.sagepub.com
SAGE

Security of unmanned aerial vehicle systems against cyber-physical attacks

Chaitanya Rani[1], Hamidreza Modares[1], Raghavendra Sriram[1], Dariusz Mikulski[2] and Frank L Lewis[3]

Abstract

The federal aviation administration has estimated that by the year 2020, the United States will have over 30,000 drones. Nowadays, drones, also known as unmanned aerial vehicles (UAVs), are ubiquitous and have numerous uses beyond military applications. This is because UAVs can be used in hazardous missions, since they exclude the risk factors involved in manned vehicles. Despite their benefits, UAVs are prone to attacks as they are equipped with numerous on-board sensors to gather data and this exposes them to various vulnerabilities. More precisely, in the absence of manual control, an attacker can gain access to sensitive sensory data and feed fraudulent information to the UAV. As a result, it can be reprogrammed to an undesirable effect and this can cause irreversible damage. This paper provides a general overview of current hacking methods, and defense and trust strategies to overcome cyber attacks on UAVs. To further highlight the importance of the requirement of developing new methods to avoid any intrusion, a hacking procedure is implemented on a commercially available UAV and its severe results are demonstrated. It is shown that the hacker can make irreparable damage and take complete control over the UAV by compromising the communication link between the operator and UAV and uses Robot Operating System-based tools to alter the flight path.

Keywords

Unmanned aerial vehicle, security, cyber-physical attacks, trust, hacking

1. Introduction

Unmanned aerial vehicles (UAVs) have gained popularity because of their ability to be remotely controlled or programmed, and their high efficiency and effectiveness due to low-cost and light weight specifications. They are becoming pervasive and have been employed in a variety of military and civil applications, including monitoring sites for gas leakage, forest fires, pipelines in factories, underground mines and caves, weather monitoring, military training, disaster relief,[1,2] and search and rescue,[3] to name a few. UAVs also have business applications. For instance, Amazon[4] announced the utilization of UAV technology to deliver packages to customers in less than 30 minutes.

Despite various advantages and applications of UAVs, they are vulnerable to attacks, as there is no pilot to monitor the activities in some situations. This results in the emergence of new challenges in the design of secure and reliable UAVs to avoid any damage to themselves and humans around them. Attacks, such as distributed denial of service (DDoS) and Trojan viruses, provide an illegal entry into the system. Once a system is attacked, it is very difficult to bring it back online. Existing information security methods, such as encryption or intrusion detection, are insufficient to protect a system from such risks. In fact, these methods do not take into account the compatibility of the sensor and actuator measurements with the UAV

[1]Department of Electrical Engineering, University of Texas at Arlington, USA
[2]Ground Vehicle Robotics (GVR), U.S. Army (RDECOM/ TARDEC), USA
[3]University of Texas at Arlington Research Institute, USA

Corresponding author:
Chaitanya Rani, Department of Electrical Engineering, University of Texas at Arlington,701 S Nedderman Dr, Arlington, TX 76019, USA.
Email: chaitanya.veerannagowda@mavs.uta.edu

physical process and control mechanism, which are of paramount importance in the protection scheme.

Most UAVs use communication over a network to transmit information to and from the ground station. Wi-Fi, GPS, Bluetooth, Infrared, and ZigBee are some forms of communication that are commonly used and are easily susceptible to attacks. Several attacks on UAVs have been reported. For instance, while filming of an Australian triathlon with an UAV, the operator lost complete control over the vehicle, causing it to crash into one of the athletes.[5] It was reported that an attacker had manipulated the communication link of the drone's control by a ''channel hop'' attack. Simple Wi-Fi and GPS jammers can be used to perform such attacks. GPS spoofing[6] is another threat that fools the receiver to track fraudulent signals by providing falsified information about the location. As another example, Sensepoint, a London-based security firm, has developed a software running on an UAV that accesses smartphones connected to open Wi-Fi networks in airports, cafes, and train stations. It extracts user information such as usernames, passwords, and credit card details.[7]

In the forthcoming years, UAVs will populate the sky and the most important necessity is to avoid incidents as a result of cyber attacks. The level of attacks will continue to rise and manufacturers, as well as governments, should make a collective attempt to safeguard these systems against attacks. Moreover, regulations on transportation should be implemented to impart safety rules and training should be provided to users of UAVs. Security officers should be deployed to uphold the safety of these devices and of civilians using them. Better regulation and restrictions on usage should be enforced by UAV manufacturers on their products. Confidentiality of personal data should be maintained and privacy should be respected to avoid intrusions.

In this paper, the vulnerabilities of UAVs to attacks are explored. To this end, a general overview of current hacking methods is given. Existing defense, prevention, and trust strategies for avoiding possible damages caused by attacks are also reviewed. The importance of the requirement of developing new defense methods is highlighted by implementing a hacking procedure on a commercially available UAV, that is, the Parrot AR.Drone, and its severe results are demonstrated. It is shown that the hacker can make irreparable damage and take complete control over the UAV by compromising the communication link between the operator and UAV and uses Robot Operating System (ROS)–based tools to alter the flight path. This helps us to identify threats and highlight the importance of securing these systems against attacks.

The remainder of this paper is organized as follows. Section 2 introduces various hacking strategies that can be used to gain control of UAVs. Section 3 presents numerous strategies to protect a system against an attack due to hacking. Section 4 outlines the use of control system and graph theory strategies to implement trust-based control in unmanned systems. Section 5 demonstrates an experiment conducted using the Parrot AR.Drone and a Wi-Fi cracking software tool to gain access over the system completely.

2. Hacking methods

Hacking refers to gaining illegitimate entry into another system or network. Wireless attacks are the most common form of hacking. In this section, we review various hacking strategies to compromise a system's ability to be controlled by its rightful owner. Some of the available hacking techniques are as follows:

a) Password theft;
b) Wireshark;
c) Man-In-The-Middle (MITM) attacks;
d) Trojan horse virus;
e) Distributed denial of service (DDoS) attacks.

2.1. Password theft

Passwords are usually a sequence of 8–16 character combinations formed using the keyboard. Unique passwords have to be created by blending upper and lower case alphabets, numbers, and special characters to ensure safety. This leads to complex passwords that are not easy to recall. Unfortunately, even complex passwords can be cracked using software tools such as dictionary attacks, brute force attacks, and statistical methods such as Aircrack-ng. Dictionary attacks scan for open ports and use a list of common words from the dictionary to perform the attack.[8] Some programs use a mixture of commonly found numbers and symbols to guess the password. Brute force attacks are efficient for cracking passwords that are shorter in length. The system will check for all possible permutations and combinations until the password is cracked. The use of Aircrack-ng is elaborated in Section 5 to gain access to an UAV platform such as the Parrot AR.Drone.

2.2. Wireshark

Wireshark is a robust tool for analyzing and capturing packets for wireless networks. By analyzing the individual transmission control protocol (TCP) or user datagram protocol (UDP) data packets, it is possible to extract all the necessary information required to launch an attack on any device on the network. This allows us to effortlessly access a client system and gain control over it. Once Wireshark is put to promiscuous mode, a list of available interfaces is displayed.[9] Wireshark is useful when websites use HTTP connections. Traffic of voice over internet (VoIP) calls

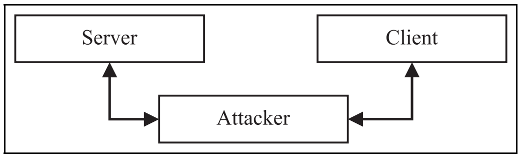


Figure 1. The Man-In-The-Middle attack: the attacker forms a fake connection between the server and the client.

and raw USB can also be captured. It fails when HTTPS protocol is used.

2.3. Man-In-The-Middle attacks

MITM is a type of attack where the attacker gains control of sensitive data by furtively modifying the communication link between two parties. The end users are usually unaware of the manipulation performed by the attacker. Figure 1 further illustrates the MITM attack.

A simple example of the MITM attack is the scam email that is disguised as a genuine email that misguides the user to a fake site. The user is then tricked to login while the attacker eavesdrops and collects credential information such as passwords, user name, and credit card numbers.[10] Other forms of MITM attacks are URL manipulation, rogue domain name server (DNS) and address resolution protocol (ARP) poisoning, and duplication of media access control (MAC).[11]

2.4. Trojan horse virus

The Trojan horse virus is a malicious program or software that monitors traffic over a network, and causes detrimental effects by destroying files and damaging hard drives in a system. Such viruses usually take leverage of a security glitch and multiply quickly. It also gives the attacker remote access to the system. This is known as a backdoor Trojan.[12] Unwanted email attachments and downloads could fool the user to open the link or install the software. A destructive Trojan can continuously delete all files and ultimately demolish the Operating System. These viruses cannot be easily identified by any anti-virus programs.

The best way to prevent Trojan attacks would be to minimize risk at the root cause. Unsolicited emails and programs should not be entertained and a good quality anti-malware should be installed in the system before being affected.

2.5. Distributed denial of service

A DDoS attack is a large-scale intrusion method performed by a host source that causes detrimental effects to legitimate users by withholding services.[13] This either causes the system to shut off completely or drains the system resources, such as computing power and bandwidth. Exposed or defenseless computers are first targeted by virulent attackers. They generally have expired or no anti-virus software installed.

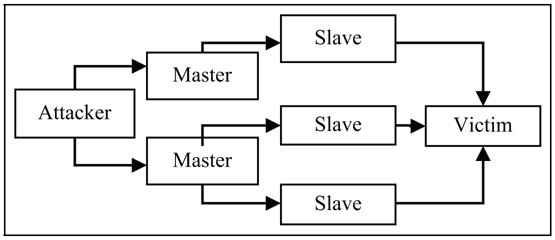


Figure 2. A conventional distributed denial of service attack.

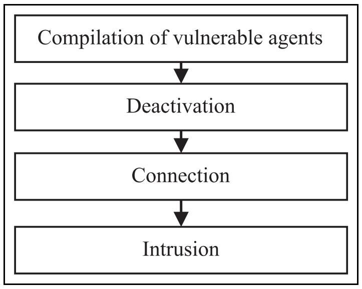


Figure 3. Steps in initiating a distributed denial of service attack.

Once the attacker gains access to the system, new tools can be used to gain control of the host. Infected systems continue to look for other vulnerable systems and attack them. This results in a master–slave process where the affected victim is being controlled by the attacker. Figure 2 illustrates the process of a DDoS attack. The attacker is now in possession of control of a large number of systems. Resources of the system are exhausted rapidly and flooding of packets occurs on the victim end. The attacker then removes all traces that could lead back to the source of the attack by using a spoofed internet protocol (IP) address, thereby preventing the victim permeating any illegal traffic targeted towards them. There are various DDoS attack techniques, such as random scanning, local subnet scanning, hit-list scanning, permutation scanning, and topological scanning.[13] Figure 3 provides an insight to the steps involved in a DDoS attack.

Steps in initiating a DDoS attack are shown in Figure 3 and are detailed as follows.[14]

2.5.1 Compilation of vulnerable agents.

The network is first scanned for vulnerable agents for the attacker to compile a list of agents to attack. Earlier, manual attempts were made to attack a system. In recent years, due to the advancement of technology, it is possible to attack systems by setting up automated software to scan the network and take over vulnerable agents.

2.5.2 Deactivation. Flaws in security and agent vulnerability are misused by the attacker. Software codes are used to automatically attack and disband the rightful owner from controlling the system. Suitable actions are taken by the attacker to safeguard the code planted in the system from being removed.

2.5.3 Connection. Protocols such as TCP or UDP are used to connect with numerous agents and plan attacks accordingly via scheduling. Attacks can be performed on either single or multiple agents.

2.5.4 Intrusion. The features of the victim, such as port numbers and time to live (TTL), are conformed to the attacker's needs. The attacker then launches the attack and alters the properties of the system. This is in favor of the attacker, since alteration of the packets would lead to complications in identification of the source of attack.

3. Defense against hacking

There are numerous methods to secure the UAV agents by hampering threats at the root or identifying them and taking suitable actions when encountered. However, no system is completely secure due to inherent flaws in the operating system that can lead to vulnerability. Most common attacks are usually due to external agents. This section provides an overview of some existing solutions for enhancing the security of the UAVs. Some of the relevant solutions are discussed as follows:

a) Encryption;
b) Defense against DDoS;
c) Intrusion detection systems (IDSs).

3.1 Encryption

Encryption is a simple method of securing sensitive data on systems. Encryption deals with encoding of data or messages that allows access only to legitimate users. It acts as a barrier against unwanted activities by denying access to secure resources such as credit or debit card data, personal details, social security numbers, and so on.[15] This encourages governments and other organizations to promote secure transmission of sensitive information. Figure 4 illustrates the process of encryption.

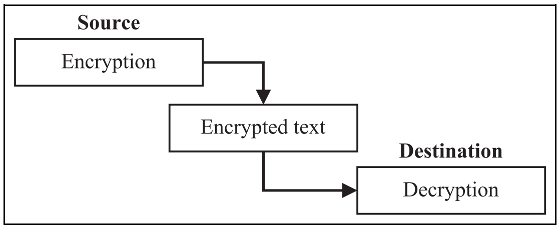


Figure 4. Process of encryption.

Civilians can use encryption keys to guard confidential data regarding personal records and protect themselves against fraudulent activities. Encryption can be utilized to insulate data being transferred over a network such as Wi-Fi, Bluetooth, intercom systems, mobile phones, and automatic teller machines (ATMs) in banks.

Although brute force attacks can beat the encryption, one can increase the key length to strengthen the encryption. Another useful approach to prevent hacking is to hide access points (APs) by disabling the service set identifier (SSID) of the system and by granting access to systems with familiar MAC addresses.[16] Sometimes spoofed MAC addresses also enable hackers to join the network. For the case of the UAV systems, a password can be used to help authenticate messages being communicated between the UAV and the operator.

3.2 Defense against DDoS attacks

There are several mechanisms to prevent DDoS, such as reactive and preventive mechanisms. Reactive methods use a mechanism to identify the attack at its source to prevent the damage. They are also called "Early Warning Systems".[13] Preventive methods minimize the feasibility of the attack by taking remedial measures. Routine scanning by virus scanners, firewalls, patches, anti-malwares, and maintaining appropriate protocols with the help of sensors and filtering mechanisms help to keep a check on abnormal activities. They maintain a log of normal behavior of the system and constantly check for unusual or peculiar behaviors. A threshold is set to decide the amount of abnormality.

Signature-based detection is also used to prevent DDoS attacks.[17] A database of well-known attacks is used to compare and identify incoming attacks. The disadvantage of this method is that it cannot identify any new attacks on the system. Attempts to safeguard a system before being attacked are extremely important as they decrease the chances of a potential DDoS attack.

A hybrid system uses the combination of preventive and reactive methods as a countermeasure for DDoS attacks.[18] Reactive mechanisms can counter the behavior of an attack once it is detected.[13] A DDoS attack exhausts the resources of the victim within a few seconds. Implementing a faster technique to protect against DDoS attacks would require a larger processing source. The consequence of having a larger processing source may affect the efficiency of the identification process.

3.2.1. Generic architecture of DDoS attack defense mechanisms. To elaborate on the above defense mechanisms, three classes of DDoS defense mechanisms can be generalized as given in Bhuyan et al.:[19] source-end, victim-end, and intermediate defense.

- Mechanism of source-end defense

Rate limiters are used to compare the traffic against a set of predefined profiles. Preventing an attack at the source-end is the most effective way as it avoids flooding the network. Identifying attacks at the source-end is usually a difficult task, since malicious traffic may pretend to behave similar to normal traffic and this poses a challenge to the classification process.

- Mechanism of victim-end defense

Intrusion detection schemes are employed to detect malicious traffic. Signatures of abnormal activities are noted in a list and their behavior is constantly compared with other incoming or outgoing traffic. A threshold is set to determine if any misleading behavior is identified. The drawback of this mechanism is that it is too late for the attack to be prevented. This denies legitimate users the resources of the system.

- Mechanism of intermediate defense

Intermediate defense combines the efforts of both source-end and victim-end defenses: that is, identifying the attack at source and using routers to deploy this information to prevent attack at the victim-end. Failure of routers or the detection mechanism at the source-end could have undesirable effects on the victim, since the detection mechanism completely relies on the source for initial defense.

An effective method for defense against DDoS is defensive cooperative overlay mesh (DefCOM). DefCOM is a conglomerate of nodes that are formulated in a network where each system can convey information directly to other systems without having to use a central system for communication. Such a network is known as peer-to-peer.

Figure 5[20] illustrates how DefCOM framework works. This framework typically consists of three nodes, namely, the alert generator, rate limiter, and classifier.

- Alert generator

An alert generator monitors the network to find probable virulent traffic and notifies all the other nodes. It must be able to detect when an attack can be expected. Firewalls and IDSs are good examples of alert generators. They are usually installed at the victim-end to detect any abnormal activity with high precision.

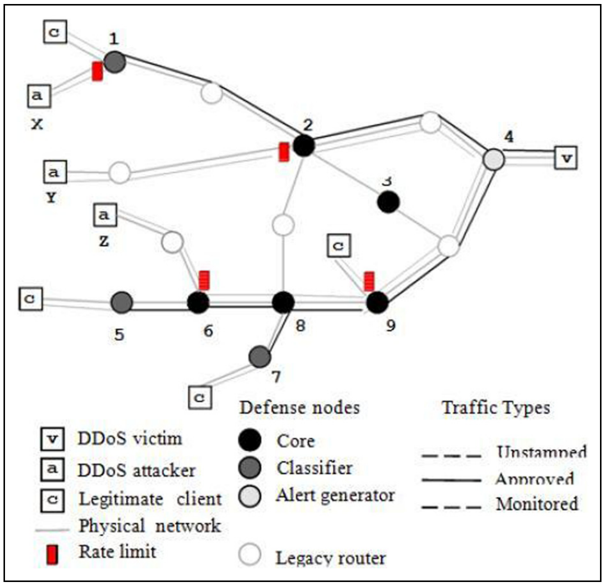


Figure 5. Illustration of the defensive cooperative overlay mesh framework. DDoS: distributed denial of service.

- Rate limiter

Rate limiters are installed at the edge of the network to allow only selective traffic by comparing it against the list of legitimate traffic on the network. It prioritizes the flow of traffic to valid users and restricts the flow to suspicious nodes. Routers are used to implement rate limiters. They are beneficial for cases in which classifiers cannot be installed.

- Classifier

Classifiers are present at the core of the network to identify authentic traffic from virulent ones and uses rate limiters to restrict the flow of traffic to suspicious users. The information from alert generators is utilized to secure the system from any malicious activity and minimize potential damage to the system.

The DefCOM identifies the attack on the source-end by using classifiers, and rate limiters are used at the victim end to strain malicious traffic and the system is alarmed using alert generators.

3.2.2 DDoS defense using soft computing methods. DDoS attacks can also be detected using soft computing methods, as demonstrated by Mirkovic et al.[20] By using artificial neural networks, fuzzy systems, genetic algorithms, and so on, the analysis of DDoS attacks can be performed and classified automatically. Soft computing is an efficient way to process the uncertainty in the system.

Neural networks are inspired from biological neural systems and have been widely used, as a machine learning tool, in various applications.[21] In attack detection application, they can be used to evaluate and categorize the incoming or outgoing traffic as malicious or normal traffic. For instance, radial basis functions (RBFs) can be used to perform feature classification.[22] Classifiers at the edge of the victim-end are employed to trigger RBF networks. They use a filtering technique to send malicious traffic to alert generators and normal traffic is sent to its destination.

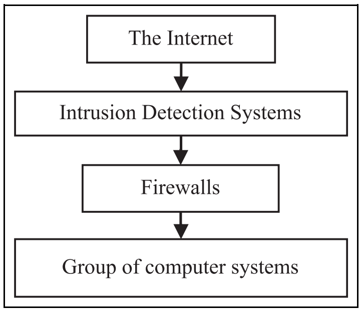


Figure 6. Intrusion detection system used as a defense mechanism.

3.3 Intrusion detection systems

Intrusion is the act of wrongful entry or compromise of a system's resources without consent of the rightful owner. Systems can be tricked, often due to glitches, into allowing access to an intruder. Intrusion detection is the act of supervising and identifying indications of abnormal activity.

The IDS is usually a software that inspects the system behavior to find anomalies. IDSs are defensive tools but do not provide preventive actions to safeguard a system. There are two kinds of IDSs: the Misuse IDS and the Anomaly IDS.

The Misuse IDS[23] deals with pre-defined attack signatures that take advantage of security loopholes. Such signatures are well-known prior to the attack and are used to test against incoming patterns for its virulent nature.

The Anomaly IDS[23] uses a system's regular performance to check its behavior and form statistics accordingly. If there is a deviation from the normal usage performance, then such a behavior is noted by the mechanism in an IDS.

An IDS identifies the incoming traffic and determines whether it has to be protected or not. It utilizes three types of information: long-term, configuration, and audit information.

Long-term information deals with building a database of techniques used in identifying attacks.[21] Configuration information relies on the present condition of the system. Audit information portrays the situation and circumstances of the system. Unwanted data from the audit and an artificial map of the state of the system is provided. Based on this, a conclusion is reached on whether the identified symptoms indicate an actual intrusion or not. Preventive measures are taken to bring the system back to a safe state.

By the use of firewalls, encryption, and authentication, the first barrier to secure a system is formed. As discussed earlier, passwords with shorter lengths or weak passwords can be easily attacked. Firewalls are prone to errors while setting them up and may not guarantee safety of the systems. Hence, there is a need for an IDS to safeguard a system from attackers. IDSs provide additional security along with the first barrier of protection with the help of encryption, authentication, and firewalls. Figure 6 illustrates the use of an IDS in a network. IDSs also provide safety measures to prevent attacks on a timely basis.

3.3.1 Neural networks in intrusion detection systems. Using classification functions, neural networks are able to perform intrusion detection. Neural networks identify the pattern or signature of the input and place it into classes accordingly at the output.[24] Neural networks used in IDSs are categorized as follows: (a) multi-layer feedforward neural networks that consist of a hidden layer in between the input and output layers; (b) Kohonen's self-organizing maps that form a mapping from input to the clusters. The detection process is expedited with the use of a neural network, since they have the ability to learn the features of past intrusions.

3.3.2. Fuzzy rule-based systems in IDSs. Fuzzy logic is a method that uses approximation techniques instead of fixed values. In binary systems, either 0 or 1 is used, but with the help of fuzzy logic we can derive ranges that lie between 0 and 1. This provides partial or intermediate values and hence fuzzy databases can be formed.[25] These systems are flexible and accommodate approximate values. Any valid set of data given based on input and output values can be translated to form a fuzzy system.

Fuzzy logic allows room for errors or uncertainty and provides a low-cost solution in a system. It provides significant advantages over other soft computing methods in intrusion detection. It detects abnormal activity in the system and formalizes strategies to resolve it accordingly. In Shanmugavadivu and Nagarajan,[26] the fuzzy system used to detect intrusions has the following steps: (a) fuzzy rule strategy creation; (b) decision module using a fuzzy rule; (c) classifying the input accordingly. The results of this provided more accuracy in the identification of anomalies based on the rules used in fuzzy logic.

By utilizing approximate data, if–then rules can be formed with the aid of an expert knowledge. The drawback in this method is that the number of if–then rules grows rapidly as the data set becomes larger. In neural networks, training is required, but in fuzzy based system rules are formed to reach conclusions pertaining to a data set.

4. Trust in system security

This section deals with different ways to secure a system based on graph theory and control system strategies. Trust propagation in networks is discussed to provide an insight on consensus algorithms. The concept of trust in systems is used as a measure to gauge the reliability of the system. The information we obtain from another agent may not always be true. Sometimes, malicious agents in a network send out unwanted information to particular nodes or agents and corrupt the system information. This can be avoided by defining certain parameters to gauge the information that we obtain from each agent. Evaluation of data constantly provides us with trust values. These are also defined as reputation or confidence of the system.

Usually, the trust parameters in a network are built over time by communicating with neighboring agents and keeping track of anomalies. Once the variations in agent behavior are noted, the techniques in Section 3 can be coupled with the trust-based method to make a system secure.

The framework of agents placed in the graph strongly influences the behavior of the agents and flow of communication between the nodes in the graph. Data from neighboring nodes is of great significance. Team-based trust propagation methods in sensor networks and UAVs have been developed. In the literature a few strategies have been used to secure a network.

In Jiang and Baras,[27] to eliminate malicious agents, the method of negative trust values has been employed. When a node behaves erratically, the trust value of that particular node is updated adversely. Securing the system against malignant nodes is also studied by Theodorakopoulos and Baras[28] using the game theory.

The concept of cooperative control on UAV platforms is discussed by Geramifard et al.,[29] by the implementation of a planner and learner that follows a safe policy initially and further moves on to avert risky behavior by employing stochastic risk models. Such systems help raise awareness in networks.

Active directory, a service on Windows networks developed by Microsoft, checks the authenticity of the user's passwords and names to be capable of enjoying administrator rights on a system. It uses a network of trust relationships to allow sharing of sensitive data between domains.[30] The default trust types are parent and child, tree–root.

Using the game theory, trust dynamics can be derived for cooperative control by Mikulski et al.[31] and Mikulski et al.[32] Various trust games are defined – additive, constant-sum, super-additive, and convex – to form trust models. This was applied to a military convoy where the vehicle in the lead can be trusted by all other vehicles following it.

In Haus et al.,[33] every agent has an individual trust vector established by a network information exchange. This is based on observations formed among local neighbors of an agent. The consolidation of local information exchange and the observations gives rise to trust-based formation control in a network. Consensus is reached by reflecting the influence of the neighboring agents.

Trust in networks such as sensors, wireless local area networks (LANs), mobile ad-hoc, UAV and unmanned ground vehicle (UGV) teams can be augmented by setting up protocols such as bilinear voting in Ballal et al.[34] It also discusses the methods of trust spreading in a network and the influence of local information on a particular agent. Behaviors in flocking are outlined to help analyze the rules of flocking and its application to trust-based systems.

For transmission of secure information over wireless networks using sensors, Zhang et al.[35] used the approach of Kulback–Liebler (KL) distance to recognize and eliminate nodes or agents that have turned rogue. The system uses MAC addresses for verification purposes, where each sensor has a key or password associated with it. Josang's belief model[36,37] is used to form opinions over the sensors during propagation of data over the wireless network. The opinions give the model a method to measure the trust in every sensor.

The RoboTrust model provides observations based on past history of the agents in Mikulski et al.[38] Acceptance functions determine the favorability of an agent by the previously established observations. The output of the acceptance function is either 1 or 0 for being favorable or unfavorable accordingly. Agents that cannot be observed directly employ the technique of indirect trust to determine the trustworthiness about other agents around them.

In Xu and Dudek,[39] the model of trust is tested on an UAV by using visual data as feedback and human intervention when a system turns rogue. This case is similar to a supervisor–worker case; where the performance of the UAV is closely monitored and when control is lost, a human supervisor takes over. The human uses external commands to control the system. Trust in the system builds over time but when the system goes rogue, the trust value drops rapidly.

Auction-based methods are used by Pippin and Christensen,[40] where each agent in the team announces the time it is inclined to allocate in performing a particular task. A list of bids is maintained and it is constantly updated with the submitted bids. The list updates the trust model adversely if the agent does not bid to perform a particular task. The trust model defines the bidding rules in the system. The agent that requires the least amount of time to perform a task is chosen as the winner. Experiments were carried out on UAVs to verify their participation in teams based on bids.

In Jiang and Baras,[41] the graph theory is involved to describe the behavior of the network. It is assumed that nodes in a network trust themselves fully. The range of trust lies between –1 (distrust) and 1 (trust). “0” is a neutral zone where the trust model has no opinion over the nodes. A confidence parameter is defined by the weight function that plays a key role in deciding the level of trust. A higher confidence correlates to a high trust value of a particular node. Headers are used and they act as previously trusted nodes whose opinions have a higher leverage. Rogue nodes are eliminated from the network.

The definition of cooperation in networks is defined by Arney and Peterson[42] as a combination of competitive and altruistic trust. Competitive trust deals with selfish behavior in a team. In the altruistic approach, the outcome is evaluated based on the perception of the team. Payoff functions were defined to describe the behavior of the team.

5. An application study: hacking the Parrot AR.Drone

This section shows and exposes certain vulnerabilities generally found in certain commercially available UAV systems, such as the Parrot AR.Drone. The hacking and control of the Parrot AR.Drone using tools such as Aircrack-ng and ROS are demonstrated. The Parrot AR.Drone is a quadcopter that has its own Wi-Fi network and allows users to connect and control it.[43] It functions on an 802.11 Wi-Fi protocol and couples with an authenticated client. An attacker can enable the packet capturing feature and intercept the flow of data and capture data packets that can be used to gain control over the UAV. Such devices are prone to being attacked. This aerial vehicle, among several others, can be controlled by various smart devices as well. The AR.Drone has frontal and base cameras to enable the user to access video streams of the drone’s view on their devices. It has gained popularity due to mass production of the product leading to low-cost availability. Hence, it is used to test loopholes and issues in security of UAVs.

Nowadays, the security protocols in Wi-Fi include wired equivalent privacy (WEP), Wi-Fi protected access (WPA), and Wi-Fi protected access II (WPA2). Here we use Aircrack-ng to de-authenticate a valid client and gain control over the system. Private or unseen SSID information can also be retrieved. All Wi-Fi signals have information about the SSID of the network and MAC address of the device.[44] An AP enables a wireless device to connect to a wireless network. Typically, the AP of the device connects to a router and has a range of 35–100 meters.

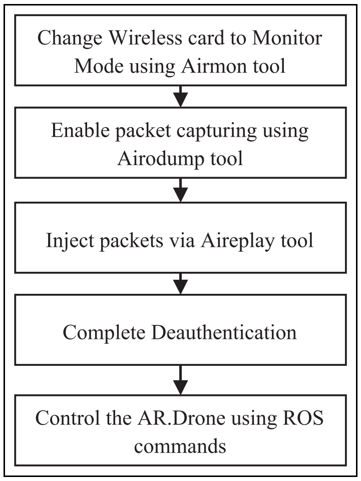

Change Wireless card to Monitor Mode using Airmon tool
Enable packet capturing using Airodump tool
Inject packets via Aireplay tool
Complete Deauthentication
Control the AR.Drone using ROS commands

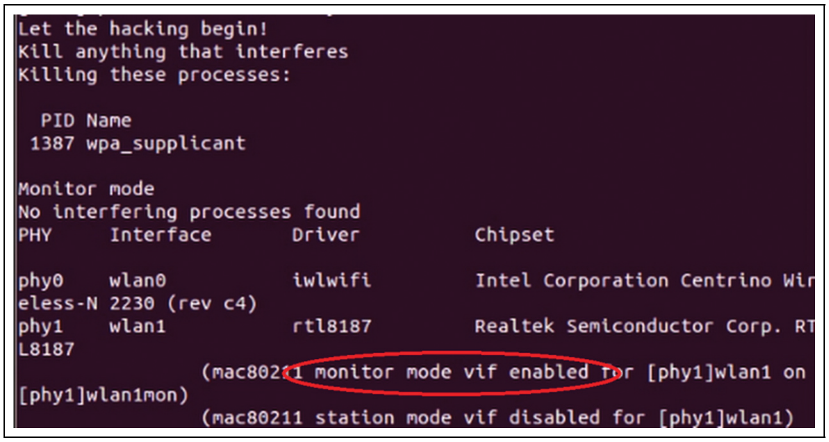
Figure 7. Steps in hacking the Parrot AR.Drone. ROS: Robot Operating System.

There are three frame types in wireless networks, namely, data packets, management packets, and control packets. An example of data packet is TCP or UDP packets.[45] Management packets usually deal with the devices associating with an AP. Control packets aid in reprogramming certain functions and tasks of the system.

The specific Wi-Fi addresses, the basic service set ID (BSSID) (which indicates the AP used), the source address (which identifies where the packets are coming from), and the destination address (which specifies the terminal point of the packets) are identified and used in establishing a backdoor entry into the network of the AR.Drone. On gaining access, we reconfigure the settings of the AR.Drone to disconnect it from its current owner and re-establish a connection with the attacker.[46]

A Unix shell script was created using Aircrack-ng, which is a cracking program that acts as a network sniffer and cracks the privacy component. Figure 7 explains the process.

Aircrack-ng is a WEP and Wi-Fi protected access preshared key (WPA-PSK) key-cracking program.[47] It is a collection of tools to crack passwords. Airmon-ng, Airodump-ng, and Aireplay-ng are tools available as part of Aircrack-ng software. As shown in the highlighted part of Figure 8, Airmon-ng puts our wireless card from managed mode to monitor mode. Necessary drivers for the chipset have to be installed for this to work. It enables the network card to view all the traffic. Airodump-ng allows



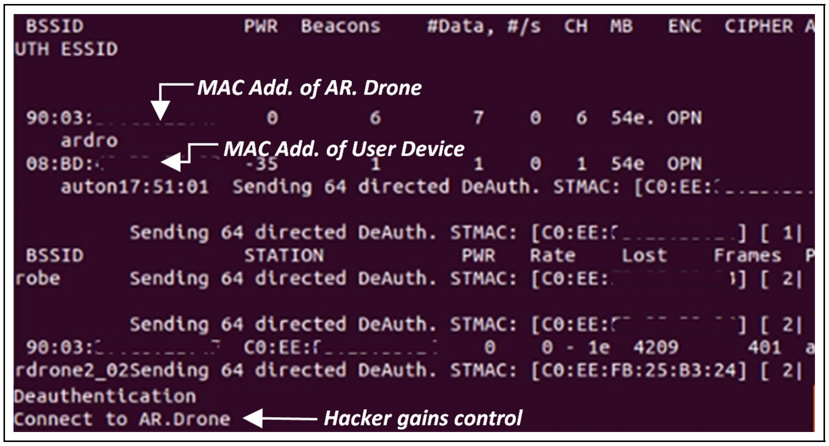
```
Let the hacking begin!
Kill anything that interferes
Killing these processes:

  PID Name
 1387 wpa_supplicant

Monitor mode
No interfering processes found
PHY     Interface       Driver          Chipset

phy0    wlan0           iwlwifi         Intel Corporation Centrino Wir
eless-N 2230 (rev c4)
phy1    wlan1           rtl8187         Realtek Semiconductor Corp. RT
L8187
                (mac80211 monitor mode vif enabled for [phy1]wlan1 on
[phy1]wlan1mon)
                (mac80211 station mode vif disabled for [phy1]wlan1)
```

Figure 8. Enabling monitor mode using Aircrack-ng.



```
 BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER A
UTH ESSID

 90:03:[redacted]     0        6        7    0   6  54e. OPN
    ardro
 08:BD:[redacted]    -35       1        1    0   1  54e  OPN
    auton17:51:01  Sending 64 directed DeAuth. STMAC: [C0:EE:[redacted]

          Sending 64 directed DeAuth. STMAC: [C0:EE:[redacted]] [ 1|
 BSSID              STATION            PWR   Rate    Lost    Frames  P
robe      Sending 64 directed DeAuth. STMAC: [C0:EE:[redacted]] [ 2|

          Sending 64 directed DeAuth. STMAC: [C0:EE:[redacted]] [ 2|
 90:03:[redacted]  C0:EE:[redacted]    0     0 - 1e  4209     401  a
rdrone2_02Sending 64 directed DeAuth. STMAC: [C0:EE:FB:25:B3:24] [ 2|
Deauthentication
Connect to AR.Drone
```

Figure 9. De-authentication process.

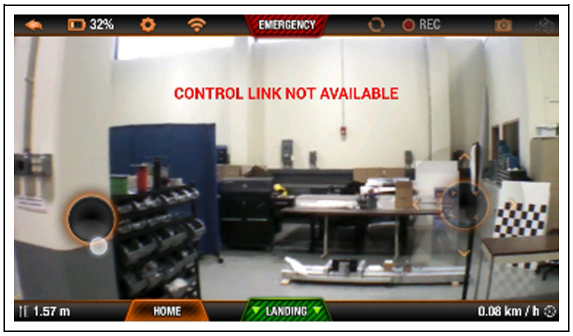
us to capture packets of a particular client out of the list of clients available.

Frames are injected using the Aireplay-ng tool. It has a wide variety of applications, such as de-authentication, fragmentation, and cafe-latte attack.[48] Using an Alfa Wi-Fi adapter based on the chipset, the airmon-ng tool command allows enabling of monitor mode as well as injects packets from any system. Airodump-ng is a tool by Aircrack to discern all the wireless networks in the scope of the wireless card. The APs, channel numbers, and BSSID of all the available stations are thus available to the hacker. The AP of the particular chosen network can be blocked by Aireplay-ng.

De-authentication packets are sent to allow the disconnection from the valid client. The extended service set identification (ESSID) of the AR.Drone is noted by the malicious client and it forcefully connects to the laptop after the de-authentication from the valid client has been successfully performed.

By using the procedure described in Figure 7, Figure 9 shows the results of the de-authentication process. Firstly, the tool identifies all the MAC addresses of the devices available on the network. Once identified, it gathers all the necessary data associated with that particular device and launches the attack. During the attack, the hacker injects data packets that reconfigure all the network parameters of the drone. As we can see in Figures 10 and 11, at the end of the process, the AR.Drone is de-authenticated and a malicious client now has access over it. The control of the drone is done using the ROS, an open-source platform that acts as a miniature operating system for a device.[49]

The drone can be controlled by publishing specific ROS-based commands over the Wi-Fi network. In this demonstration, the Parrot is controlled by the hacker joystick and various new tasks are performed and executed while the user is cut off from the system. In Figure 10, we see the resulting error of ‘‘Control Link not Available’’ being displayed on the legitimate user’s device screen,



CONTROL LINK NOT AVAILABLE

Figure 10. Valid client de-authenticated.

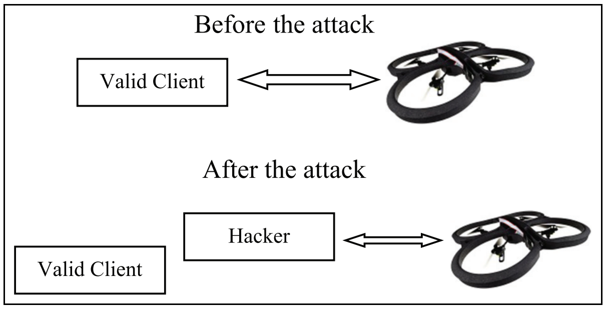


Figure 11. System architecture illustration before and after the attack.

when the user's network has been cut off and reconfigured to that of the hacker's specifications. The flight path is adjusted according to the malicious client. The video feed of the drone's view could also be obtained. Multiple demonstrations of the attack are implemented successfully. The duration of the entire process of the attack took less than a minute. This experiment confirms that UAVs are prone to attacks and this raises the need to secure them.

6. Conclusion

In summary, the hacking of the Parrot AR.Drone provides a clear example of how similar systems can be susceptible to de-authentication attacks. Exploitation of UAVs like the Parrot AR.Drone is a threat to civilians. The AR.Drone is inexpensive and has the capability to transport hazardous payloads. It can be effortlessly infiltrated by a third party. Since the drone can be de-authenticated by another user, it helps us explore the issues in security of the device. The control of the drone can be switched easily from a valid client to a malicious client.

The use of UAVs is expanding rapidly and its use in private and public sectors has seen a noticeable rise in demand. In this paper we have analyzed the importance of privacy linked to UAVs, so as to prevent exposure and exploitation of sensitive and personal information. Different attacks and defense strategies are outlined as a caution for industries manufacturing UAVs.

The goal of this study was to investigate and analyze the flaws in security in UAVs and also to prevent such attacks. By testing on small-scale UAVs, the results can be applied on larger systems. With this demonstration we understand how certain communication loopholes can be exploited and to stress the need of better security features and anti-hacking systems. The majority of UAVs today use Wi-Fi or radio frequency (RF)-based communication protocols. By studying the vulnerabilities of a Wi-Fi based device such as the Parrot AR.Drone, we expose certain drawbacks that can be misused.

The solutions to defending such attacks were articulated. Various techniques, such as neural networks, DefCOMs, and fuzzy logic based IDSs, were discussed. This helped gain exposure to vulnerabilities during cyber attacks.

Declaration of conflicting interest

The authors declare that there is no conflict of interest.

Funding

This work was supported by NSF grant ECCS-1405173, ONR grant N00014-13-1-0562, ONR grant N000141410718, AFOSR EOARD Grant 13-3055, and ARO grant W911NF-11-D-0001.

References

1. Liu S, et al. Towards aerial natural gas leak detection system based on TDLAS. In: *International symposium on optoelectronic technology and application*, 2014, pp.92990X-92990X.
2. McGonigle AJS, et al. Unmanned aerial vehicle measurements of volcanic carbon dioxide fluxes. *Geophys Res Lett* 2008; 35: 1–4.
3. Waharte S and Trigoni N. Supporting search and rescue operations with UAVs. In: *International conference on emerging security technologies (EST)*, 2010, pp.142–147.
4. Stolaroff JK. *The need for a life cycle assessment of drone-based commercial package delivery*. Livermore, CA: Lawrence Livermore National Laboratory, 2014.
5. Gallagher S.Triathlete injured by "hacked" camera drone", http://arstechnica.com/security/2014/04/triathlete-injured-by-hacked-camera-drone/ (7 April 2014, accessed 24 February 2015).
6. Hartmann K and Steup C. The vulnerability of UAVs to cyber attacks-An approach to the risk assessment. In: *IEEE international conference on cyber conflict (CyCon)*, 2013, pp.1–23.
7. Paganini P. Snoopy software can turn a drone is a data stealer, http://securityaffairs.co/wordpress/23374/

hacking/snoopy-drone-data-stealer.html (24 March 2014, accessed 6 March 2015).

8. Pinkas B and Sander T.Securing passwords against dictionary attacks. In: *Proceedings of the 9th ACM conference on computer and communications security*, 2002, pp.161–170.
9. Banerjee U, et al. Evaluation of the capabilities of WireShark as a tool for intrusion detection. *Int J Comput Appl* 2010; 6: 1–5.
10. Ornaghi A and Valleri M. Man in the middle attacks. In: *Blackhat conference Europe*, 2003.
11. Hatamleh H. A review and comparing of all hacking techniques and domain name system method. *Contemp Eng Sci* 2012; 5: 239–250.
12. Chunming ZMXQL. Analysis of Trojan horse and its detection. *Comput Eng Appl* 2003; 28: 1–4. .
13. Patrikakis C, Masikos M and Zouraraki O. Distributed denial of service attacks. *Internet Protoc J* 2004; 4: 13–35.
14. Nemati H (ed.) Information security and ethics: concepts, methodologies, tools, and applications: concepts, methodologies, tools, and applications. In: *IGI global*, 2007.
15. Rumale A and Chaudhari D. IEEE 802.11 x, and WEP, EAP, WPA/WPA2. *Int J Comput Technol Appl* 2011; 2: 1945–1950.
16. Lindqvist J, et al. Privacy-preserving 802.11 access-point discovery. In: *Proceedings of the second ACM conference on wireless network security*, 2005, pp.123–130.
17. Douligeris C and Mitrokotsa A. DDoS attacks and defense mechanisms: classification and state-of-the-art. *Comput Netw* 2004; 44: 643–666.
18. Hashmi MJ, et al. Classification of DDoS attacks and their defense techniques using intrusion prevention system. *Int J Comput Sci Commun Netw* 2012; 2: 607–614.
19. Bhuyan MH, et al. Detecting distributed denial of service attacks: methods, tools and future directions. *Comput J*. Epub ahead of print 2013. DOI: 10.1093/comjnl/bxt031.
20. Mirkovic J, et al. *Distributed defense against DDoS attacks*. University of Delaware CIS Department Technical Report CIS-TR-2005 2, 2005.
21. Abraham A and Jain R. *Soft computing models for network intrusion detection systems. Classification and clustering for knowledge discovery*. Berlin, Heidelberg: Springer, 2005, pp.191–207.
22. Gavrilis D and Dermatas E. Real-time detection of distributed denial-of-service attacks using RBF networks and statistical features. *Comput Netw* 2005; 48: 235–245.
23. Depren O, et al. An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. *Expert Syst Appl* 2005; 29: 713–722.
24. Zhang C, et al. Intrusion detection using hierarchical neural networks. *Pattern Recognit Lett* 2005; 26: 779–791.
25. Nauck D and Rudolf K. A neuro-fuzzy method to learn fuzzy classification rules from data. *Fuzzy Sets Syst* 1997; 89: 277–288.
26. Shanmugavadivu R and Nagarajan N. Network intrusion detection system using fuzzy logic. *Indian J Comput Sci Eng* 2011; 2: 101–111.
27. Jiang T and Baras JS. Trust evaluation in anarchy: a case study on autonomous networks. In: *IEEE international conference on computer communications (INFOCOM)*, 2006.
28. Theodorakopoulos G and Baras JS. Malicious users in unstructured networks. In: *IEEE international conference on computer communications (INFOCOM)*, 2007, pp.884, 891.
29. Geramifard A, et al. How UAV cooperative control with stochastic risk models. In: *American control conference (ACC)*, 2011, pp.3393–3398.
30. Dias J. *A guide to Microsoft active directory (ad) design*. Department of Energy Lawrence Livermore National Laboratory, 2002.
31. Mikulski DG, et al. Trust-based coalition formation in multiagent systems. *J Defens Model Simulat* 2014; 11: 19–32.
32. Mikulski DG, et al. Trust dynamics in multi-agent coalition formation. In: *Proceedings of SPIE, defense, sensors, and security*, 2011.
33. Haus TP, et al. Trust-based self-organising network control. *IET Control Theory Appl* 2014; 8: 2126–2135.
34. Ballal P, et al. *Trust-based collaborative control for teams in communication networks. Recent advances in nonlinear dynamics and synchronization*. Berlin, Heidelberg: Springer, 2009, pp.347–363.
35. Zhang W, Das SK and Liu Y. A trust based framework for secure data aggregation in wireless sensor networks. In: *Sensor and Ad Hoc Communications and networks (SECON'06)*, 2006, pp.60–69.
36. Jøsang A, Ismail R and Boyd C. A survey of trust and reputation systems for online service provision. *Decis Support Syst* 2007; 43(2): 618–644.
37. Josang A. Trust-based decision making for electronic transactions. In: *Proceedings of the fourth Nordic workshop on secure computer systems (NORDSEC'99)*, 1999.
38. Mikulski DG, et al. Trust method for multi-agent consensus. In: *Proceedings of SPIE, defense, sensors, and security*, vol. 8387, 2012.
39. Xu A and Dudek G. Trust-driven interactive visual navigation for autonomous robots. In: *IEEE international conference on robotics and automation (ICRA)*, 2012, pp.3922–3929.
40. Pippin CE and Christensen H. Cooperation based dynamic team formation in multi-agent auctions. In: *Proceedings of SPIE defense, security, and sensing*, 2012, pp.838919–838919.
41. Jiang T and Baras JS. Graph algebraic interpretation of trust establishment in autonomic networks. *Wiley J Netw*. Preprint 2009.
42. Arney DC and Peterson E. *Cooperation in social networks: communication, trust, and selflessness*. Army Research Office Research, Triangle Park, NC, 2008.
43. Krijnen D and Dekker C. AR Drone 2.0 with subsumption architecture. In: *Artificial intelligence research seminar*, 2014.
44. Sheng Y, et al. Detecting 802.11 MAC layer spoofing using received signal strength. In: *The 27th IEEE conference on computer communications (INFOCOM)*, 2008.
45. Bellardo J and Savage S. Denial-of-service attacks: real vulnerabilities and practical solutions. In: *USENIX security*, 2003, pp.15–28.

46. Akyazi U and Turkish Air Force. Possible scenarios and maneuvers for cyber operational area. In: *13th European conference on cyber warfare and security (ECCWS)*, 2014.
47. Mavridis IP, et al. Real-life paradigms of wireless network security attacks. In: *15th panhellenic conference on informatics (PCI)*, 2011.
48. Sheldon FT, et al. The insecurity of wireless networks. *IEEE Secur Privacy* 2012; 10: 54–61.
49. Quigley M, et al. ROS: an open-source Robot Operating System. In: *ICRA workshop on open source software*, 2009.

Author biographies

Chaitanya Rani is from Bangalore, India. She received her Bachelor in Engineering in Electronics and Communication Engineering from PESIT, Bangalore South, India, in 2013. She is currently working toward the Master degree in Electrical Engineering from the University of Texas at Arlington and works as a Graduate Research Assistant at the University of Texas at Arlington Research Institute. Her current research interests include Autonomous vehicle systems, control systems, distributed control systems, automation, embedded firmware, robotics, consumer electronics, and real time systems.

Hamidreza Modares received the BS degree from the University of Tehran, Tehran, Iran, in 2004, the MS degree from the Shahrood University of Technology, Shahrood, Iran, in 2006, and the PhD degree from The University of Texas at Arlington, Arlington, TX, USA, in 2015. He was a Senior Lecturer with the Shahrood University of Technology, from 2006 to 2009. His current research interests include cyber-physical systems, reinforcement learning, distributed control, robotics, and pattern recognition.

Raghavendra Sriram is from Mumbai, India. He received his Bachelor in Engineering in Electrical and Electronics Engineering from Canara Engineering College, India, in 2012. He is currently a PhD student in Electrical Engineering at the University of Texas at Arlington and works as a Graduate Research Assistant at the University of Texas at Arlington Research Institute. His current research interests include Control Systems and Robotics.

Dariusz Mikulski received the MS in computer science engineering in 2006, and the PhD in Electrical and Computer Science Engineering in 2013, both from Oakland University. He is a Cybersecurity Research Scientist at the US Army TARDEC - Ground Vehicle Robotics. His research interests include computational trust, vehicle cybersecurity, and behavioral decision theory

Frank L Lewis, National Academy of Inventors, IEEE Fellow, IFAC Fellow, Fellow Inst. Measurement & Control, PE Texas, U.K. Chartered Engineer. Distinguished Scholar Professor, Distinguished Teaching Professor, and Moncrief-O'Donnell Chair at University of Texas at Arlington. Works in feedback control and intelligent systems. Author of six US patents, 15 books, and several journal papers. Awards include Fulbright Research Award, NSF Research Initiation Grant, ASEE Terman Award, International Neural Network Society Gabor Award, and IEEE CIS Neural Network Pioneer Award. He was selected as Engineer of the Year by Ft. Worth IEEE Section.