Specification Based Intrusion Detection for Unmanned Aircraft Systems

Robert Mitchell
Department of Computer Science
Virginia Tech
Falls Church, VA
rrmitche@vt.edu

Ing-Ray Chen
Department of Computer Science
Virginia Tech
Falls Church, VA
irchen@vt.edu

ABSTRACT

In this paper, we propose a specification-based intrusion detection system for securing infrastructure (sensors or actuators) embedded in an unmanned aircraft system (UAS) in which continuity of operation is of the utmost importance. We investigate the impact of attacker behaviors on the effectiveness of our malware detection technique. Using unmanned air vehicles (UAVs) as examples, we demonstrate that our intrusion detection technique can effectively trade false positives off for a high detection probability to cope with more sophisticated and hidden attackers to support ultra safe and secure UAS applications.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection (e.g., firewalls)*

Keywords

intrusion detection, unmanned aircraft system, uas, cyber physical system

1. INTRODUCTION

UASs comprise a large part of the warfighting capability of modern militaries. Also, they are emerging in civilian applications such as surveillance for law enforcement, situational awareness for emergency services, content for news outlets and data collection for researchers. While they pose the same risk as piloted aircraft, the operator is removed from the vehicle in time and space; this calls for enhanced automated security systems to guarantee the safe operation of UASs.

The state of the art in intrusion detection for airborne networks and communications is limited. Trafton and Pizzi [7] did an investigation which motivated and broadly described the role of intrusion detection in this application but did not propose let alone measure a solution. Lauf and Robinson [2] investigated Distributed Apt Resource Transference System (DARTS). They built DARTS on their prior work, HybrIDS [3], which prompts (triggers) resource reallocation. HybrIDS is an anomaly based approach. Specifically, HybrIDS is a semi-supervised approach; the first of its two stages of operation is a training phase. Lauf, et al. measure the performance of HybrIDS using pervasion, which they define as the percentage of bad nodes in the system.

One broader research area that could encompass intrusion detection for airborne networks and communications are ad hoc networks. However, ad hoc networks typically extend an 802.11 technology; therefore, their typical radio range is 250 m. Long distance, high latency, low capacity communication links (such as line of sight VHF, LOS UHF or SATCOM) and processor and storage constraints distinguish airborne networks and communications intrusion detection within this larger research area. Some researchers argue that reputation management is a more natural fit than traditional intrusion detection for ad hoc networks.

Another broader research area that could encompass intrusion detection for airborne networks and communications are cyber physical systems (CPSs). CPSs have multiple control loops, strict timing requirements, a wireless network segment, predictable network traffic and contain legacy components [6]. CPSs fuse cyber (network components and commodity servers) and physical (sensors and actuators) domains. They may contain human actors and mobile nodes. A high degree of mobility distinguishes airborne networks and communications intrusion detection from this larger research area. CPSs may operate in locations that are dangerous due to heat, hazardous materials or violence. The focuses for CPS IDSs are leveraging unique CPS traits (sensor inputs, algorithms and control outputs) and detecting unknown attacks.

This work proposed and measured a specification based IDS for UASs. It requires minimal run time resources, detects unknown attacks and effectively trades between false positive and detection rates. We consider specification based detection rather than signature based detection to deal with unknown attacks. We consider specification based rather than anomaly based techniques to avoid high false positives (treating good nodes as bad nodes) in mission-critical UASs.

2. SYSTEM MODEL

2.1 Reference UAS

We consider a UAS comprising tens or hundreds of unmanned air vehicles (UAVs) each embedding CPS physical components (sensors and actuators) as a reference UAS

Airborne'12, June 11, 2012, Hilton Head, South Carolina, USA.

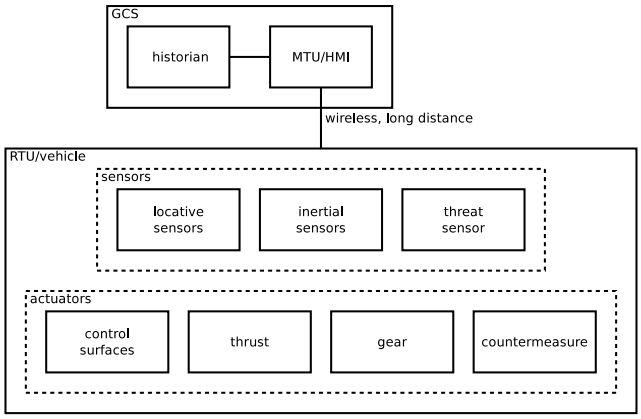

Figure 1: UAS.

model. For example, the United States Air Force 42d Attack Squadron has 18 assigned [1]. Figure 1 illustrates the reference UAS. One cyber physical loop in this model is the flight control system in each UAV. Inertial sensors drive realtime adjustment of control surfaces and thrust. In addition, locative sensors (navigational components), such as Global Positioning System (GPS), Global Navigation Satellite System (GLONASS), Compass, Galileo or inertial sensors drive non-realtime adjustment of control surfaces and thrust. Another cyber physical loop in this model is the threat countermeasures system. Radar components detect physical presence of threats, and specifically tuned radios detect RF signatures of threats. These sensors drive the realtime deployment of countermeasures like flare, chaff and electronic countermeasures (ECM). On top of UAVs sit the historian and human machine interface (HMI) devices which may be replicated to provide UAS control functions over long distance. UASs have been the victim of actual cyber physical attacks [5, 8].

2.2 Threat Model

We consider five threats when modeling a UAS. The first threat is an attacker that violates the integrity of the CPS in order to direct a UAV's weapon against a friendly resource. The second threat is an attacker that violates the integrity of the CPS in order to decrease a UAV's endurance by wasting its energy. The third threat is an attacker that violates the integrity of the CPS in order to increase the visibility/vulnerability of a UAV by activating its countermeasures unnecessarily. The fourth threat is an attacker that violates the integrity of the CPS in order to capture the UAV. The fifth threat is an attacker that violates the privacy of the CPS in order to exfiltrate mission data. Our IDS approach is based on behavior rules specifying expected good behaviors of a device (e.g., a sensor or actuator embedded in a UAV) for detection of the above threats.

2.3 Attacker Archetypes

We differentiate two attacker archetypes: reckless and random. A reckless attacker performs attacks whenever it has a chance. The main objective is to impair the UAS functionality at the earliest possible time. A random attacker, on the other hand, performs attacks only randomly to avoid detection. It is thus insidious and deceptive with the objective to cripple the UAS functionality. We model the attacker behavior by a random attack probability p_a. When $p_a = 1$ the attacker is a reckless adversary.

Table 1: UAV Behavior Rules

Description	Trustee	Monitor
Safe weapons if outside target area	UAV	UAV or HMI
Use minimum thrust if loitering	UAV	UAV or HMI
Turn off countermeasures if no threat	UAV	UAV or HMI
Stow landing gear if outside air base	UAV	UAV or HMI
Do not send to non-whitelisted destinations	UAV	UAV or HMI
Produce accurate data	UAV	UAV or HMI
Provide true recommendations	UAV	UAV or HMI

3. UAS INTRUSION DETECTION DESIGN

3.1 Behavior Rules

Our IDS design for the reference UAS model relies on the use of lightweight specification based *behavior rules* for each sensor or actuator component embedded in a UAV. They are oriented toward detecting an inside attacker attached to a specific physical component, provide a continuous output between 0 and 1 and allow a monitor device to perform intrusion detection on a neighboring trustee through simple monitoring. One possible design is to have a sensor (actuator) monitor another sensor (actuator respectively) within the same UAV. However, this design requires each sensor (actuator) to have multiple sensing functionalities. Another design which we adopt is to have a neighbor UAV or a remote HMI monitor a UAV. Table 1 lists the UAS behavior rules for the UAVs. This table specifies the trustee and monitor devices for applying our IDS technique.

3.2 Transforming Rules to State Machines

The following procedure transforms a behavior specification into a state machine: First we identify the "attack state" as a result of a behavior rule being violated. Then we transform this attack state into a conjunctive normal form predicate and identify the involved state components in the underlying state machine. Next we combine the attack states into a Boolean expression in disjunctive normal form. Then we transform the union of all predicate variables into the state components of a state machine and establish their corresponding ranges. Finally we manage the number of states by state collapsing and identifying combinations of values that are not legitimate. Below we exemplify how a state machine is derived from the behavior specification in terms of behavior rules for the reference UAS model.

3.2.1 Identify Attack States

Attacks performed by a compromised sensor/actuator embedded in a UAV will drive the UAV into certain attack states identifiable through analyzing the specification based behavior rules. There are seven attack states as a result of violating the seven behavior rules for a UAV listed in Table 1.

Table 2: UAV Attack States in Conjunctive Normal Form

(Weapons = READY) ∧ (Location ≠ TARGET)
(Thrust > T) ∧ (Status = LOITER)
(Countermeasures = ACTIVE) ∧ (Threat = FALSE)
(Gear = DEPLOYED) ∧ (Location ≠ AIRBASE)
Destination ≠ WHITELISTED
|Trustee Data - Monitor Data| > δ
Trustee Audit ≠ Monitor Audit

The first UAV attack state is that a UAV readies its weapon when outside its target area (not within its air base, ingress corridor or egress corridor). This state catches attackers that intend to direct a UAV's weapon against a friendly resource; these attackers attach to the UAV weapon module. The second UAV attack state is that a loitering UAV uses more than the minimum thrust required to maintain altitude. This state catches attackers that intend to decrease a UAV's endurance by wasting its energy; these attackers attach to the UAV thrust module. The third UAV attack state is that a UAV uses countermeasures without identifying a threat. This state catches attackers that intend to increase the vulnerability of a UAV; these attackers attach to the UAV countermeasures module. A fourth UAV attack state is that a UAV deploys landing gear when outside its air base. This state catches attackers that intend to capture the UAV; these attackers attach to the UAV landing gear module. One way an attacker could pursue this goal is to launch a shellcode attack that diverts the UAV to an area they control. A fifth UAV attack state is that a node sends bytes to unauthorized parties. This state catches attackers that intend to exfiltrate mission data; these attackers attach to the database. A sixth UAV attack state is that a trustee's embedded sensor reading differs from the monitor's embedded sensor reading. The monitor is in the neighborhood of the trustee, measuring the same physical phenomenon. A seventh UAV attack state is that a monitor provides badmouthing attacks, i.e., providing bad recommendations toward a well-behaving trustee node, or good-mouthing attacks, i.e., providing good recommendations toward a misbehaving trustee node. This is detected by comparing recommendations provided by multiple monitor nodes and detecting discrepancies.

3.2.2 Express Attack States in Conjunctive Normal Form

Table 2 lists the UAV attack states in Conjunctive Normal Form.

3.2.3 Consolidate Predicates in Disjunctive Normal Form

((Weapons = READY) ∧ (Location ≠ TARGET)) ∨ ((Thrust > T) ∧ (Status = LOITER)) ∨ ((Countermeasures = ACTIVE) ∧ (Threat = FALSE)) ∨ ((Gear = DEPLOYED) ∧ (Location ≠ AIRBASE)) ∨ (Destination ≠ WHITELISTED) ∨ (|Trustee Data - Monitor Data| > δ) ∨ (Trustee Audit ≠ Monitor Audit)

3.2.4 Manage State Space

To manage the number of states, we reduce the size of the state machine by abbreviating the values for and consolidating some components. Our rules only consider three values for position (air base, corridor or target), so we consolidate latitude, longitude and altitude into a single position component and restrict this component to three values. Our rules only consider four values for flight status (takeoff, travel, loiter or land), so we consolidate bank, pitch, yaw, stall, aileron, elevator and rudder into a single status component and restrict this component to four values. Our rules only consider three values for thrust (sub-stall, minimal or super-minimal) so we restrict this component to three values. This treatment yields a modest state machine with $2 \times 2 \times 2 \times 2 \times 3 \times 4 \times 2 \times 3 \times 2 \times 2 = 4608$ states, out of which 165 are identified as safe states and 4443 are unsafe states.

3.2.5 Behavior Rule State Machine

The state machine consisting of 165 safe states and 4443 unsafe states based on the behavior rules is generated as follows. First we label these states as states 1, 2, ..., n=4608. Next we assign p_{ij}, the probability that state i goes to state j, for each (i, j) pair in the state machine to reflect a good (or bad) UAV's behavior.

For a compromised UAV, p_{ij} depends on its attacker type: A reckless attacker will not go from an unsafe state to a safe state because it continuously attacks. So $p_{ij} = 0$ if i is a bad state and j is a good state. For a random attacker with attack probability p_a, p_{ij} values sum to p_a for a given i for all bad states j; p_{ij} values sum to $1 - p_a$ for a given i for all good states j because it will stop attacking with probability $1 - p_a$. Thus, for a compromised UAV with random attack probability p_a, p_{ij} is $(1 - p_a)/165$ when j is one of the 165 good states, and is $p_a/4443$ when j is one of the 4443 bad states.

For a good UAV, it will stay in safe states most of the time; occasionally it may go to an unsafe state due to non-behavior related faults such as communication faults or application specific human faults. The p_{ij} values associated with transitions from good states to bad states thus are exceptionally low. Thus, for a good UAV, p_{ij} is 0.99/165 when j is one of the 165 good states, and is 0.01/4443 when j is one of the 4443 bad states. Here 0.01 accounts for the error probability due to occasional communications or human faults.

3.3 Collect Compliance Degree Data

We use the state machines to collect compliance degree data of a good and a bad UAV during the system testing and debugging phase before it is released for operational use. Let c be the compliance degree of a node. With the above formulation, it is calculated as the sum of the products of the each state's grade and probability, i.e., $c = \sum_j c^j \times \pi_j$, where c^j is the "grade" assignment to state j, measuring the closeness between the observed behavior (in state j) and the specified "good" behavior, and π_j is the limiting probability that the node is in state j of the state machine. We consider a binary grading policy, i.e., c_j is 1 if state j is a safe state, and is 0 otherwise. With binary grading, the compliance degree c of a device essentially is equal to the proportion of the time the device is in safe states.

We collect compliance degree history $c_1, c_2, ..., c_n$ of a device by means of Monte Carlo simulation. That is, given a device's state machine we start from state 0 and then follow

the stochastic process of this device as it goes from one state to another. We continue doing this until at least one state is reentered sufficiently often (say 100 times). Then we calculate π_j using the ratio of the number of transitions leading to state j to the total number of state transitions. Then we collect one instance of compliance degree. We repeat a sufficiently large n test runs to collect $c_1, c_2, ..., c_n$ needed for computing the distribution of the compliance degree of a good or a bad UAV performing reckless or random attacks.

3.4 Compliance Degree Distribution

The measurement of compliance degree of a device frequently is not perfect and can be affected by noise and unreliable wireless communication in the UAS. We model the compliance degree by a random variable X with $G(\cdot) = Beta(\alpha, \beta)$ distribution [4], with the value 0 indicating that the output is totally unacceptable (zero compliance) and 1 indicating the output is totally acceptable (perfect compliance), such that $G(a)$, $0 \leq a \leq 1$, is given by

$$G(a) = \int_0^a \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}\, dx \qquad (1)$$

and the expected value of X is given by

$$E_B[X] = \int_0^1 x\, \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}\, dx = \frac{\alpha}{\alpha+\beta} \qquad (2)$$

The α and β parameters are to be estimated based on the method of maximum likelihood by using the compliance degree history collected $(c_1, c_2, ..., c_n)$ during the system's testing phase. We consider a single parameter $Beta(\beta)$ distribution with α equal to 1. In this case, the density is $\beta(1-x)^{\beta-1}$ for $0 \leq x \leq 1$ and 0 otherwise. The maximum likelihood estimate of β is

$$\hat{\beta} = \frac{n}{\sum_{i=1}^{n} \log(\frac{1}{1-c_i})} \qquad (3)$$

3.5 False Positive and Negative Probabilities

Our intrusion detection is characterized by false negative and false positive probabilities, denoted by p_{fn} and p_{fp}, respectively. A false positive occurs when a good UAV is misdiagnosed as bad, while a false negative occurs when a bad UAV is missed as good. While neither is desirable, a false negative is especially impactful to the system's continuity of operation. In this paper we consider a threshold criterion. That is, if a bad node's compliance degree denoted by X_b with a probability distribution obtained by Equation 1 above is higher than a system minimum compliance threshold C_T then there is a false negative. Suppose that the compliance degree X_b of a bad node is modeled by a $G(\cdot) = Beta(\alpha, \beta)$ distribution as described above. Then the host IDS false negative probability p_{fn} is given by:

$$p_{\text{fn}} = \Pr\{X_b > C_T\} = 1 - G(C_T). \qquad (4)$$

On the other hand, if a good node's compliance degree denoted by X_g is less than C_T then there is a false positive. Again suppose that the compliance degree X_g of a good node is modeled by a $G(\cdot) = Beta(\alpha, \beta)$ distribution. Then the host false positive probability p_{fp} is given by:

$$p_{\text{fp}} = \Pr\{X_g \leq C_T\} = G(C_T). \qquad (5)$$

Table 3: β in Beta(1,β) and Resulting p_{fn} and p_{fp} Values under Various Attack Models for UAV (C_T = 0.90).

Attack Type	β	p_{fn}	p_{fp}
Random ($p_a = 1.00$)	99.3	$< 0.001\%$	7.39%
Random ($p_a = 0.80$)	4.33	0.005%	7.39%
Random ($p_a = 0.40$)	1.10	8.02%	7.39%
Random ($p_a = 0.20$)	0.632	23.3%	7.39%
Random ($p_a = 0.10$)	0.449	35.5%	7.39%
Random ($p_a = 0.05$)	0.353	44.3%	7.39%

4. NUMERICAL DATA

We report numerical data in this section. We execute the procedure described in Section 3 to collect a sequence of compliance degree values $(c_1, c_2, ..., c_n)$ for $n = 1000$ Monte Carlo simulation test runs for the UAV. We then apply Equation 3 to compute the β parameter value of $G(\cdot) = Beta(\alpha, \beta)$ for the probability distribution of the compliance degree for a good or a bad device controlled by malware to perform random attacks. We then calculate p_{fn} and p_{fp} by Equations 4 and 5, respectively. We adjust the minimum compliance threshold C_T to control p_{fn} and p_{fp} obtainable.

Table 3 shows the β values and the resulting p_{fn} and p_{fp} values when C_T is 0.9 (C_T is a design parameter to be fine-tuned to trade high false positives for low false negatives as described below), and the binary grading strategy is being used to assign c^j to state j. We observe that when the random attack probability p_a is high, the attacker can be easily detected as evidenced by a low false negative probability. Especially when $p_a = 1$, a reckless attacker can hardly be missed. On the other hand, as p_a decreases, the attacker becomes more hidden and insidious and the false negative probability increases. The false positive probability remains the same regardless of the random attack probability because it is not related to the attacker behavior.

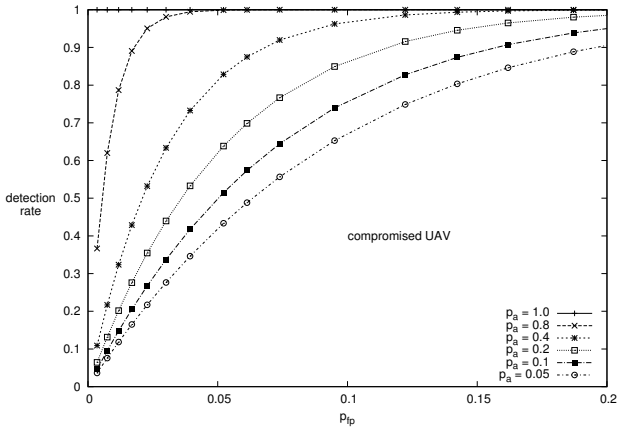

Figure 2: UAV Receiver Operating Characteristic Graph.

By adjusting C_T, our specification based IDS technique can effectively trade higher false positives off for lower false negatives to cope with more sophisticated and hidden random attackers. This is especially desirable for ultra safe

and secure UAS applications for which a false negative may have a dire consequence. Figure 2 shows a *Receiver Operating Characteristic* (ROC) graph of intrusion detection rate (i.e., $1 - p_{\text{fn}}$) vs. false positive probability (p_{fp}) obtained as a result of adjusting C_T. In Figure 2 there are several curves for each node type, one for each random attacker case with a different attack probability p_a. As we increase C_T, the detection rate increases (vertically up on a ROC graph) while the false probability increases (toward the right of a ROC graph). We see that with our specification based IDS technique, the detection rate of the UAS node can approach 100% for detecting attackers, i.e., an attacker is always detected with probability 1 without false negatives, while bounding the false positive probability to below 5% (for reckless attackers) to 20% (for random attackers).

5. CONCLUSIONS

For UASs, being able to detect attackers while limiting the false alarm probability is of utmost importance to protect the continuity of operation. In this paper we proposed a behavior-rule specification based IDS technique for intrusion detection of compromised UAVs in a UAS. We demonstrated that the detection probability approaches one (that is, we can always catch the attacker without false negatives) while bounding the false alarm probability to below 5% for reckless attackers and below 20% for random attackers.

6. REFERENCES

[1] `http://en.wikipedia.org/wiki/42d_Attack_Squadron`.

[2] A. Lauf and W. Robinson. Fault-tolerant distributed reconnaissance. In *IEEE Military Communications Conference*, pages 1812–1817, November 2010.

[3] A. P. Lauf, R. A. Peters, and W. H. Robinson. A distributed intrusion detection system for resource-constrained devices in ad-hoc networks. *Ad Hoc Networks*, 8(3):253–266, 2010.

[4] S. M. Ross. *Introduction to Probability Models, 10th Edition*. Academic Press, 2009.

[5] `http://www.cnn.com/2011/12/08/world/meast/iran-drone`.

[6] S. Shin, T. Kwon, G.-Y. Jo, Y. Park, and H. Rhy. An experimental study of hierarchical intrusion detection for wireless industrial sensor networks. *IEEE Transactions on Industrial Informatics*, 6(4):744–757, November 2010.

[7] R. Trafton and S. Pizzi. The joint airborne network services suite. In *IEEE Military Communications Conference, 2006.*, pages 1–5, October 2006.

[8] `http://security.blogs.cnn.com/2011/10/13/in-rare-admission-air-force-explains-and-downplays-drone-computer-virus`.