AIAA SciTech Forum
7-11 January 2019, San Diego, California
AIAA Scitech 2019 Forum

10.2514/6.2019-1459



# An Interface-Based Cybersecurity Subsystem Analysis on a Small Unmanned Aerial Systems

Corey A. Ippolito[1], Kalmanje Krishnakumar[2]
*NASA Ames Research Center, Moffett Field, CA, 94035*

**The small Unmanned Aerial System (sUAS) industry continues to make rapid advances in technology, features, and capabilities, with an associated increase in complexity and sophistication. The rapid emergence of sUAS as complex information technology (IT) system-of-systems (SoS) and the increasing connectivity of sUAS components has placed renewed emphasis on the need to establish formal cybersecurity analysis methods for this domain. A common issues has arisen where new and previously unanalyzed vulnerabilities are reported on a system within an existing and previously-certified UAS SoS architecture, which may require UAS to be taken out of operation until the new cybersecurity vulnerabilities are analyzed, mitigated, verified, then recertified for return to flight operational status. Unfortunately this is a time-consuming and resource-intensive process, and is hindered by the lack of formal methods for cybersecurity analysis on UAS in general, and particularly methods that can be applied to address issues of SoS component-level vulnerabilities. In this paper, we present an interface-based analysis methodology for analyzing cybersecurity vulnerabilities for subsystem components within an established UAS system architecture. This methodology provides a risk-based assessment through analysis of data and information movement across interfaces within a SoS architecture model. We present an information theoretic formalization of the models used for analysis then present the methodology on these models to assess cybersecurity vulnerablilies. Finally, we apply this framework to a NASA sUAS system where a potential vulnerability was identified and this process was applied. We present a summary of the issue, the UAS SoS models, the resulting analysis of vulnerabilities, describe mitigations, and present final risk assessmement used towards recertification of the aircraft.**



## I. Introduction

The small Unmanned Aerial System (sUAS) industry continues to make rapid advances in technology, features, and capabilities, with an associated increase in complexity and sophistication. The rapid emergence of sUAS as complex information technology (IT) system-of-systems (SoS) and the increasing connectivity of sUAS components – such as smart-tablet-based ground control stations with live streaming over the Internet – has placed renewed emphasis on establishing formal cybersecurity methods for this domain. A common issue has arisen where new and previously unanalyzed vulnerabilities are reported on a system within an existing and previously-certified UAS SoS architecture, which may require UAS to be taken out of operation until the new cybersecurity vulnerabilities are analyzed, mitigated, verified, then recertified to return to flight operational status. Unfortunately this is a time-consuming and resource-intensive process, and is hindered by the lack of formal methods for cybersecurity analysis on UAS in general, and particularly methods that can be applied to address the issue of newly identified vulnerabilities within subsystems.

This document presents a preliminary cybersecurity threat analysis methodology developed in response to identification of new potential cybersecurity vulnerabilities that were common to several flight system components used onboard small Unmanned Aerial System (sUAS) platform configurations at NASA Ames Research Center (ARC). The scope of this concern was limited to assessing and mitigating the Information Technology (IT) threat of existing components within the approved configuration, which had been previously reviewed and certified for airworthiness. Under direction from the Ames Aircraft Management Office (AMO), a review was conducted of the

[1] Aerospace Scientist, NASA Ames Research Center, Moffett Field, CA 94035, AIAA Senior Member.
[2] NASA Ames Research Center, Moffett Field, CA 94035, AIAA Senior Member.

potential vulnerability which identified the list of flight system components that may have been suspect to this vulnerability. The list of vulnerable flight system hardware components were compared to each sUAS platform configuration approved for operation. The vehicle configurations that contained components with suspect vulnerabilities were subsequently removed from flight operational status until the cybersecurity vulnerability was sufficiently addressed by the engineering team, then reviewed and approved by the Ames AMO through an Airworthiness and Flight Safety and Review (AFSR).

This paper presents an interface-based analysis framework for analyzing IT vulnerabilities developed to addess vulnerabilities due to suspect components within an established UAS systems architecture. This framework provides a methodology applicable to assessing cybersecurity vulnerabilities posed by these suspect components in an existing system-of-systems (SoS) architecture. Through analysis of data and information movement across these interfaces, an assessment is made as to cybersecurity vulnerabilities introduced. Mitigations to these vulnerabilities are then designed and re-evaluated in terms of severity and likelihood, which is then utilized for the final assessment of overall cybersecurity risk. We present an information theoretic formalization of the models used for analysis, then present the methodology on these models to assess cybersecurity vulnerablilies. We apply this framework to a NASA sUAS system where a potential vulnerability was identified and this process was applied. We present details of the UAS, describe application of the process, and present the resulting risk analysis.



## II. Related Work

### A. sUAS Cybersecurity Issues

Small Unmanned Aerial Systems (sUAS) operating at NASA must be compliant with aircraft operations management requirements and risk management requirements as specified in NPR 7900.3D [1], with specific requirements for UAS operating at NASA Ames Research Center specified under APR 8000.4 [2]. As the small UAS industry advances in terms of market size, application, and technological sophistication, IT security and cybersecurity have been an increasing concern [3]-[8]. The Federal Aircraft Administration (FAA) has expressed a need to develop more robust, fault-tolerant UAS designs in response to cyber threats [3]. Cybersecurity is an outstanding need highlighted in the NASA/FAA UAS Traffic Management (UTM) project, where new security vulnerabilities and challenges need to be addressed as UAS resliance on interconnectivity and integration increases [4]-[8]. For military sUAS applications, the implications of a cybersecurity breach could put military missions at risk while endangering the lives of soldiers. Under the United States Department of Defense (DoD), sUAS are predominantly used to support intelligence, surveillance and reconnaissance (ISR) missions [9][10]. As of 2012, the United States military operates more than 7,500 drones making up more than forty percent of the DoD aircraft [11]. Smart-phone and smart-tablet technology utilized for ground station control has been an area of recent focus [10], as many of these devices do not meet federal requirements for IT security. For instance, BlackBerry smart devices were the only smart devices that met the Federal Information Processing Standards certification as of a report in 2013 [9][12].

A growing number of cybersecurity incidents have been documented in the recent literature [9]-[18]. In 2009, several reports indicate interception of live video from Predator drones [13]. In 2011, an incident was documented where keylogging malware was found in the Predator and Reaper ground control stations at Creech Air Force Base. The keylogging malware was introduced through a removable hard drive and spread to both classified and unclassified computer systems [15]. In July 2012, a University of Texas research team demonstrated the ability to hijack and take complete control of military UAS using using inexpensive ground equipment through spoofing of Global Positioning System (GPS) signals. In December 2012, an U.S. RQ-170 Sentinel UAV was hijacked and captured by the Iranian government on the Afghanistan border. The Iranian government was able to successfully land the UAV to obtain sensitive data including the mission and maintenance data [16]. Ground station vulnerabilities have been documented in other industries as well, such as incidents in 2008 where hackers took command of the Landsat-7 and TerraEOSAM-1satellites through compromised internet-connected ground control stations [19].

### B. Cybersecurity Threats on UAS

The Department of Homeland Security (DHS) Risk Lexicon [20] describes instances of operational cyber security risks in greater detail. A taxonomy of operational cybersecurity risks proposed by Cebula and Young [21] identifies four main categories of risk:

1. Actions of People - action, or lack of action, taken by people either deliberately or accidentally that impact cyber security;

2. Systems and Technology Failures - failure of hardware, software, and information systems;
3. Failed Internal Processes - problems in the internal business processes that impact the ability to implement, manage, and sustain cybersecurity, such as process design, execution, and control;
4. External Events - issues often outside the control of the organization, such as disasters, legal issues, business issues, and service provider dependencies.

These four main classes are further elaborated as summarized in Table 1.

**Table 1. Taxonomy of Operational Risk (Cebula and Young [21])**

| 1. Actions of People | 2. Systems and Technology Failures | 3. Failed Internal Processes | 4. External Events |
|---|---|---|---|
| **1.1 Inadvertent** | **2.1 Hardware** | **3.1 Process design or execution** | **4.1 Disasters** |
| 1.1.1 Mistakes | 2.1.1 Capacity | 3.1.1 Process flow | 4.1.1 Weather event |
| 1.1.2 Errors | 2.1.2 Performance | 3.1.2 Process documentation | 4.1.2 Fire |
| 1.1.3 Omissions | 2.1.3 Maintenance | 3.1.3 Roles and responsibilities | 4.1.3 Flood |
| **1.2 Deliberate** | 2.1.4 Obsolescence | 3.1.4 Notifications and alerts | 4.1.4 Earthquake |
| 1.2.1 Fraud | **2.2 Software** | 3.1.5 Information flow | 4.1.5 Unrest |
| 1.2.2 Sabotage | 2.2.1 Compatibility | 3.1.6 Escalation of issues | 4.1.6 Pandemic |
| 1.2.3 Theft | 2.2.2 Configuration management | 3.1.7 Service level agreements | **4.2 Legal issues** |
| 1.2.4 Vandalism | 2.2.3 Change control | 3.1.8 Task hand-off | 4.2.1 Regulatory compliance |
| **1.3 Inaction** | 2.2.4 Security settings | **3.2 Process controls** | 4.2.2 Legislation |
| 1.3.1 Skills | 2.2.5 Coding practices | 3.2.1 Status monitoring | 4.2.3 Litigation |
| 1.3.2 Knowledge | 2.2.6 Testing | 3.2.2 Metrics | **4.3 Business issues** |
| 1.3.3 Guidance | **2.3 Systems** | 3.2.3 Periodic review | 4.3.1 Supplier failure |
| 1.3.4 Availability | 2.3.1 Design | 3.2.4 Process ownership | 4.3.2 Market conditions |
| | 2.3.2 Specifications | **3.3 Supporting processes** | 4.3.3 Economic conditions |
| | 2.3.3 Integration | 3.3.1 Staffing | **4.4 Service dependencies** |
| | 2.3.4 Complexity | 3.3.2 Funding | 4.4.1 Utilities |
| | | 3.3.3 Training and development | 4.4.2 Emergency services |
| | | 3.3.4 Procurement | 4.4.3 Fuel |
| | | | 4.4.4 Transportation |

Similary, Javaid et al. in [22] propose a system cybersecurity threat model taxonomy for analysis of a general large-scale system of interconnected vehicles. The authors present a high-level diagram of the system, model communication between elements, and decompose the vehicle element into a proposed block diagram. The authors enumerate possible cybersecurity threats under the cybersecurity threat model and hierarchy presented in Figure 2. Each threat is then analyzed through a risk-based assessment adapted from [23], which utilizes a three-category likelihood and impact assessment to identify risk. The authors extend this analysis with additional subjective criteria that are to be considered in likelihood and impact. The additional likelihood analysis criteria are 'difficulty' and 'motivation', and the additional impact criteria are an assessment of the impact to the end user and impact to system availability. The method by which additional evaluation criteria are to be incorporated is not formalized. The risk assessment results included explicit design requirements surrounding need for counter-measure mitigations.

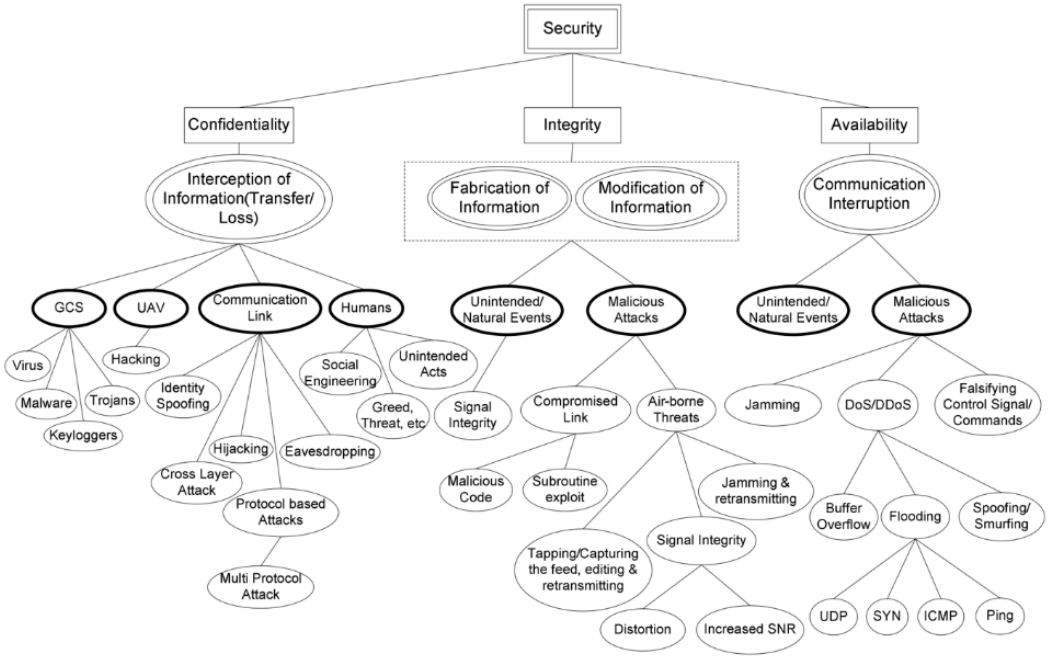
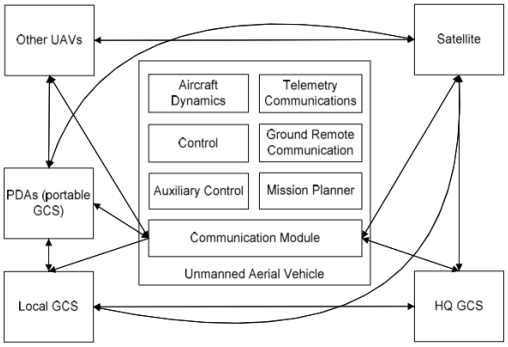
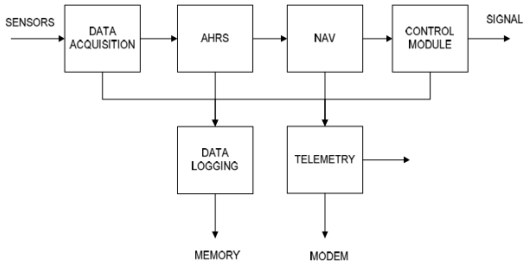



**Figure 1. UAS Cybersecurity Threat Model (from [22])**

A high-level cybersecurity risk assessment is also provided for a general UAS system in [22]. The process is summarized in Figure 2. A hierarchical communication architecture model is developed, with the highest level is shown in (a) and the airborn UAV element is shown in (b). A UAV block diagram is shown in (c). The definition of risk in terms of likelihood and impact is subjective, requiring subject matter expert (SME) input to generate a rating of 1-3 for each the difficulty and motivation, as shown in the table in (d). For each thread, SME then list the 1-3 rating results, the analysis summary is shown in (e). A formal methodology is not presented, and the evaluation relies on subjective SME input to establish the rating.




(a) UAV Communication Model




(b) Simple UAV Block Diagram

| | | *Rationale* | | |
|---|---|---|---|---|
| **Criteria** | **Cases** | **Difficulty** | **Motivation** | **Ranks** |
| Likelihood | Unlikely | Strong | Low | 1 |
| | Possible | Solvable | Reasonable | 2 |
| | Likely | None | High | 3 |
| | | ***User*** | ***System*** | |
| Impact | Low | Annoyance | Very Limited Outages | 1 |
| | Medium | Loss of Service (LoS) | Limited Outages | 2 |
| | High | Long time LoS | Long time Outages | 3 |
| Risk | Minor | No need for countermeasures | | 1,2 |
| | Major | Threat need to be handled | | 3,4 |
| | Critical | High priority | | 6,9 |

(c) Risk Evaluation Grid

| **Threat** | **Algorithm(s)** | **Likelihood** | **Impact** | **Risk** |
|---|---|---|---|---|
| Jamming | | 3 | 1 | 3 |
| Scrambling/Distortion | | 2 | 1 | 2 |
| Eavesdropping | | 3 | 2 | 6 |
| Cross Layer Attacks | | 2 | 1 | 2 |
| Multi-Protocol Attack | | 2 | 1 | 2 |
| Social Engineering | | 2 | 2 | 4 |
| Spoofing | Device List | 3 | 3 | 9 |
| | X.509 device Auth. | 2 | 3 | 6 |
| Command and Control Message Modification | No MAC | 3 | 3 | 9 |
| | SHA-1 MAC | 2 | 3 | 6 |
| | AES MAC | 1 | 3 | 3 |
| Data Traffic Modification | Without AES | 3 | 1 | 3 |
| | With AES | 1 | 1 | 1 |
| DoS on UAV/GCS | EAP/SHA-1/AES/MAC | 3 | 3 | 9 |
| Signal Integrity | | 3 | 2 | 6 |
| Malicious Code, Subroutine Exploit | | 1 | 3 | 3 |
| Virus, Malware, Trojans and Keyloggers | | 3 | 2 | 6 |

(d) Analysis Summary

**Figure 2. Example Cybersecurity Risk Analysis (as presented in [22])**

A 2015 U.S. Department of Defense (DoD) Acquisition Research Journal (ARL) publication [10] presents a cybersecurity analysis for smart-devices used as a ground control station (GCS) for small UAS. The authors present a seven-step process for generating a threat model: (1) characterize the system, (2) understand the adversary's objectives, (3) identify system assets and vulnerabilities, (4) identify threats and attacks, (5) conduct threat analysis and prioritization, (6) identify countermeasures, and (7) determine the mitigation plan. The Mansfield threat model categorized threats in four vulnerability catagories: hardware, software, humans, and communication networks. The threat analysis process utilizes likelihood on a scale of 0.0 to 1.0, and impact on a scale of 0 to 100. Similarly to previous risk analyses, the methodology for determining likelihood/impact ratings are subjective based on SME input, and the risk model presented only used three values in the assessment (0.1/10, 0.5/50, or 1.0/100). The resulting risk value for each threat is determined by multiplying the likelihood rating with the impact rating. The vulnerabilities, threats, and mapping to IT security is shown in Figure 3. The risk assessment results are summarized in Figure 4.

| **Vulnerabilities** | **Threat** | **Security Objectives** | | |
|---|---|---|---|---|
| | | *Confidentiality* | *Integrity* | *Availability* |
| **Hardware** | *Battery Exhaustion* | | | X |
| | *Flooding* | | X | X |
| | *Surveillance* | X | X | |
| | *USB* | X | X | |
| **Software** | *Malware* | X | X | X |
| | *Phising* | | X | X |
| | *Data Leakage* | X | | |
| **Communication Network** | *Eavesdropping* | X | | |
| | *Spoofing* | X | X | |
| | *Denial of Service* | | | X |
| | *Jamming* | | | X |

**Figure 3. Table of Categorized Threats from Smart-Devices (as presented in [10]).**

| TABLE 4. RISK ANALYSIS SUMMARY | | | |
|---|---|---|---|
| **Threat** | **Likelihood** | **Impact** | **Risk** |
| **HARDWARE** | | | |
| Battery Exhaustion | 0.5 | 100 | 50 |
| Flooding | 1.0 | 50 | 50 |
| Surveillance | 1.0 | 100 | 100 |
| USB | 0.1 | 10 | 1 |
| Storage Snooping | 0.5 | 50 | 25 |
| Storage Jamming | 0.5 | 10 | 5 |
| Storage Erasure/Alteration | 0.1 | 50 | 5 |
| **SOFTWARE** | | | |
| Malware | 1.0 | 100 | 100 |
| Phishing | 0.5 | 50 | 25 |
| Data Leakage | 1.0 | 50 | 50 |
| Spyware | 1.0 | 100 | 100 |
| Data Tampering | 1.0 | 50 | 50 |
| Elevation of Privilege | 1.0 | 100 | 100 |
| **COMMUNICATION NETWORK** | | | |
| Eavesdropping | 1.0 | 100 | 100 |
| Spoofing | 0.5 | 100 | 50 |
| Denial of Service | 1.0 | 100 | 100 |
| Jamming | 1.0 | 10 | 10 |
| Weak/Compromised Cryptography | 0.5 | 50 | 25 |
| Unencrypted Communication | 0.1 | 50 | 5 |
| Impaired Quality of Service | 0.5 | 100 | 100 |
| **HUMAN** | | | |
| Breaking Policy | 1.0 | 100 | 100 |
| Inadequate Policy | 1.0 | 100 | 100 |
| Unencrypted Communication | 0.5 | 50 | 25 |
| Carelessness with Cryptographic Keys | 1.0 | 50 | 50 |
| Harmful Data Leakage | 0.5 | 50 | 25 |
| Compromise of Personnel | 0.5 | 100 | 50 |
| Poor Risk Decisions | 0.5 | 100 | 50 |
| Poor Management/Maintenance | 1.0 | 100 | 100 |
| Overloading the Operator | 0.5 | 10 | 5 |
| Prevention of Accountability from Being Stored | 0.1 | 10 | 1 |
| Destruction of Accountability Data | 0.1 | 10 | 1 |
| Modification of Accountability Data | 0.1 | 10 | 1 |

**Figure 4. Cybersecurity Risk Assemement (as presented in [10])**

### C. Threat Models and Analysis

Many approaches to threat modeling have been presented in the literature across many domains [24]-[27]. Unfortunately, threat modeling methods are generally not comprehensive. A challenge for threat model-based approaches is the need to continually maintain and evolve these models over all phases of the life-cycle of a system, from concept to end-of-life decommissioning. As technology evolves and new threats emerge, threat models must be continually reviewed, revised and updated to avoid gaps [28].

Several processes have been suggested to develop threat model assessments. Torr in [24] presents a general framework for analyzing threats to software systems during the software design and development phases. Abi-Antoun, Wang, and Torr expand on this in [29], proposing a threat modeling methodology based on data flow diagram models that verify conformance and security threats, such as spoofing, tampering and information disclosure. They present analysis and semi-formal proofs of completeness and correctness. Swiderski and Snyder in [30] present threat modeling methodologies and structured approaches for identifying, evaluating, and mitigating risks to system security. Oladimeji et al. in [31] propose a general process for threat modeling and analysis as shown in the activity diagram in Figure 5 below.

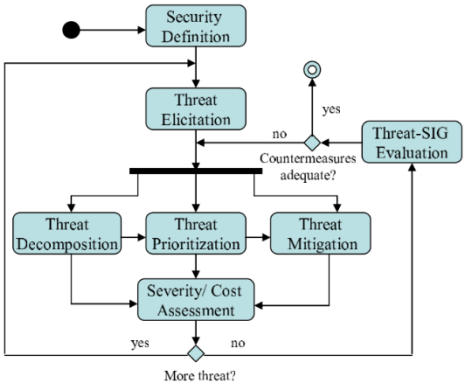



**Figure 5. Threat Modeling and Analysis Process (from [31])**

Data flow diagrams (DFD) are a well-established threat modeling technique used to analyze threats against components in a system (e.g., [24][30]). An example DFD from [24] is shown in Figure 6. Methodologies for analyzing threats directly from DFD for conformance and security in software were presented [29]. Definitions for DFD are shown in Figure 7. DFD can be utilized in a leveled approach that decomposes a large system from a single or smaller set of nodes; the highest level DFD is referred to as a Context Diagram in [30]. The node is then broken into multiple elements in a Level-0 diagram, and into subsequently lower diagrams (Level-1, Level-2, etc.). Conformance can then be checked between two Level-n DFDs. However, these definitions are specifically geared towards software systems, where basic diagram elements are defined, for instance, in terms of processes that capture software tasks and data stores for modeling databases and files.

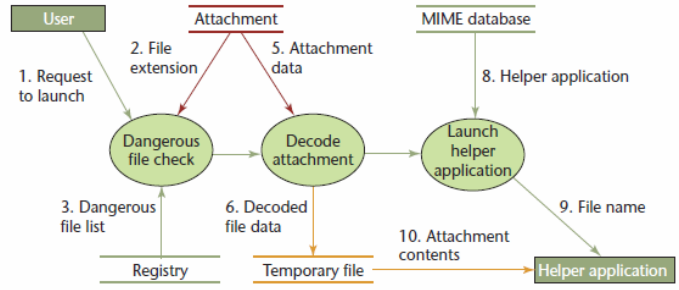



**Figure 6. Example Data-Flow Diagram (DFD) (from [24])**

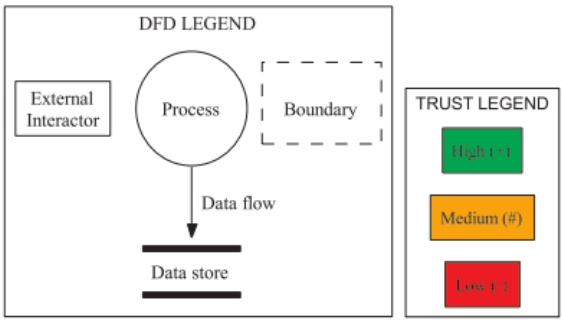



**Figure 7. Data-Flow Diagram Definition for Software Threat Modeling (from [29]).**

Another common threat modeling technique is based on attack trees [32][33][34]. Attack tree methods model threats against computer systems to help understand all the different ways in which a system can be attacked [32]. Attack tree methods provide a systematic way to analyze threats to a system by enumerating and analyzing attack goals in a top-down then bottom-up analysis process. The models are captured in attack tree graphs that can be maintained throughout the life of the project. Commercial attack tree modeling tools are widely available (e.g., [35]).

An example methodology for attack tree analysis is presented in [34], shown in Figure 8. Generally, the process for generating the initial attack tree model is as follows: (1) identify and enumerate all possible attack goals against a system, with each attack goal elaborated in a separate attack tree as the root node; (2) for all leaf nodes in the tree, enumerate all possible methods to achieve that goal, adding these methods as child nodes to the tree, (3) repeat this process and grow the tree downward until all leaf nodes are no longer decomposable (top-down), (4) analyze and label each leaf node based on pre-established metrics, (5) compute the metrics for each parent node (bottom-up) until the root node metric is computed. After the attack trees are generated, typical best-practices suggest the attack trees should be continually reviewed and maintained throughout the entire life-cycle of a project.

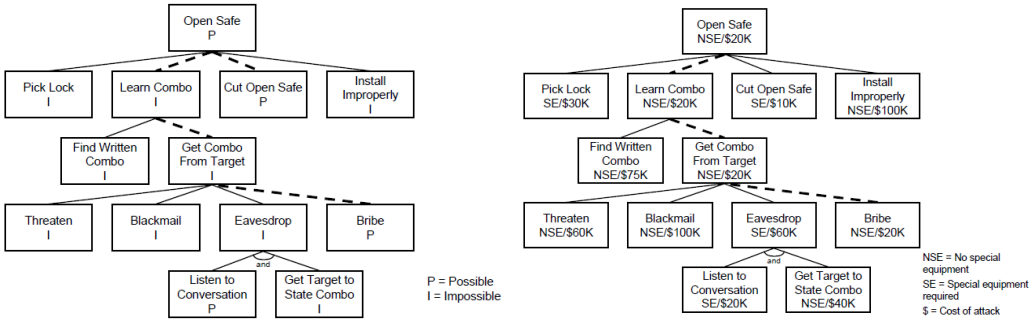



**Figure 8. Example Attack Tree Diagrams and Analysis Methodology (from [34])**

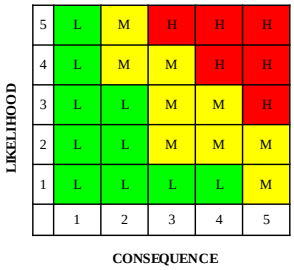
Oladimeji, Supakkul, and Chung in [31] present a method for security threat modeling using the formal semantics of the Non-Functional Requirements (NFR) framework. This methodology provides a goal-oriented approach to security threat modeling and analysis utilizing visual model elements to explicitly capture threat-related concept. The NFR framework is a software modeling tool that allows for goal-oriented design by soft-goals and provides qualitative assessments of design alternatives in terms of quality attributes [36]. In the proposed approach, threats are modeled as negative soft-goals under NFR. The NFR models should be utilized during early stages of the design process as an analysis tool to assist in evaluating design alternatives. The NFR models need to be maintained throughout the development process, updated and reanalyzed as the application evolves and requirements are better defined.

Guariniello and DeLaurentis propose a formal method for cybersecurity threat assessment on complex system-of-systems utilizing Functional Dependency Network Analysis (FDNA) [37][39]. FDNA establishes structural models of a system, the topology of which is determined by functional dependencies between system components. FDNA models can handle analysis of partial degradation and partial capabilities, and can measure robustness of the system to maintain operability in the presence of communication disruption. The proposed methodology applies metrics to dependencies along each directed edge of the network topology. Each dependency is characterized by a Strength of Dependency (SOD) metric that quantifies how much the behavior of a system depends on the behavior of another system, a Criticality of Dependency (COD) metric that quantifies the negative impact that a system has on another in critical conditions, and an Availability of Data (AOD) metric.

### D. Risk Analysis

A general issue with risk-based assessments is the level of subjectivity in the analysis, resulting in different risk values for different researchers, which may be limited by the information currently available to the analyst and level of granularity to which the analysis will be conducted (e.g., the depth of decomposion chosen for a DFD or attack tree) [22]. There have been many methods for assessing risk proposed in the literature. For example, a mathematical risk assessment method was proposed in [38], which presented an evaluation algorithm that estimates risk indices by layering based on the intruding process in a zero-sum network interdiction game. A risk-based approach to cybersecurity threats is presented in [23], which presents a matrix relationship between threats and security objectives. This document enumerates various lists of threats, then employs a risk analysis assessment to evaluate and classify threats into 3 categories. Countermeasures are determined to counter the critical and major risks, and security requirements are derived. For the risk assessment, the occurrence likelihood of threats is estimated with values from 1 to 3. The impact of a threat is also estimated with values from 1 to 3. Risk is then calculated as the multiplicative product of the likelihood metric with the impact metric, resulting in a risk metric from 1 to 9. NASA Procedural Requirements NPR 7900.3D specify general requirements for aircraft operations management and risk management. The specific risk methodology and definitions used in this analysis are specified in the NASA Ames Procedural

Requirements APR 8000.4. This document specifies a general process that utilizes a 5x5 risk matrix in the assessment of the severity of risk based on analysis of likelihood and consequence to element (Figure 5). The risk likelihood and consequence definitions are specified in NASA APR 8000.4 based on several categories: safety, cost, schedule, mission success, facilities/equipment/assets, and environmental.




| Attribute | | Level | | | | |
|---|---|---|---|---|---|---|
| | | 1 (Very Low) | 2 (Low) | 3 (Moderate) | 4 (High) | 5 (Very High) |
| Likelihood (L) Non-Human Safety Risks | | ≤ 0.01 | 0.01 < ≤ 0.10 | 0.10 < ≤ 0.33 | 0.33 < ≤ 0.50 | > 0.50 |
| Consequence (C) Non-Human Safety Risks | Cost | Overrun of ≤ 2% | Overrun of 2% < ≤ 5% | Overrun of 5% < ≤ 10% | Overrun of 10% < ≤ 15% | Overrun of > 15% |
| | Schedule | Overrun of ≤ 2% or No impact to critical path | Overrun of 2% < ≤ 5% or No impact to critical path | Overrun of 5% < ≤ 10% or ≤ 1 Month impact to critical path/milestones | Overrun of 10% < ≤ 15% or >1 to ≤ 6 Month impact to critical path/milestones | Overrun of > 15% or > 6 Month impact to critical path/milestones |
| | Mission Success (Technical Performance) | Loss of ≤ 2% success/exit criteria | Loss of 2% < ≤ 5% success/exit criteria | Loss of 5% < ≤ 10% success/exit criteria | Loss of 10% < ≤ 15% success/exit criteria | Loss of > 15% success/exit criteria |
| | Facilities, Equipment. or other Assets | More than normal wear and tear | Property damage $1K < ≤ $25K | Property damage $25K < ≤ $250K | Destruction of non-critical assets or damage $250K < ≤ $1M | Destruction of critical assets or damage > $1M |
| | Environmental | Non reportable OSHA/EPA violation | Reportable OSHA/EPA violation that does not require immediate remediation | Reportable OSHA/EPA violation which requires immediate remediation | Reportable OSHA/EPA violation causing temporary stoppage | OSHA/EPA violations resulting in termination of project |

**Figure 9. NASA 5x5 Risk Matrix (left), Likelihood and Consequence Definitions (right)**

## III. An Interface-Based Cybersecurity Methodology

### A. Problem Formulation

Given an existing cyber-physical system architecture containing a predefined set of suspect components, assess cybersecurity vulnerabilities and risks in this architecture posed by the set of suspect components. As a motivating example, consider a system-of-systems modeled as a graph as shown in Figure 10, where a cybersecurity assessment needs to be determined on component A. Here, components in a system are modeled as vertices labeled A through K, and edges in the graph will be utilized to capture different types of interdepencies between components in system.

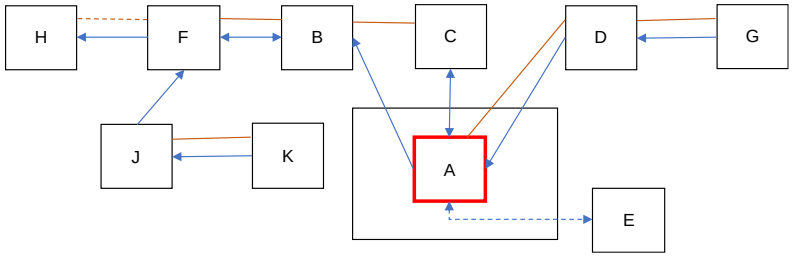



**Figure 10. Motivating Problem Model**

The methodology presented in this paper utilizes graphs to represent a system of systems. A graph G = (V,E) consists of a set V of vertices or nodes, and an edge set E ⊂ V × V. The elements of E are called edges or links. A directed edge is an ordered pair e = <u, v> ∈ E, where we say the edge e leads from the tail vertex u to the head vertex v, or starting at u and ending at v. Further u is called the initial vertex of e and v is called the terminal vertex of e.

Let G = (V,E) be a graph, and let v0 ∈ V. The connected component of v0 is the set of all v ∈ V such that there exists a path from v0 to v. The connected components of G are the subsets of vertices in V which arise as connected components of v0. For undirected graphs, the connected components of G = (V,E) partition V into disjoint subsets.

Let G = (V,E) be a graph. We say a (non-empty) sequence of vertices γ = (v1, v2, . . . , vr, vr+1) in V is a path or walk if (vi, vi+1) ∈ E for 1 ≤ i ≤ r. The length of the path is len(γ)=r. We say v1 is the start vertex and vr+1 is the end vertex of γ. If vi = vj for 1 ≤ i = j ≤ r + 1, we say the path is a simple path. If vr+1 = v1, we say γ is closed. If γ = (v1,

. . . , vr, v1) is a closed path with vi = vj for $1 \leq i = j \leq r$, we say $\gamma$ is a (simple) cycle or circuit. Alternatively, we can specify a path by a sequence of edges, rather than a sequence of vertices. Namely, a sequence of adjacent edges (e1, e2, . . . , er) defines a path of length r.

### B. Analysis Methodology

This analysis focuses on cybersecurity implications due to the presence of a suspect system component on an existing complex cyberphysical system. The analysis methodology presented analyzes information flow and relationships across the physical architecture of the UAS. The physical architecture is presented and elaborated to the level required to identify all suspect components of concern. Information interfaces, functional dependency, and communication flow are documented and captured in this process. This assessment considers the UAS as a System-of-Systems (SoS) which includes the flight vehicle system (FVS) element, a ground control station (GCS) element, project team element, and operational procedure elements.

To accomplish these goals, this analysis utilizes and extends methods presented in the cybersecurity literature, particularly methods for network dependency graph analysis. In this configuration, the flight vehicle system is single-string and all onboard components are flight critical, so there is no additional risk presented to the SoS by suspect components through potential induced failure to other components. Therefore, this analysis focuses on cybersecurity vulnerability through potential information exchange with external systems. This includes implications to confidentiality, integrity, and availability of the SoS.

Communication over a directed network dependency graph model occurs between two component blocks (nodes) in the graph. One necessary condition is there must exist two cooperative nodes – a source node and a destination node - to exchange information in the graph. A second necessary condition for communication is directed-graph reachability. The destination node must be reachable from the source node through the network. Particularly, this implies the existence of one or more directed paths between a source node and a destination node.

The general approach for this analysis is as follows.

a. Analyze the physical architecture of the system to identify the suspect components and all cyberphysical interfaces in the SoS.

b. Define the ***physical interface***, and assess vulnerabilities through the following:

    a. Analyze communication across the physical interface

        i. Designed communication

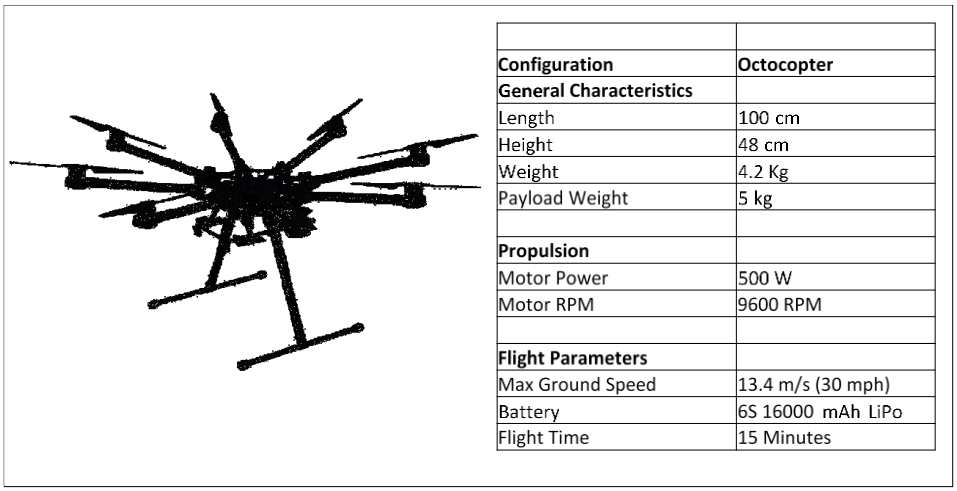
        ii. Hidden communication

    b. Analyze communication external to the physical interface (for example, through hidden communication paths)

In this process, components that cannot be affected by the suspect component are eliminated from the analysis.

### C. UAS Vehicle Specification and Model

The NASA Ames UAS flight vehicle analyzed is a custom configuration of third-party components from various third-party vendors, integrated on an octocopter airframe. This vehicle was designed to share a common architecture with other UAS vehicles in operation at ARC. The general specifications for the vehicle is shown in Figure 11.



| | |
|---|---|
| **Configuration** | **Octocopter** |
| **General Characteristics** | |
| Length | 100 cm |
| Height | 48 cm |
| Weight | 4.2 Kg |
| Payload Weight | 5 kg |
| | |
| **Propulsion** | |
| Motor Power | 500 W |
| Motor RPM | 9600 RPM |
| | |
| **Flight Parameters** | |
| Max Ground Speed | 13.4 m/s (30 mph) |
| Battery | 6S 16000 mAh LiPo |
| Flight Time | 15 Minutes |

**Figure 11. NASA UAS Vehicle Specifications**

The high-level analysis elements of the SoS architecture being considered are shown in Figure 12.

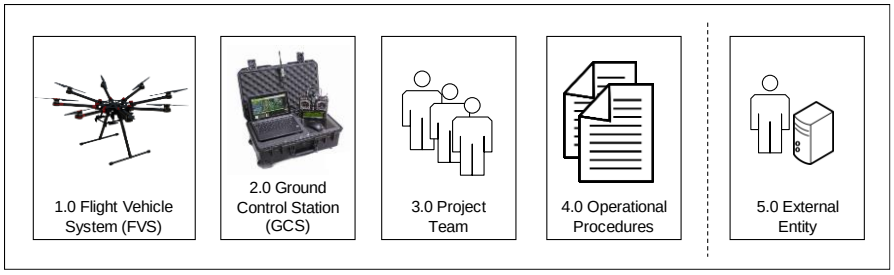



**Figure 12. System Model Level 1 Components.**

The Flight Vehicle System (FVS) is decomposed into the elements shown in Figure 13. The colored boxes indicate the FVS elements with suspect components and level of security concern. In this analysis, we define the airframe element to be composed of the mechanical structure of the vehicle platform only, specifically excluding any electronic components in this element definition. The airframe element provides mechanical support functions for the other FVS elements only, and does not provide any function that would pose an IT security threat (such as electrical, power, or communications). The motor and electronic speed controller (ESC) assembly element, however, does have potential concerns. The onboard avionics element is not comprised of any suspect components. However, this element does interface with the motor element (electrical, communications, power), and must be evaluated.

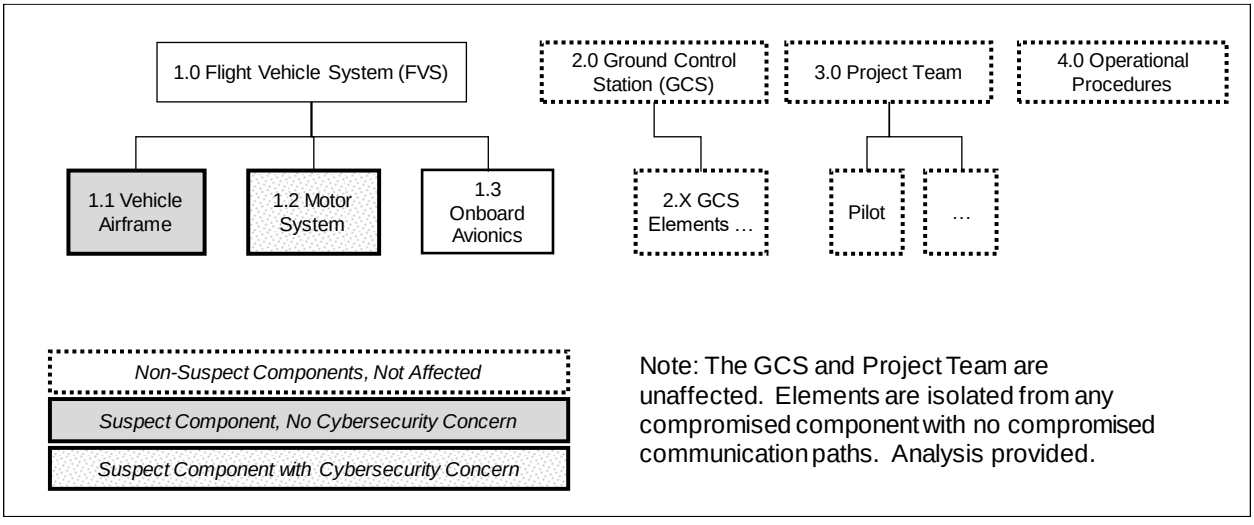


**Figure 13. System Model Decomposition of the FVS.** *Qualitative level of concerns are highlighted. The elements containing components with no cybersecurity concerns are highlighted as shown. The elements containing components with cybersecurity concerns highlighted as shown.*

Further decomposition of the GCS, project team, or operational procedures element will not be necessary for this vehicle configuration, as shown in Figure 14. The interaction between the suspect components and the GCS will be highlighted in the architecture description.

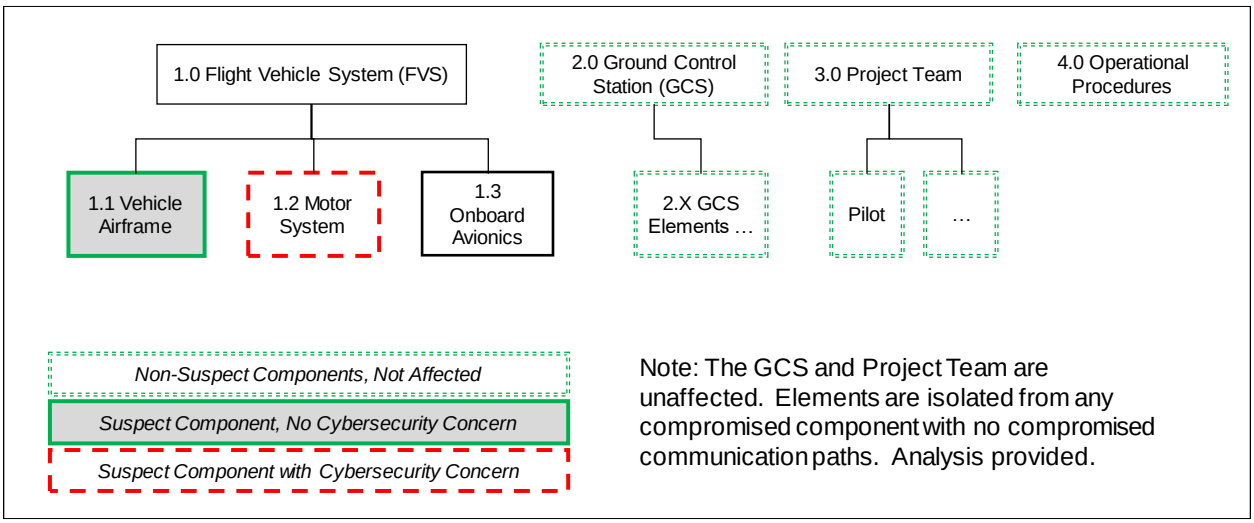



**Figure 14. Isolation of GCS, Project Team, and Operational Procedure Elements.**

The motor and ESC assembly is physically shown in Figure 15, in relation to the fully assembled vehicle system. One motor and ESC assembly is integrated into each one of eight motor arms. The interface with the avionics, as indicated, is comprised of two connectors: a two-power signal connector, and a two-wire power connector. The motor assembly is integrated into the arm and is not easily replaced without significant re-engineering. The airframe manufacturer provides the propeller arm as a single device, and does not provide the motor or ESC electronics as separate components (Figure 16).

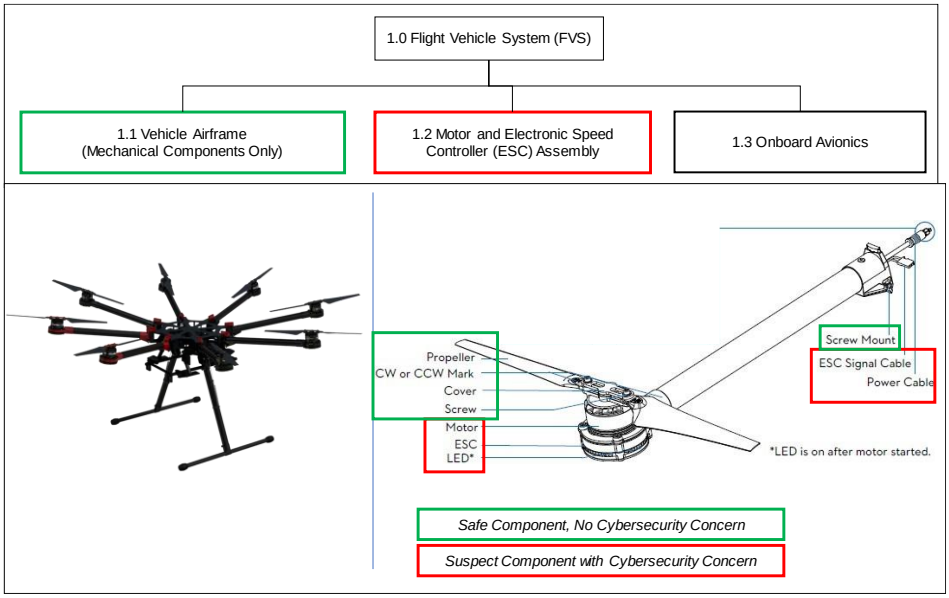



**Figure 15. Motor and ESC Assembly Element.**

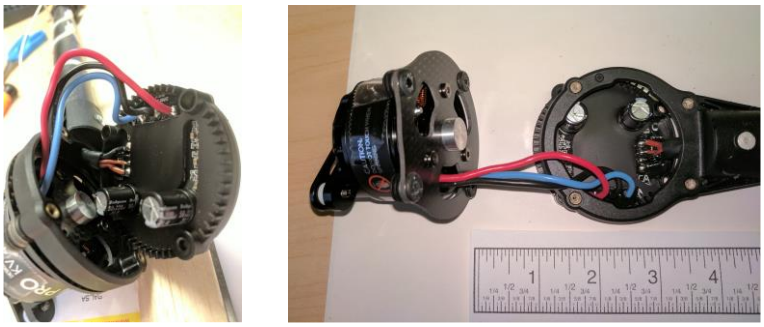


**Figure 16. Motor System Disassembly**

The electrical, power, and communications architecture utilized in the avionics element is shown in Figure 17. The specific suspect components have been highlighted, as are elements with programmable computational units (CPU). The Flight Control System (FCS) provides functions such as flight control logic and communication, and sends signals to the motor controllers based on onboard sensors integrated within the unit.

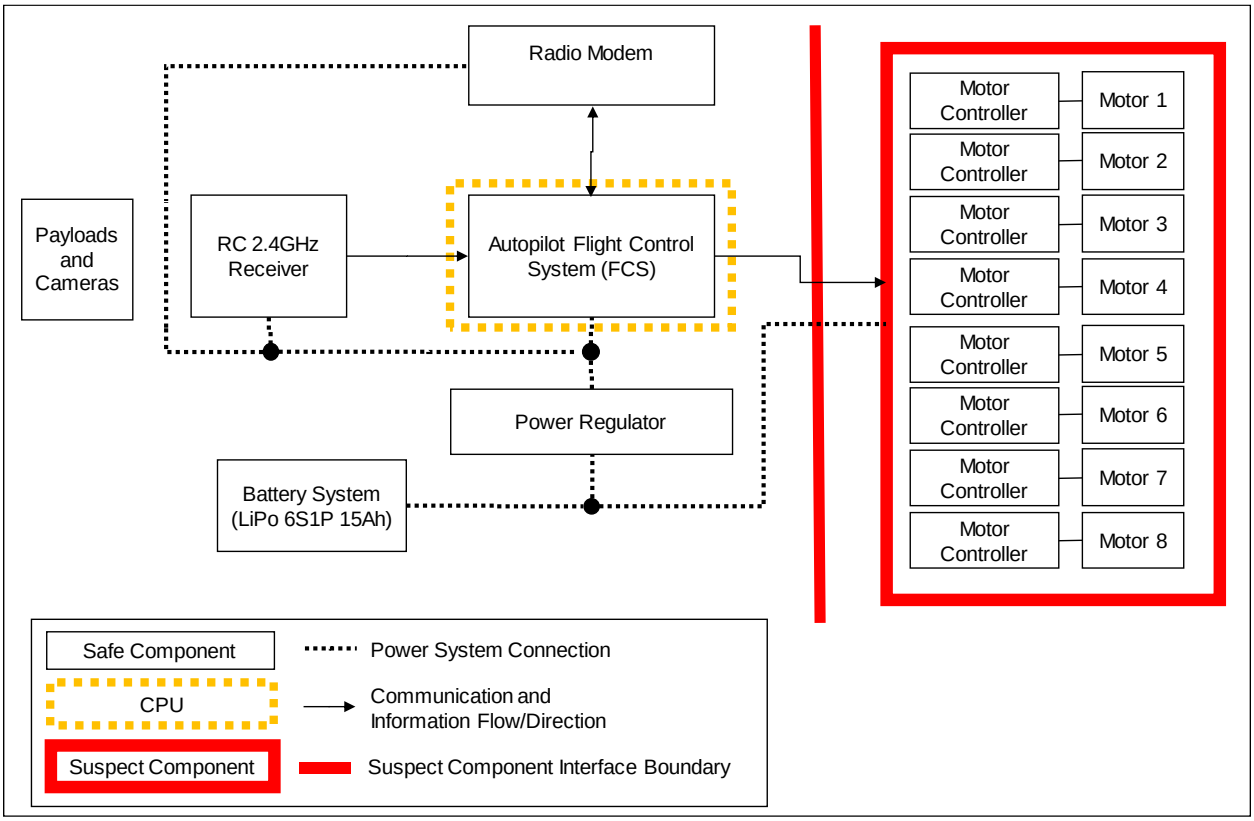



**Figure 17. Electrical, Power, and Communications Architecture for the Avionics Element and Motor ESC Element.**

The motor system assembly is a simple electronic device. The motor system assembly receives the electrical analog 'throttle' control signal from the FCS and power from an external power system. The motor system assembly must translate the control signals into a mechanical torque to drive the propellers based on this throttle command. This device has no access to any type of flight data or external data of any kind, other than this control signal. This is a one-way signal, with no communication back to the FCS or any other external system. There are no externally accessible ports to access the ESC microprocessor in this architecture and configuration.

Also shown in Figure 17 are the air-to-ground wireless communication elements. In this configuration, the ground station components are isolated from vulnerabilities.

## D. Security Concerns

The following items were identified as security concerns (SC). These concerns include possible threats, attack vectors, attack goals, and fall under the categories of Confidentiality, Integrity, and Availbility.

- SC1. Secret component communicating in a secret way.
- SC2. Hijacking by real-time communication.
- SC3. Hijacking through latent software/hardware (supply-chain risk).
- SC4. Unauthorized release of information to external entity. Includes vehicle telemetry, position, video streams, images from onboard cameras.
- SC5. Corruption of data.
- SC6. Introduction of failure into the system.

These concerns are categorized under IT security as shown in Figure 18.

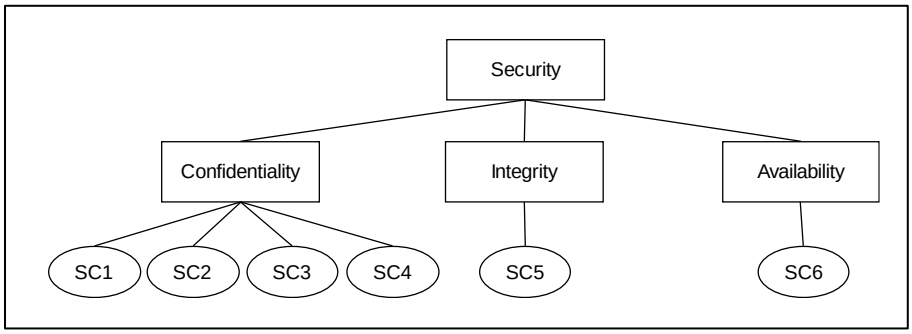



**Figure 18. Categorization of Security Threat**

## E. Interface Analysis

From the architectural analysis, the interface of interest for this analysis has been identified, as is illustrated in Figure 19. The interface of concern between the avionics and the motor system is comprised of two connectors and a total of four wires. The battery system element provides unregulated battery power from the Lithium-Polymer (LiPo) chemistry battery system, and is comprised of two wires: power and power system ground. The communication interface is an analog one-way electrical connection communicating over two wires: signal and signal ground. The signal commanded from the FCS to motor along the signal wire is an analog pulse coded modulation signal. This signal is a standard analog servomotor control signal. There are no other communication paths to other external systems in this architecture.

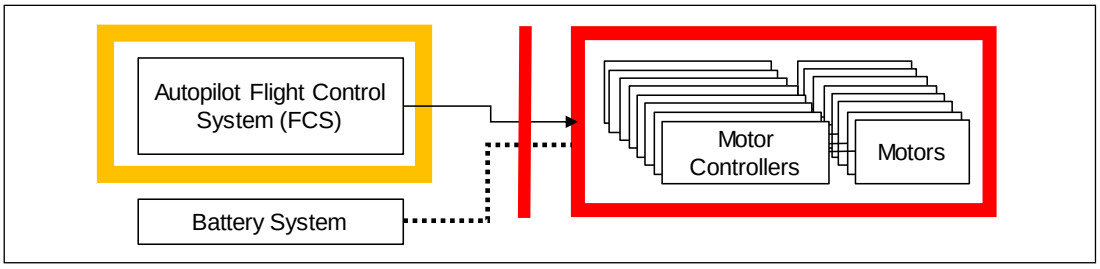



**Figure 19. Interface focusing on the suspect components in relation to the rest of the system.**

We analyze two general categories of attack on this system: attacks through the documented interface above, and attacks not-through or external-to this interface (i.e., such as through a hidden wireless modem embedded by the manufacturer).

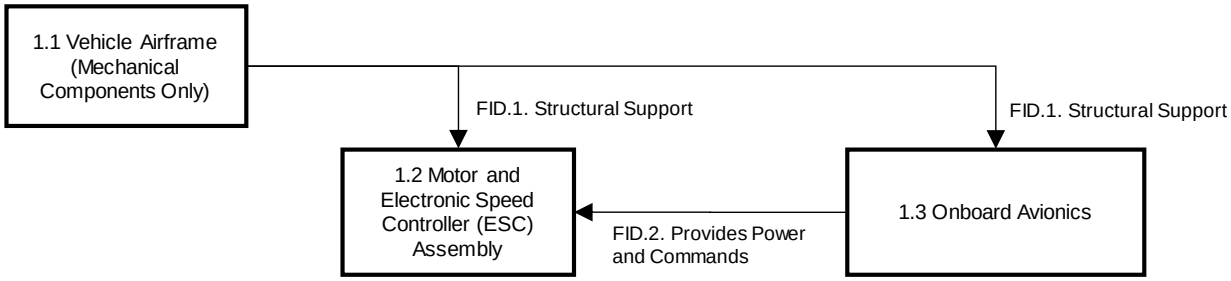
The first category of attack occurs through the documented interface. Two-way communication over analog signal or power lines is a common feature of these devices. Electronic speed controller (ESC) manufacturers commonly

provide provisions for programming ESCs from dedicated hardware programmers, often through embedding digital communication signals on the analog servo signal interface. To be exploited as a vulnerability in this specific architecture, however, it would require intentional and deliberate modification of the architecture. For instance, if an attacker gained access to the vehicle, the attacker could conceivably effect hardware modification to the FCS and attempt insertion of malicious code into the FCS software/firmware. However, this would require the attackers gain direct access to the UAS platform and the FCS, at which point the attackers already have access to the data being sent from the FCS. The motor controller and motor subsystem do not contain any useful information beyond the commands being sent. This would represent an unusually elaborate and sophisticated attack vector in which the attacker has gained no additional information on the already compromised system. Further, any vector for an attack through the interface requires the attacker to compromise the interface lines (power, communication), but these interface lines already contain all the information of value; there would be no further information to be gained from the motor controllers. Therefore, cybersecurity vulnerability that category of attack poses to the system is nearly non-existent.

The second category of attack occurs external to this documented interface. A vector for external interface attacks, for instance, through an undocumented wireless antenna/transmitter, would also be unlikely. The electronics board in these components are very small and the board does not contain any indication of capability beyond its expected function. The electronics are integrated into a small, lightweight board adjacent to the motors. Long range communication for electronics of this size would not be practical over long distances. Further, the motor and motor controller boards are major sources of RF noise and interface in flight, often interfering with primary air-ground communication from air-ground communication hardware placed at much greater distances and requiring much greater size/weight/power to send and receive signals. An embedded long-range device is not likely, especially one that could be operate in flight. A short-range communication method would be more technically feasible but equally unlikely, for instance, seeking to communicate with an external server through any open/vulnerable Wi-Fi network found while the vehicle was powered in the lab. As mentioned, the only information available to this subsystem is commanded throttle commands and bus voltage, which has little to no practical value to an attacker.

From this analysis, we find the overall cybersecurity vulnerability that the suspect motor controller electronics pose to this particular UAS system architecture is very low. This analysis finds the cybersecurity vulnerability for this particular vehicle configuration is very low. In a comparative analysis, the vulnerability is the same as other currently approved and currently operating UAS manufactured by other operators.

### F. Functional Interface Analysis




| FID | Source PBS ID | Destination PBS ID | Description | Information Flow across Interface (Implicit/Explicit) |
|---|---|---|---|---|
| FID.1 | 1.1 | 1.2 and 1.3 | Provide structural support for PBS 1.0 and sub-elements. Provides rigid attachment to the vehicle airframe. | Implicit: (1) Vehicle motion; (2) Structural motion. |
| FID.2 | 1.3 | 1.2 | Provide power and communications. | (Analyzed below.) |

**Figure 20. Functional Interface Analysis**

## G. Physical Interface Analysis

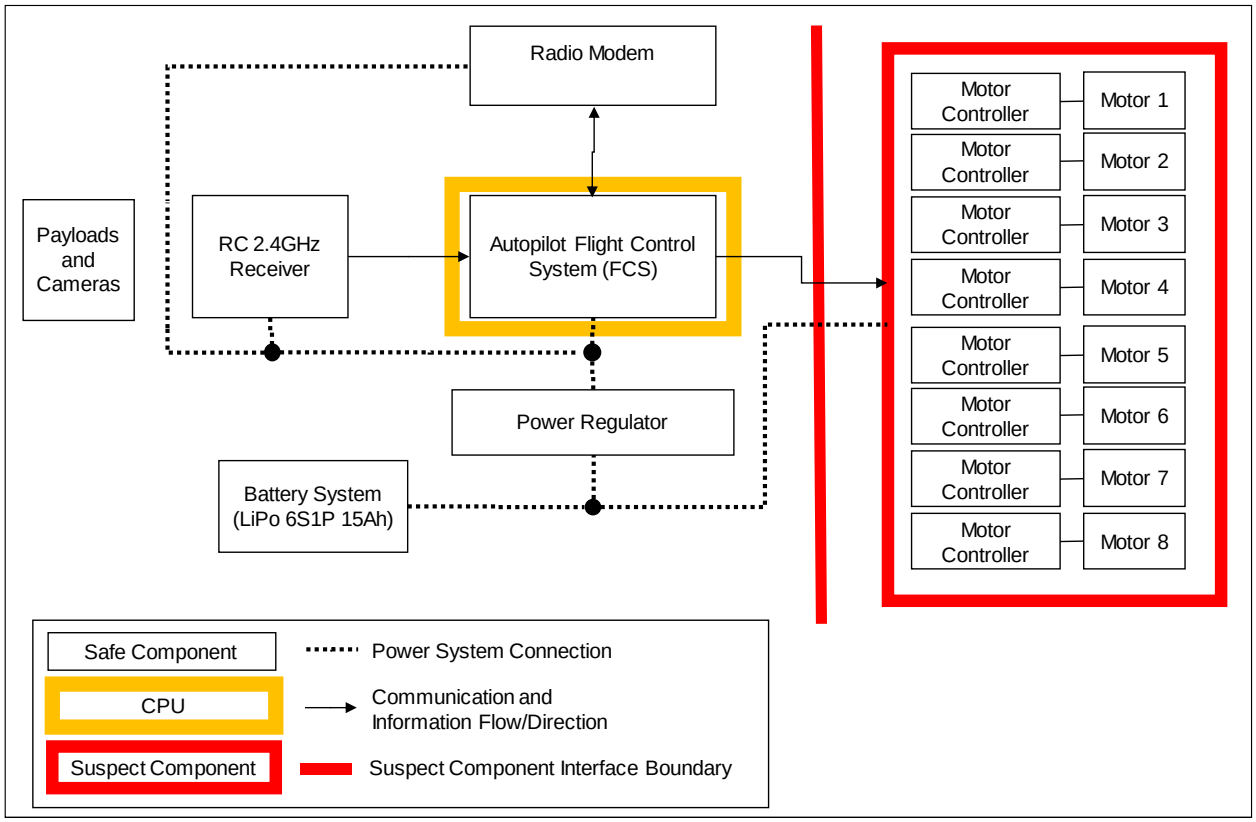



**Figure 21. Physical Architecture Interface Analysis**

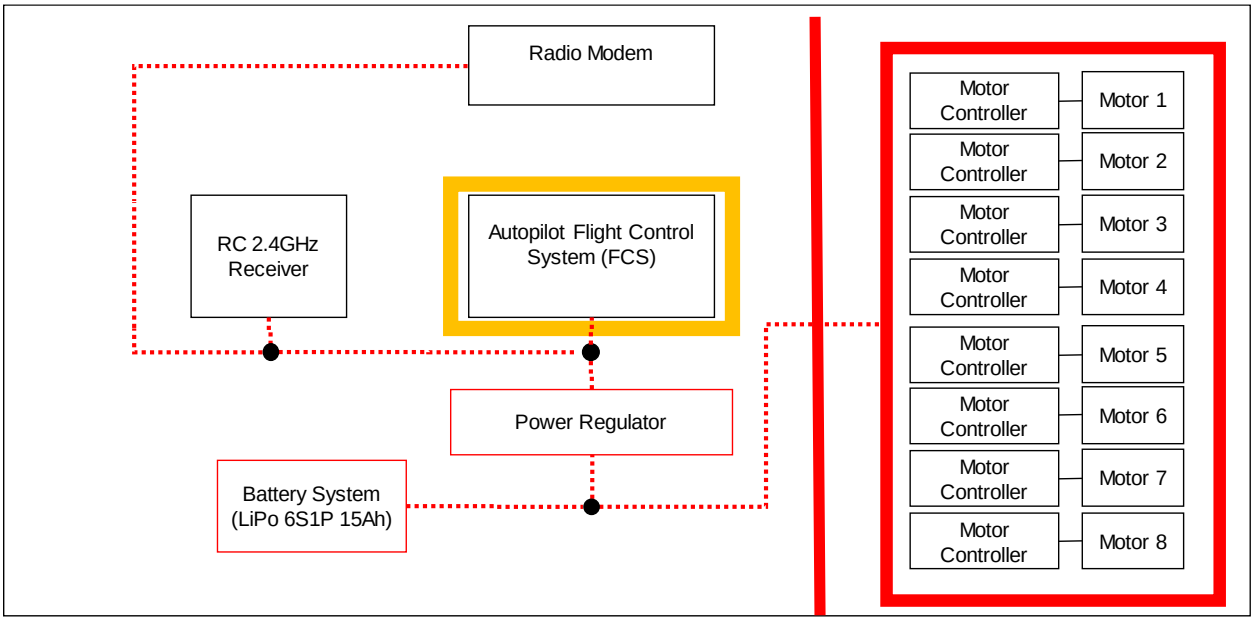



**Figure 22. Electrical Power System Analysis**

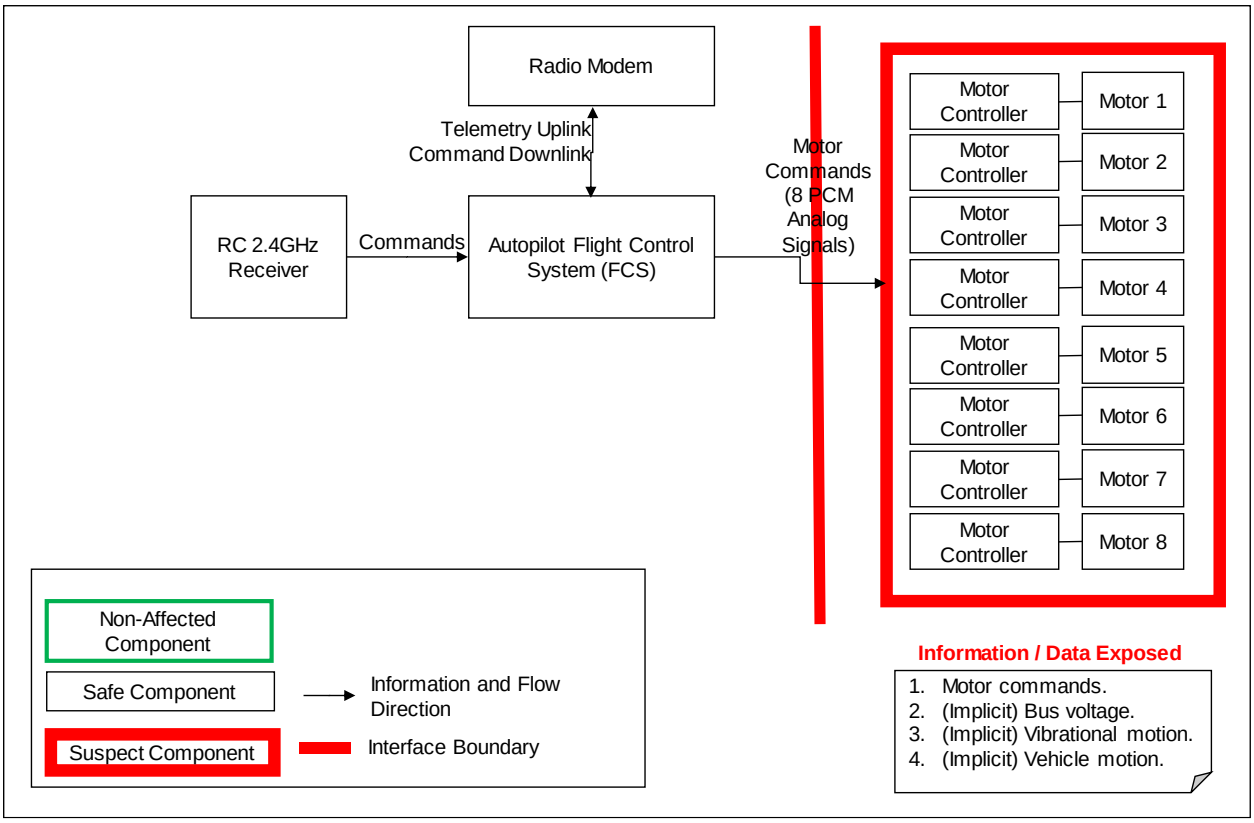



**Figure 23. Information Flow, Interface Definition, and Data Identification (Data Flow Diagram)**

The model of one-way information flow over the directed graph was verified by analyzing the board-level components that interface the FCS with the motor controller hardware (Figure 24).

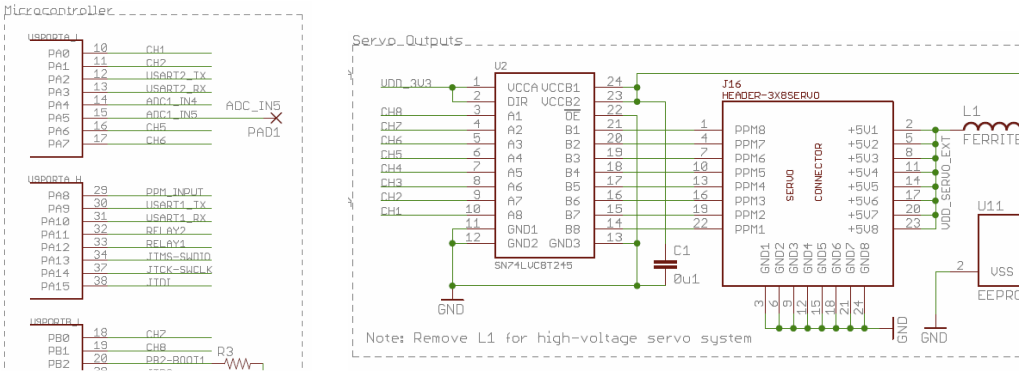



**Figure 24. Verification of One-Way Communication from the FCS**

From the network diagram, we construct a dependency network diagram to analyze the vulnerability dependency between components (Figure 25).

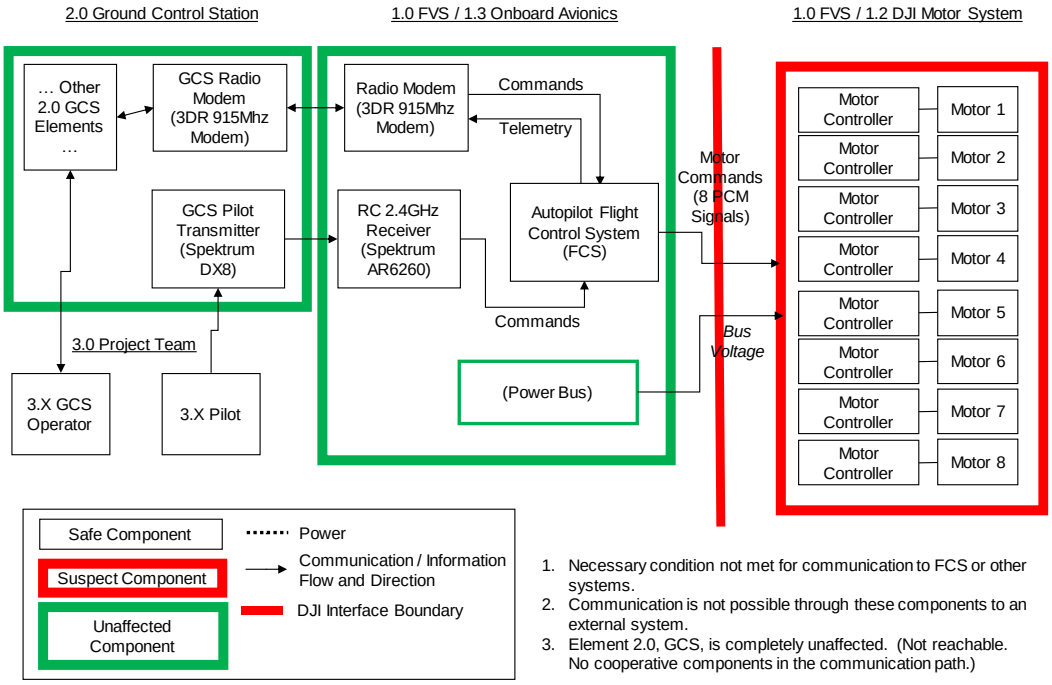



**Figure 25. Dependency Network Analysis**

The final category of risk is the potential for communication over unmodeled interfaces, to address anything that wasn't explicitly enumerated (Figure 26).

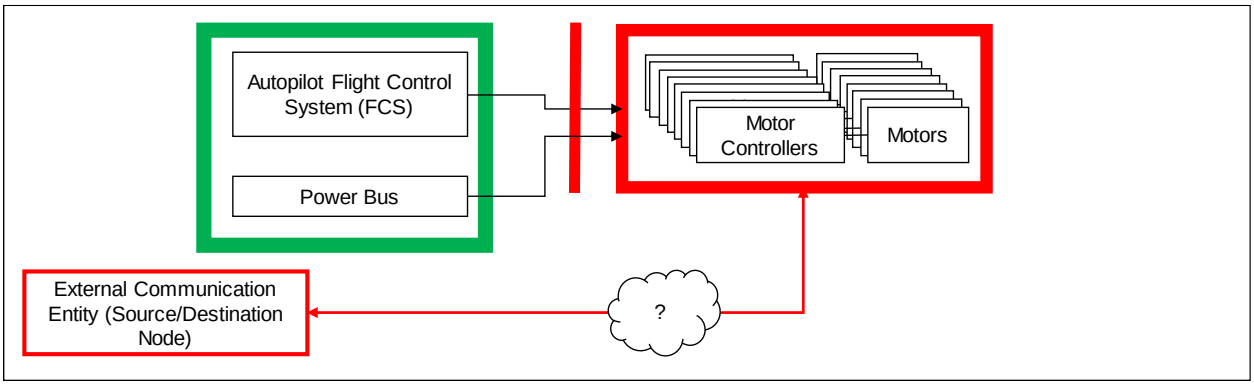



**Figure 26. Threat Analysis of Unmodeled or Hidden Communication**

An attack tree approach was utilized to model this attack. The developed attack tree shows the most likely method is using short-range radio-frequency (RF) wireless communication (WC), such as Wi-Fi. RF communication is unlikely during flight due to motors proximity and interference. Less-likely alternatives include malicious cameras installed at a flight test site watching LED's for covert optically encoded signals. As summarized in the attack tree (Figure 27), the assessment factors for hidden-communication vulnerability (such as Wi-Fi) requires a hidden modem, antenna, CPU, and memory storage device. This further requires a vulnerable Wi-Fi access point with internet connectivity (such as an open network). This risk tree analysis finds it unlikely that a such hardware could be hidden in the motor controller, and even less likely there are additional flight sensors or monitoring sensors hidden in the motor controller. The probability of this vulnerability being exploited is very low. The resulting severity of risk is very low, given the limited value of the data available if compromised.

Two methods were utilized to define the likelihood of exposed data. The first was an attack-tree method, shown in Figure 27.

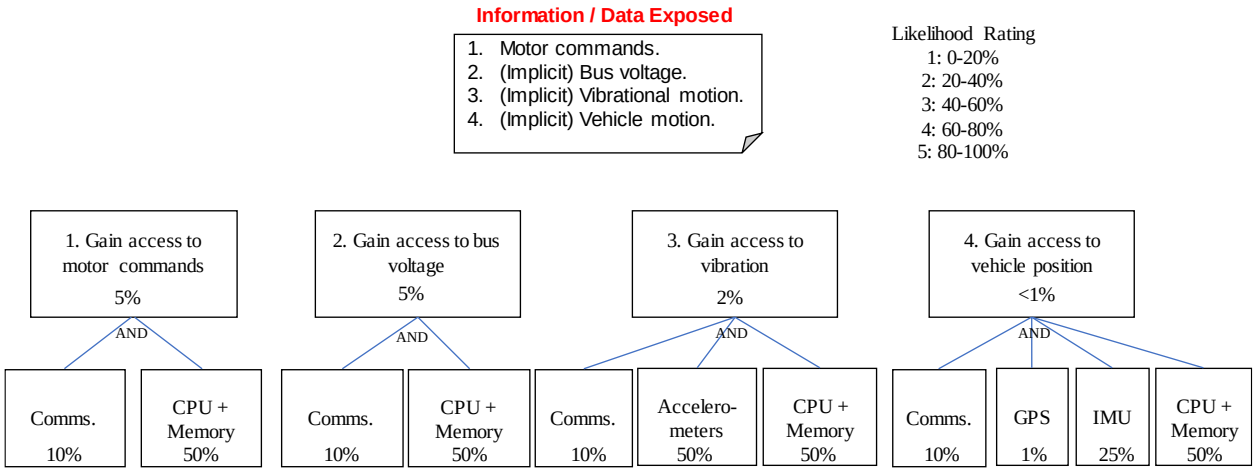



**Figure 27. Risk Assessment of Exposed Data Threat**

The risk elaborated from this method was compared to a risk evaluation grid (Figure 28), from the definitions from (Javaid 2015).

| ID | Exposed Data | Difficulty | Motivation (1-5) | Likelihood Rank (1-5) | Value of Data | Mission Impact | Consequence Rank (1-5) |
|---|---|---|---|---|---|---|---|
| 1 | Motor commands | High - 5 | 1 | 1 | 1 | 1 | 1 |
| 2 | Bus voltage | High - 5 | 1 | 1 | 1 | 1 | 1 |
| 3 | Vibrational motion | High - 5 | 1 | 1 | 1 | 1 | 1 |
| 4 | Vehicle motion | High – 5 (Infeasible) | 1 | 1 | 1 | 1 | 1 |

**Figure 28. Alternative Risk Assessment - Exposed Data Threat**

The risk analysis for the cybersecurity vulnerabilities was conducted in compliance with NASA NPR 7900, extending the analysis definitions per NASA APR 8000.4 (Figure 29). The risk definitions were extended to include cybersecurity risk severity levels as follows.

- Cyber Severity 1: Cyber Security procedural requirement violation, no data breach, no penetration or release of data.
- Cyber Severity 2: Reportable Cyber Security requirement deviation, no data breach, no penetration or release of data.
- Cyber Severity 3: Serious Data Breach delaying project. Denial of Service attack.
- Cyber Severity 4: Critical Data Breach causing High Visibly Event resulting in Project stoppage and delay, or firewall penetration, or data release, or other Critical (4) hazard.
- Cyber Severity 5: Catastrophic Data Breach, or takeover, or penetration resulting in High Visibly Event, resulting in termination of program or other Catastrophic (5) hazard.

| ID | Exposed Data | Likelihood of Data Release | Consequence of Unintended Data Release |
|---|---|---|---|
| 1 | Motor commands | 1 | 1 |
| 2 | Bus voltage | 1 | 1 |
| 3 | Vibrational motion | 1 | 1 |
| 4 | Vehicle motion | 1 | 2 |

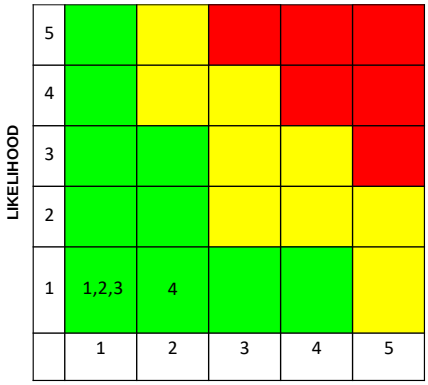



**Figure 29. Risk Assessment – Exposed Data Threat**

The risk of communication over the interfaces is summarized below (Figure 30).

| ID | Communication Threat | Likelihood | Consequence / Severity | Risk Rating |
|---|---|---|---|---|
| 1 | Vulnerability through communication across established interfaces. | 1 : Lowest (0 : Improbable) | 1 : Lowest | |
| 2 | Vulnerability through communication across unknown/hidden wireless communication. | 1 : Lowest (0 : Improbable) | 1 : Lowest | |
| 3 | Vulnerability through unknown/hidden Wi-Fi system (not in flight). | 1 : Lowest | 2 | |

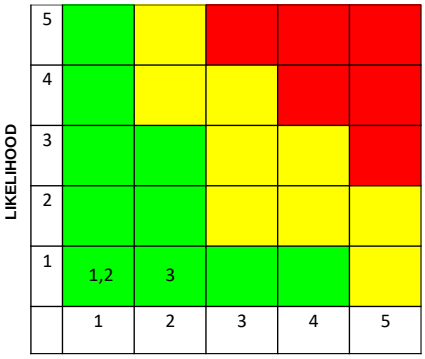



**Figure 30. Risk of Communication over Interfaces**

A summary of the risk assessment is shown below (Figure 31).

| ID | Communication Threat | Likelihood | Consequence / Severity | Risk Rating |
|---|---|---|---|---|
| SC1 | Secret component communicating in a secret way | 1 : Lowest | 2 | |
| SC2 | Hijacking by real-time communication | 1 : Lowest | 4 | |
| SC3 | Hijacking through latent software/hardware | 1 : Lowest | 4 | |
| SC4 | Unauthorized release of information to external entity. | 1 : Lowest | 2 | |
| SC5 | Corruption of data | 1 : Lowest | 3 | |
| SC6 | Introduction of failure in the system | 1 : Lowest | 4 | |

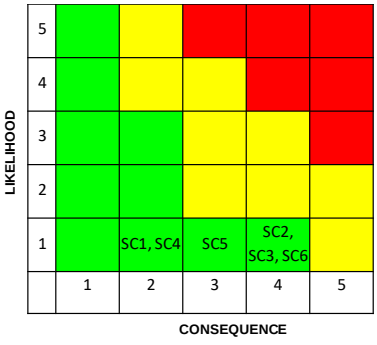



**Figure 31. Summary Risk Assessment of Cybersecurity Concerns**

## IV. Conclusion

As Unmanned Aerial Systems (UAS) continue to advance in terms of technology, there must be recognition by everyone involved that UAS are increasingly becoming complex cyberphysical systems. Cybersecurity is a known risk addressed by NASA's information technology security plan. UAS, as with all cyberphysical systems, should continue to be managed according to the established requirements. Security compliance and oversight must continue to be enforced as with all complex information technology systems.

The analysis provided in this report is a general model-based approach for cybersecurity risk assessment on an existing system where specific components have been identified as being suspect. A summary of the analysis approach, findings, and conclusions is as follows. The analysis approach assesses cybersecurity vulnerabilities through interface analysis of suspect components with the rest of the system. The only suspect component utilized in this system that posed a potential cybersecurity threat in this situation was narrowed to a motor assembly, which includes an electronic speed controller (ESC) and a brushless DC motor, and which operates in a standard manner consistent with other manufacturer ESC and motor systems. The interface-based analysis finds the suspect component only has access to the provided analog command signal and power, with no access to any other external system data. The analysis finds there is little to no possibility the suspect component can send any type of communication signal to any external system. In the resulting assessment, the motor assembly posed little to no cybersecurity threat or vulnerability. There is a very low likelihood of vulnerability, and a very low overall cybersecurity threat from either motor systems.

The implication for this analysis methodology is that any change in the architecture should be analyzed for IT security issues on a case by case basis. This analysis does not automatically apply to any future configuration change effected on this vehicle. Configuration changes, including introduction of new payloads, could invalidate this analysis or introduce new vulnerabilities. For instance, the presented configuration of this NASA small UAS vehicle is substantially different from the initial preliminary configuration conceived at the time of procurement. Vulnerability assessments may change between approved vehicle configuration revisions on the same vehicle. Consideration of vulnerabilities and awareness of cybersecurity must continue throughout the vehicle life cycle, and be reanalyzed as new vulnerabilities are identified.

The methodology presented here is a small step towards establishing a comprehensive suite of formalized methodologies to address UAS cybersecurity. This particular methodology is narrowly focused on existing system of system architecture that contains suspect components, as arises when new vulnerabilities are identified during the operational phase of life in the aircraft lifecycle. Additional application to more complex vehicles and problems would be needed to expand this approach to a larger class of problems and to validate this approach within a larger cybersecurity and IT security framework.

## Acknowledgments

The authors would like to thank our collaborators and colleagues in the NASA UAS Traffic Management (UTM) project, the NASA SAFE50 project, and the Aircraft Management Office (AMO) at NASA Ames Research Center.

## References

[1] NASA NPR 7900.3D. NASA Procedural Requirements: Aircraft Operations Management Requirements. Effective 1 May 2017. Available at https://nodis3.gsfc.nasa.gov/npg_img/N_PR_7900_003D_/N_PR_7900_003D_.pdf. [retrieved 1 Dec 2018]

[2] NASA APR 8000.4. NASA Ames Research Center Procedural Requirements: Risk Management Process Requirements. Effective 27 Jan 2011. Available at https://cdms.nasa.gov/assets/docs/centers/ARC/Dirs/APR/APR8000.4.html. [retrieved 1 Dec 2018]

[3] Cabler, S. J. M. "FAA UAS Symposium: Cybersecurity and Mitigations". 2017 FAA UAS Symposium. Available at https://www.faa.gov/uas/resources/event_archive/2017_uas_symposium/media/Workshop_2_Cybersecurity.pdf.

[4] Unmanned Aircraft Systems (UAS) Traffic Management (UTM) Concept of Operations. Office of NextGen, Federal Aviation Administration. Washington, DC. May 18, 2018

[5] Kopardekar, P., Rios, J., Prevot, T., Johnson, M., Jung, J., & Robinson, J. Unmanned Aircraft System Traffic Management (UTM) Concept of Operations. 2016 AIAA Aviation Forum. June 2016.

[6] Sampigethaya, K., Kopardekar, P., & Davis, J. Cyber security of unmanned aircraft system traffic management (UTM). In 2018 Integrated Communications, Navigation, Surveillance Conference (ICNS) (pp. 1C1-1). IEEE. April, 2018.

[7] Kerczewski, R. J., Apaza, R. D., Downey, A. N., Wang, J., & Matheou, K. J. Assessing C2 communications for UAS traffic management. In 2018 Integrated Communications, Navigation, Surveillance Conference (ICNS) (pp. 2D3-1). IEEE. April, 2018.

[8] Shoufan, Abdulhadi; Hassan AlNoon; Joonsang Baek. "Secure Communication in Civil Drones." In International Conference on Information Systems Security and Privacy, pp. 177-195. Springer, Cham, 2015

[9] Mansfield, Katrina; Timothy Eveleigh; Thomas H. Holzer; Shahryar Sarkani. "Unmanned Aerial Vehicle Smart Device Ground Control Station Cyber Security Threat Model." In Technologies for Homeland Security (HST), 2013 IEEE International Conference on, pp. 722-728. IEEE, 2013.

[10] Mansfield, Katrina, Timothy Eveleigh, Thomas H. Holzer, and Shahryar Sarkani. DoD Comprehensive Military Unmanned Aerial Vehicle Smart Device Ground Control Station Threat Model. Defense Acquisition Research Journal (ARJ), April 2015, Vol. 22, No. 2: 240-273. Available at http://www.dtic.mil/get-tr-doc/pdf?AD=ADA622358. [retrieved 1 Dec 2018]

[11] Blackhurt, R. "The air force men who fly drones in Afghanistan by remote control". The Telegraph. Published 24 Sep 2012. Available at https://www.telegraph.co.uk/news/uknews/defence/9552547/The-air-force-men-who-fly-drones-in-Afghanistan-by-remote-control.html. [retrieved 1 Dec 2018]

[12] NIST. "Standards for Security Categorization of Federal Information and Information Systems.", FIPS PUB 199. Federal Information Processing Standards Publication (FIPS), National Institute of Standards and Technology (NIST), U.S. Department of Commerce.

[13] Gorman, S., Dreazen, Y. R., and Cole, A. "Insurgents Hack U.S. Drones". Published 17 Dec 2009. Wall Street Journal. Available at https://www.wsj.com/articles/SB126102247889095011 [retrieved 1 Dec 2018].

[14] Faughnan, M. S., Hourican, B. J., Macdonald, G. C., Srivastava, M., Wright, J. A., Students, G., White, J. C. (2013). Risk Analysis of Unmanned Aerial Vehicle Hijacking and Methods of its Detection. 2013 IEEE Systems and Information Engineering Design Symposium, University of Virginia, Charlottesville, VA, USA. April 26, 2013.

[15] Nguyen, T. C. "Virus attacks military drones, exposes vulnerabilities." ZDNet, Published 11 Oct 2011. Available https://www.zdnet.com/article/virus-attacks-military-drones-exposes-vulnerabilities/ [retrieved 1 Dec 2018].

[16] Paganini, P. "Hacking Drones ... Overview of the Main Threats." Published 4 June 2013, Infosec Institute. Available http://resources.infosecinstitute.com/hacking-drones-overview-of-the- main-threats/ [retrieved 1 Dec 2018].

[17] Shepard, D. P., Bhatti, J. A., Humphreys, T. E., & Fansler, A. A. (2012). Evaluation of smart grid and civilian UAV vulnerability to GPS spoofing attacks. In Radionavigation Laboratory Conference Proceedings 2012. http://hdl.handle.net/2152/63231.

[18] Goppert, J., Liu, W., Shull, A., Sciandra, V., Aldridge, H., Electronics, S., & States, U. (2014). Numerical Analysis of Cyberattacks on Unmanned Aerial Systems. Terra, 11(5), 1-17. https://doi.org/10.2514/1.I010114

[19] Fritz, J. (2013). Satellite hacking: A guide for the perplexed. Culture Mandala: The Bulletin of the Centre for East-West Cultural and Economic Studies, 10(1), 3.

[20] DHS Risk Lexicon. Department of Homeland Security (DHS) Risk Steering Committee. September 2008. Available at http://www.dhs.gov/xlibrary/assets/dhs_risk_lexicon.pdf [retrieved 1 Dec 2018].

[21] Cebula, J and Young, L. A Taxonomy of Operational Cyber Security Risks. Software Engineering Institute, Carnegie Mellon University. CMU/SEI-2010-TN-028. 2010. Available at https://www.sei.cmu.edu/reports/10tn028.pdf [retrieved 1 Dec 2018].

[22] Javaid, A. Y., Sun, W., Devabhaktuni, V. K., & Alam, M. (2012). Cyber security threat analysis and modeling of an unmanned aerial vehicle system. 2012 IEEE Conference on Technologies for Homeland Security (HST), 585-590. https://doi.org/10.1109/THS.2012.6459914

[23] ETSI TS 102 165-1. Telecommunications and Internet Protocol Harmonization over Networks (TIPHON) Release 4; Protocol Framework Definition; Methods and Protocols for Security; Part 1: Threat analysis. ETSI Technical Specification TS 102 165-1 V4.1.1, 2003.

[24] Torr, Peter. "Demystifying the threat modeling process." IEEE Security & Privacy 3, no. 5 (2005): 66-70.

[25] Alghamdi, A. S., Hussain, T., & Faraz Khan, G. (2010, March). Enhancing C4I security using threat modeling. Proceedings of 12th International IEEE Conference on Computer Modelling and Simulation (UKSim) (pp. 131–136), Cambridge, UK, March 24-26.

[26] Clark, John A., John Murdoch, John A. McDermid, Sevil Sen, H. Chivers, Olwen Worthington, and Pankaj Rohatgi. "Threat modelling for mobile ad hoc and sensor networks." In Annual Conference of ITA, pp. 25-27. 2007.

[27] Di, J., & Smith, S. (2007). A hardware threat modeling concept for trustable integrated circuits. 2007 IEEE Region 5 Technical Conference, TPS, 65–68. https://doi.org/10.1109/TPSD.2007.4380353

[28] Stango, A., Prasad, N. R., & Kyriazanos, D. M. (2009). A threat analysis methodology for security evaluation and enhancement planning. Proceedings of IEEE Third International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2009) (pp. 262–267), Athens, Greece, June 18-23.

[29] Abi-Antoun, Marwan, Daniel Wang, and Peter Torr. "Checking threat modeling data flow diagrams for implementation conformance and security." In Proceedings of the 22nd IEEE/ACM International Conference on Automated Software Engineering, pp. 393-396. 2007.

[30] Swiderski and W. Snyder. Threat Modeling. Microsoft Press, 2004.

[31] Oladimeji, Ebenezer A., Sam Supakkul, and Lawrence Chung. "Security threat modeling and analysis: A goal-oriented approach." Proc. of the 10th IASTED International Conference on Software Engineering and Applications (SEA 2006). 2006.

[32] B. Schneier. Attack trees: Modeling Security Threats. Dr. Dobbs Journal, December 1999, pp 21-29.

[33] P. Moore, R. J. Ellison, and R. C. Linger. Attack mod- eling for information security and survivability. Technical Report CMU/SEI-2001-TN-001, Software Engineering Institute, Carnegie Mellon University, March 2001

[34] Saini, Vinee; Qiang Duan; Vamsi Paruchuri. "Threat modeling using attack trees." Journal of Computing Sciences in Colleges 23, no. 4 (2008): 124-131

[35] Amenaza Technologies SecureITree. Available at http://www.amenaza.com/.

[36] Van Lamsweerde, A. Goal-oriented Requirements Engineering: A Roundtrip from Research to Practice, invited keynote paper. In Proc. of the 12th IEEE Joint Inter'l Requirements Engineering Conference (RE'04), pages 4-8, Kyoto, Sept. 2004.

[37] Guariniello, C., & DeLaurentis, D. (2014). Communications, information, and cyber security in systems-of-systems: Assessing the impact of attacks through interdependency analysis. Procedia Computer Science, 28(Cser), 720-727. https://doi.org/10.1016/j.procs.2014.03.086

[38] Sanjab, Anibal; Saad, Walid; and Başar, Tamer. "Prospect Theory for Enhanced Cyber-Physical Security of Drone Delivery Systems: A Network Interdiction Game." arXiv preprint arXiv:1702.04240 (2017).

[39] C. Guariniello and D. DeLaurentis, "Dependency Analysis of System-of-Systems Operational and Development Networks", 2013 Conference on Systems Engineering Research, Procedia Computer Science, Vol. 16, 2013, pp. 265-274.