DISSERTATION

ANALYSIS OF A CYBERSECURITY ARCHITECTURE FOR SATELLITES USING MODEL-BASED SYSTEMS ENGINEERING (MBSE) APPROACHES

Submitted by

Daniel Johnson

Department of Systems Engineering

In fulfillment of the requirements

For the Degree of Doctor of Engineering

Colorado State University

Fort Collins, Colorado

Spring 2025

Doctoral Committee:

Advisor: Thomas Bradley

Heidi Poturalski
Jim Adams
Daniel Herber
Steve Reising

ABSTRACT

ANALYSIS OF A CYBERSECURITY ARCHITECTURE FOR SATELLITES USING MODEL-BASED SYSTEMS ENGINEERING APPROACHES

Historically, satellites have been relatively isolated from cybersecurity threats. However, during the 2020s, cyberattacks on critical ground-based infrastructure became more common and prevalent, and with the increase in technological advancement of peer adversaries, the United States government has come to recognize and define an increasing level of vulnerability in space-based assets as well. This doctoral research seeks to understand and address cybersecurity vulnerabilities inherent in commercial small-scale satellite architectures by demonstrating how model-based systems engineering (MBSE) can enable the design and analysis of a cyber-secure satellite architecture.

To determine the cybersecurity vulnerabilities applicable to satellites, a scholarly review of literature on cybersecurity threats and mitigation techniques was performed and applied to satellite systems. The result of this scholarly review is an assessment of the cybersecurity threats applicable to satellites with a particular focus on small satellite architectures, and an understanding of current cybersecurity threat agents and the categories of cyber threats applicable to such satellites. Common architectures and satellite components were analyzed to determine vulnerabilities that could be exploited.

The next phase of research then evaluated how industry has applied cybersecurity practices to satellite systems. We were able to determine the gaps which industry currently faces

and recommended a set of generic requirements that could help create a cyber-secure satellite from early in the program lifecycle.

The final phase of research synthesized the findings from the first two phases to build an MBSE model that integrates cybersecurity engineering and satellite architecture into a singular design process. We also analyzed the benefits to a company of applying the MBSE architectural process, paying particular attention to reusability of the model, cost, and human-centered benefits of committing to MBSE for multiple programs.

A finding of this research is that the cybersecurity vulnerabilities for satellites are due to two main factors. First, as technology has advanced and become more available, there is a changing threat landscape where satellites launch is more accessible, increasing the risk that threat actors can compromise unprotected satellites. Second, space technology has lagged behind terrestrial information and cyber technology in its ability to adapt and overcome cybersecurity threats, creating vulnerabilities in satellite architectures. Another revelation is the disconnect between traditional software engineers and their cyber engineer counterparts, leading to a lack of understanding of key cyber-vulnerabilities during the design process. This leads to a consequential need to build cyber-protections into the design process from program initialization. Finally, the cyber tools in use today are also disconnected from the other traditional architectural design tools, leading to our conclusion that all of the tools must be integrated together under an MBSE design process, furthering the evolution of systems engineering while also encouraging the industry to incorporate cybersecurity into satellite programs from the beginning.

Upon completion of this research project, the contributions are a scholarly review of the literature on cybersecurity threats and mitigation techniques in space and satellite systems, an evaluation of a set of cybersecurity requirements for satellite systems application, an MBSE

example case for a cyber-security embedded satellite system, and an evaluation of the costs and benefits of an MBSE-enabled architecting process as applied to an industrial satellite system architecting process. The combination of this research represents novel contributions to the state of the field by defining the cybersecurity vulnerabilities for Space Systems and exhibiting how MBSE can aid in a cyber-secure architecting process.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1.

Introduction and Background

1.1 The Growth of Cybersecurity Threats

The 21st century has been marked by a pivotal change in human economy and industry as humanity continues to shift from an industrial age to a digital age [1]. The ubiquity of digital systems has also changed the types of threats present to our society and economies [2]. Threats to security that were predominantly physical are now digital, having adapted to the digital realm, and are now generically known as "cybersecurity threats" [3]. Cybersecurity incidents resulting from these threats happen every day with varying degrees of impact [4]. Cybersecurity threats have occurred since the invention of networked computers [5], but as our lives become more intertwined with digital systems, these threats are becoming more prevalent and impactful to industry, economy, and personal safety. See **Appendix A** for a cybersecurity vulnerability case study on the Stuxnet and the Natanz Uranium Refining Centrifuges.

The United States industry and government has moved towards a "4th Industrial Revolution" where technology is becoming more and more connected under a construct of an Internet of Things (IOT) [6]. In IOT, many vital corporate systems have become highly interconnected to enable efficiency in their operations and data sharing with other companies. However, this has also created many access points and vulnerabilities for threat actors to compromise. Modern cyberattacks are impacting consumers through availability of physical products and services [7].

Cybersecurity threats in the past few years against the critical infrastructure of the United States highlight a gap in the cybersecurity protection of US critical space assets. 2021 was a year in which cyberattacks shifted from obscure to prominent [8]. For example, in May 2021 the Colonial Pipeline in the Southeastern US suffered a ransomware attack where foreign actors compromised the information technology system behind the oil pipeline and effectively shut off the flow of petroleum. This resulted in consumers panicking and hoarding gasoline which in turn increased the cost of gas and decreased its availability across the United States [9]. In May 2021, JBS Foods suffered a ransomware attack when foreign actors compromised the information technology system behind this company which supplies 20% of the meat to the world. This attack rendered United States meat facilities inoperable, which in turn increased the cost of meat and decreased its availability [10]. These incidents have forced a reevaluation of the extent to which the critical infrastructure of the United States is secured in the new digital age.

1.2 Cybersecurity Threats Are a Concern for Space Systems

One important area of critical infrastructure which is particularly vulnerable to a deliberate cyberattack is our space infrastructure [11]. The risk of a cybersecurity event is classically defined as the product of the impact of an event, and the probability of its occurring. In space systems, the impact of a cybersecurity event would be very high. Much of consumers' daily lives depend on satellites for communications, weather, science, navigation, and national security, to name only a few categories. The wide-ranging effects of space-based cyberattacks against satellites have similar economic implications to terrestrial attacks, including the possibilities of critical financial systems and infrastructure being infiltrated via satellite connection [12]. However, despite the critical implications of cyberattack, very few examples of cyberattack on space systems are present in the literature. As a result, there have been limited

public case studies [13] into the cyber exploitation of satellites because historically the academic community has not been involved in the detailed development and operation of satellites. The few publicly available case studies only describe how threat agents compromised the command-and-control interface to the satellite, and in all of these cases the operators were able to successfully recover the satellites [14] without long-term mission impact. Because of this lack of reporting, and because of the added expense associated with protecting systems up-front, satellite developers are beginning to make the same mistakes as terrestrial infrastructure companies such as pipelines and electrical companies did in their own infancy: They believe imprudently in the protective power of the "air gap" between Earth and Space, and they are therefore not protecting their developing space infrastructure, instead choosing to focus their efforts and expense on the protection of the ground systems.

An aging terrestrial infrastructure, not designed with the future of computing and network power in mind, grows increasingly vulnerable to cyberattacks. There is, however, an opportunity to avoid this pitfall for small-satellite space infrastructure, which is still in its infancy. If satellites and constellations are designed today with present-day cyberattack capabilities as well as protections against a prediction at what the near- and far-reaching future of cyberattacks might hold, then their operators and customers can mitigate future cyberattacks, their expense and interruption, and the expense of retro-fitting space-based systems with security features once attacks on satellites become more widespread.

Due to the specialized considerations of satellite operations, satellite hardware has traditionally been designed and built for selective applications and was only available to the specific government agencies which had paid for its development [15]. Much like the wartime secrecy surrounding designs, specialized parts design and development provided an exclusionary

form of cybersecurity protection which was sufficient for the time, since the knowledge of how to interact with and operate satellites was not well known outside space manufacturers and governments [16]. The Soviet Union launched the first satellite, Sputnik, on October 4, 1957, winning the first skirmish of the Cold War era Space Race. For the rest of the twentieth century, substantive access to space depended on cooperation with a government, for either the launch vehicle or the launch site and usually for both. Once a satellite launched, governments generally considered the vehicle unreachable – they would not be able to easily change it, fix it, or recover it, and often it would be cheaper to make a new satellite upon premature failure. Furthermore, in a time of icy international tensions, even commercial satellite designs were much more secretive so any attempt to access an in-space satellite would have been very difficult. As such, until the twenty-first century, government-based entities posed the only realistic threats to a satellite system, by either compromising the terrestrial ground station or using a missile to destroy a space-based satellite. In an era of developing cyber technologies, governments of the Cold War and the late twentieth/early twenty-first centuries were not interested in spending billions to co-opt each other's relatively primitive space-based satellites – it would have been much simpler, cheaper, and easier to destroy the other's satellites and so both the attack and defense avenues focused on protecting the ground station and on detecting and preventing destructive attacks.

As satellite technology advanced, however, industry started to sell the components that make up these satellites as commercial-off-the-shelf (COTS) components. The availability of COTS components to the public enables threat actors to purchase them for on-site evaluation. These hackers can learn how the COTS products work and practice hacking them at their own leisure, taking their time to determine the specific cybersecurity vulnerabilities for each COTS product which could be exploited once the parts are integrated into a satellite [17]. The

affordability of COTS components drives the industry to use them, which could embed vulnerabilities into a system in an effort to save money [18] [19]. Furthermore, as competition increases between companies who build satellites, the market price to build satellites falls, encouraging some companies to underbid the cost to properly protect the system in order to make their bids appear more favorable and thus win the contract.

Orbital cybersecurity is especially important as there is a rise in proliferated Low Earth Orbit (LEO) with small satellites, these orbits may provide easier access to this critical satellite infrastructure [20]. Until recently, only government entities could afford the cost to launch a rocket to space. With the concept of "rideshares" where many small satellite manufacturers can split the financial burden to share a single rocket launch, along with the rise of companies like SpaceX which has launched thousands of broadband satellites into LEO, space has become more accessible to everyone. This has changed the threat landscape in LEO by making it more congested and has enabled threat actors to gain access to unprotected satellites that lack modern cybersecurity best practices. Furthermore, remote-piloted drone technology has improved, with governments and even commercial companies mastering high-altitude and space-based remote and drone technologies, raising the "easily-accessed" space ceiling to include objects in LEO.

One of the biggest challenges in defending against satellite cybersecurity vulnerabilities is the remote and harsh environment of space. Terrestrial solutions to successful infrastructure hacks often involve shutting down the hacked system – if the hacker is preventing the owner from accessing their system, then the owner may attempt a shut down and reboot to try and kick out any malicious code while possibly relying on an ancillary backup that is built in to allow for main system shutdown and maintenance. However, in the frigid temperatures of space, rebooting the satellite will also turn off the heating component which keeps many of the other mission-

critical components alive, potentially for long enough that the satellite will not be able to come back online. Today, however, the risk/reward equation has shifted. As previously discussed, satellites are increasingly integral to the infrastructure of whole countries and economies. In the 1980s, taking over a communications satellite might have resulted in a slightly inconvenient loss of television service for a population. Today, much of consumers' daily lives depend on satellites for communication services and thus taking over the right satellites could trigger huge impacts as financial transactions would be unable to complete. The exploitation of vulnerabilities of satellites results in serious impact including loss of revenues, economic disruption, and compromised national security.

1.3 A Transformed Cybersecurity Outlook

The United States can no longer afford to think of cybersecurity for space systems as an afterthought to meeting functional requirements [21]. Satellite designers need to think of cybersecurity as an integral and emergent part of the system which must be planned into the design process of a space system [22], rather than an expensive afterthought to be avoided in the interest of keeping costs low.

The illusion that including cybersecurity considerations into the design process drives lifecycle costs up must also be emphasized and debunked – nothing is more expensive than getting hacked and having to pay a ransom to regain control of the satellite from a rogue entity or losing the satellite completely. While cost is important to all satellite manufacturers, commercial manufacturers have different solutions to minimizing cost when compared to their DoD counterparts. DoD tends to focus on protecting the ground stations and terrestrial access points and strips the design processes down to the bare minimum, while commercial companies tend to cut down on design systems such as cybersecurity implementation in favor of deploying the

satellite on a faster timeline so it can start generating income (operational phase) instead of costing money (design/build phase). This cost-reduction mindset, as well as a lack of thought towards threat conjecture, leads to a prioritization of delivering the satellite services first without considering the survivability or hackability of the satellite itself, and explains why it is atypical for a program to have strict cyber requirements as part of the baseline design. Even when cybersecurity is considered and designed for, generally the design focuses on the more accessible, terrestrial-based parts of the system such as the ground station and the encryption between the ground and the satellite, while the possibility that a threat actor could attack the satellite itself is rarely designed for or considered, for historical reasons.

On the basis of the state of the field, there is a need to reevaluate the current approach to cybersecurity protection on satellites. The tools necessary to explicitly design and execute cyber-secure satellite systems are not present in practice or in the literature. By using modern tools such as MBSE, the satellite can be modeled and then subjected to cyber threats to evaluate and identify the vulnerabilities of the system, and those findings can be systematically integrated into modern system design process [23]. This is especially important because MBSE provides a means of modeling the satellite, the ground system, other systems that interact with the satellite, and the threat actors. Using MBSE will enable the incorporation and design of a cyber-secure architecting process which applies terrestrial cybersecurity applications to satellites to show what needs to be accounted for and how to test the system. MBSE will enable developers to visualize and model their cybersecure system instantiations and evaluate them against threats. A goal of this research is to help guide the development of cybersecurity for satellites by using MBSE to model a satellite and the cybersecurity solution and then to dynamically perform threat modeling

to ensure corner cases are validated. In turn, this research aims to create a framework for others to apply MBSE to their cybersecurity challenges for satellites.

Chapter 2.

Research Questions and Tasks

This research fills this gap by posing and answering a set of research questions with associated tasks.

Overarching Research Question: What are the cybersecurity threats to space critical infrastructure, and what are the costs and benefits of a model-based systems engineering process in developing cybersecure satellite systems.

The tasks for defining the cybersecurity threats to critical space infrastructure in the context of Model-Based Systems Engineering are detailed below. A summary of the research questions and tasks are shown in **Figure 1**.

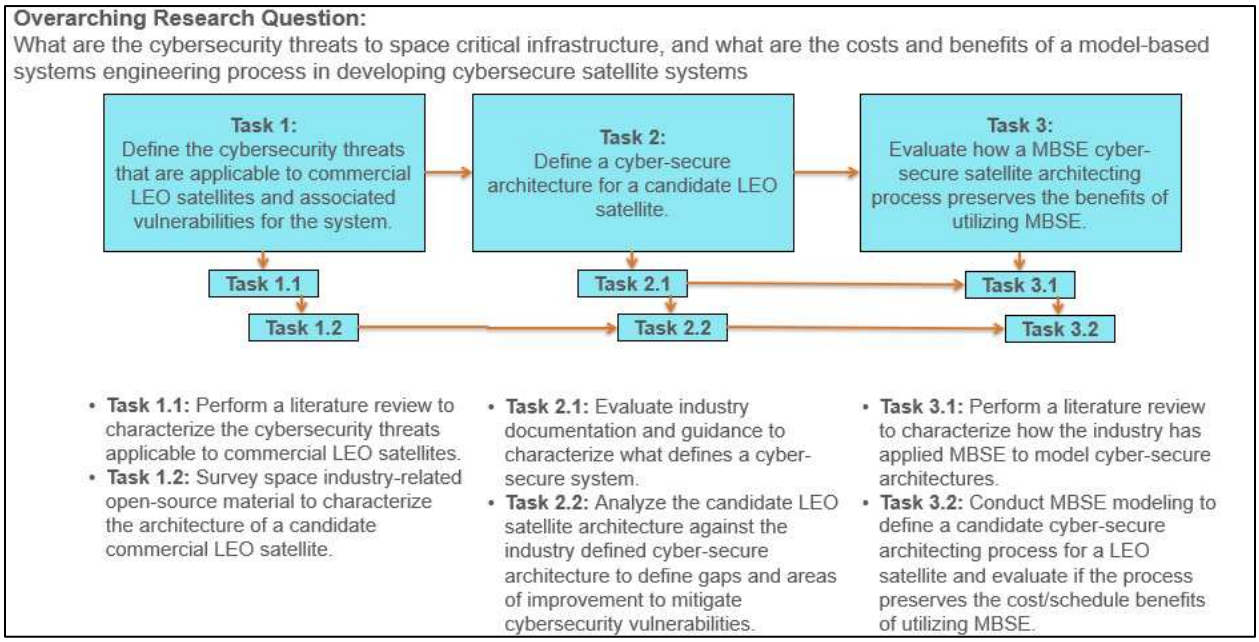

Figure 1: Summary of Research Questions and Tasks

Research Question 1: Define the cybersecurity threats that are applicable to commercial LEO satellites and associated vulnerabilities for the system. The tasks for defining the cybersecurity threats and vulnerabilities to satellites are:

Task 1.1: Perform a literature review to characterize the cybersecurity threats applicable to commercial LEO satellites. A broad range of literature exists on terrestrial systems and the associated cybersecurity vulnerabilities that can compromise them. The nature of satellites (their remoteness, their networking characteristics, their cyber physical nature, and their high value) limits the applicable threats and requires an application-specific list of cybersecurity threats. In addition, the way attacks are performed on satellites is different due to the default security settings of encrypted communication with secure ground stations. As a result, it is important to determine threat commonalities and how they apply specifically to present and future space applications.

Task 1.2: Survey space industry-related open-source material to characterize the architecture of a candidate commercial LEO satellite. Historically satellite design and development has been handled by industry and there are limited examples of robust institutionally built satellite designs that can be analyzed. Within the space system enterprise defined in Task 1.1, satellites typically have common architectures composed of standard subsystems to control the satellite bus and to host a mission-specific payload. Satellites have external interfaces which are all possible entry points for cybersecurity threats that can compromise their mission.

Research Question 2: Define a cyber-secure architecture for a candidate LEO satellite. The tasks for defining the cyber-secure architecture for a candidate LEO satellite are:

Task 2.1: Evaluate industry documentation and guidance to characterize what defines a cyber-secure system. Industry cybersecurity guidance has been rigorously developed for terrestrial systems. There have been various attempts by industry to "band aid" the space industry with tailoring of these documents, without a thorough dedicated tailoring effort. To define a cyber-secure satellite architecture, the existing guidance that industry experts recommend needs to be evaluated.

Task 2.2: Analyze the candidate LEO satellite architecture against the industry-defined cyber-secure architecture to define gaps and areas of improvement to mitigate cybersecurity vulnerabilities. This task will define a cyber-secure LEO satellite architecture.

Research Question 3: Evaluate how an MBSE cyber-secure satellite architecting process preserves the benefits of utilizing MBSE. The tasks for defining how an MBSE cyber-secure satellite architecting process preserves the benefits of utilizing MBSE are:

Task 3.1: Perform a literature review to characterize how the industry has applied MBSE to model cyber-secure architectures. MBSE enables early architecting, verification, and validation of a system [24]. When applied to cybersecurity, the MBSE model can act as an single source of truth (SSOT) to provide a complete picture of all the system interfaces and functionality in one location. This enables cybersecurity designers to create security-centric viewpoints to understand where the vulnerabilities are in the design and minimize the chance of missing critical interface information housed in

separate disconnected artifacts. In addition, an MBSE model enables cybersecurity engineers to define cybersecurity centric requirements, identify where in the architecture the requirements are verified, outline the security boundaries, explain the security controls necessary within each security boundary, and model defense-in-depth or zero trust techniques [25].

Task 3.2: Conduct MBSE modeling to define a candidate cyber-secure architecting process for a LEO satellite and evaluate if the model preserves the cost/schedule benefits of utilizing MBSE. MBSE is a relatively new tool to space technology, and applying terrestrial-based cybersecurity practices in satellites is even more novel than MBSE. This task will integrate the findings from research questions 1 and 2 to define a candidate cyber-secure MBSE model of a LEO satellite and will evaluate if the model preserves the cost/schedule benefits of utilizing MBSE.

.

Chapter 3.

Research Question 1: Review of the Cybersecurity Threats That Are Applicable to Commercial LEO Satellites and Associated Vulnerabilities for the System

3.1 Research Question 1 Introduction

This chapter addresses Research Question 1, which is restated as follows: Define the cybersecurity threats that are applicable to commercial LEO satellites and associated vulnerabilities for the system. To answer this research question, two tasks were defined.

Task 1.1: Perform a literature review to characterize the cybersecurity threats applicable to commercial LEO satellites. To answer Task 1.1, we evaluated literature to define the cybersecurity threats to a satellite. By evaluating Information Technology (IT) and Operational Technology (OT) threats, we were able to synthesize an understanding of common cybersecurity threats.

Task 1.2: Survey space industry-related open-source material to characterize the architecture of a candidate commercial LEO satellite. To answer Task 1.2, we leveraged our industry experience and researched common satellite components to define a baseline

architecture that a LEO satellite could utilize. The findings from this task are further expanded in Research Question 3, when we integrate the architecture into an MBSE model.

Our research seeks to address this question by leveraging our industry experience and an evaluation of industry-generated sources to serve as a foundation for the subsequent research questions.

3.2 An evaluation of the literature to characterize the cybersecurity threats applicable to commercial LEO satellites

Cybersecurity threats in the past few years against the critical infrastructure of the United States highlight a gap in the cybersecurity protection of US critical space assets. Historically programs have had weak cybersecurity hygiene and many National Security Space programs have been built with inadequate protection [26]. Space systems are critical to the infrastructure of the United States of America as they provide Global Positioning Systems (GPS), Communication Systems, and various global environmental monitoring functionalities. However, as with anything that is digital, remotely located, and has external system interfaces, these satellite constellations are vulnerable to cyberattacks. If one of these systems becomes compromised, it can have a direct impact on government as well as consumer functionality and livelihood. The United States seeks to protect these systems to ensure the critical infrastructure is not compromised. The White House came to the realization that as a country the US has been very focused on protecting terrestrial critical infrastructure but has not focused on critical infrastructure located in space [27]. In September of 2020, the White House signed Space Policy Directive-5 in order to establish cybersecurity principles that government and contracting organizations should follow to protect critical space systems from pending cybersecurity threats [27]. Many of the same “principles and practices” applicable to ground systems are also

applicable to space systems and can be tailored to fit space systems and their entire lifecycle development [27].

3.2.1 Comparing IT Systems to OT Systems

Terrestrial systems range from IT systems to OT systems. IT systems make up the computers and databases at various company locations and their interconnecting networks. OT systems are control systems for critical infrastructure systems and can be interfaced with the IT systems [28]. An overall space system is composed of both IT and OT aspects, and cyber threats affect the entirety of the space system. For the purpose of this research project, as shown in **Figure 2**, a Space System is composed of two main components:

- **Space Segment**: The Space Segment is composed of the spacecraft which will perform a specific mission. It is composed of multiple subsystems which handle the various functionalities of the spacecraft. For the purpose of this research project, the Space Segment is a small Low Earth Orbit (LEO) satellite, part of a constellation, that includes communication to the satellite from the ground and between other satellites in the constellation. The Space Segment is composed of OT systems.
- **Ground Segment**: The ground segment is a network of operation centers, operators, networks, and ground antennas. The Ground Segment allows the users to communicate and control the satellite from Earth. The Ground Segment is typically the only way to communicate with and control the spacecraft. The Ground Segment is composed of both IT and OT systems.

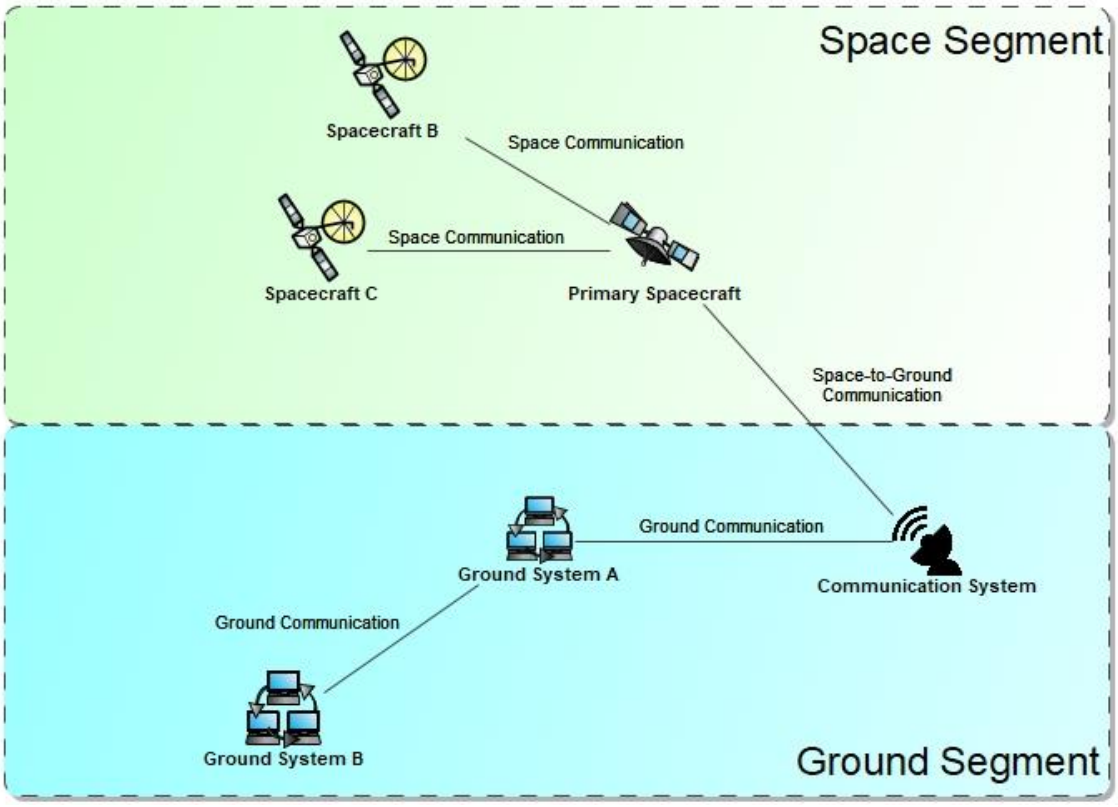

Figure 2: Space Segment and Ground Segment Partitioning

A space system is analogous to a typical OT Supervisory Control and Data Acquisition (SCADA) type system. In a SCADA system, control systems monitor and control endpoint machines, or systems of machines, for specific applications. In a Space System, the Space Segment is a satellite (the endpoint) being controlled by a user in the Ground Segment (the control and processing systems). The SCADA concept is shown in **Figure 3**. Applying the SCADA concept to a Space System allows for the identification of the vulnerabilities applicable to the system.

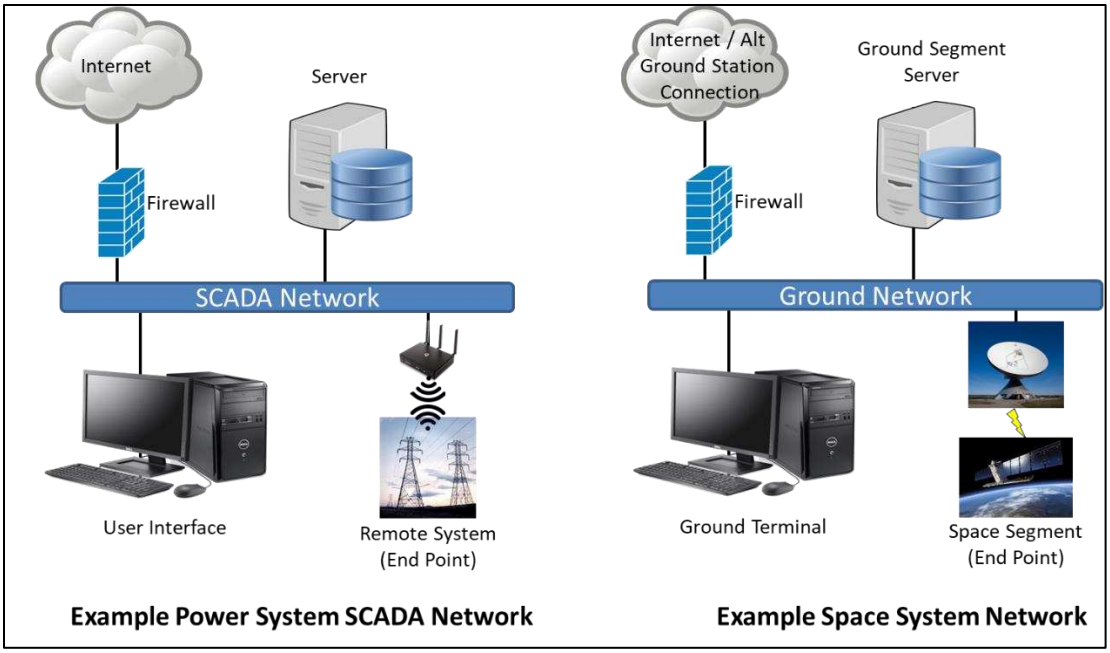

Figure 3: SCADA OT Concept Applied to Space

The Space and Ground Segments can each induce cybersecurity vulnerabilities into the overall Space System throughout its entire life cycle, from supply chain parts procurement through decommissioning of the satellite. Satellites are remotely operated systems that can either be flown autonomously or by commands sent by an operator on the ground. Due to their remote location in space, they are difficult to physically monitor for the purpose of determining the timing and source of an attack when it occurs. Any digital entry point into a satellite is a potential avenue of exploitation and/or infiltration. The Ground Segment is a significant external interface to the Space Segment and is a potential avenue for cyber threats. The focus of this research is on a LEO satellite within the Space Segment and its vulnerabilities to cyber threats. However, it is also important to understand the interfaces between the Ground Segment and the Space Segment in order to determine the cyber threats that could compromise the Space Segment.

To fully understand the cyber vulnerabilities of satellites and the implications of cyberattacks, the type of malicious actors that are a threat to our critical infrastructure and the common types of cyber threats that are applicable to space applications need to be determined.

3.2.2 Defining Threat Agents and Cybersecurity Threats for LEO Satellites

The malicious actors that are a threat to our infrastructure can be categorized as both domestic and foreign and can have varying levels of impact on an overall system. Their motivation varies from Political to Profit, Ideological to Violence, and Satisfaction to Discontent [29] [30]. As shown in **Table 1**, there are six distinct threat categories for these malicious actors.

Table 1: Types of Threat Agents to Cybersecurity [31] [32]

Tier	Name	Description	Goal	Potential Impact to A Space System
I	Script Kiddies	Typically, newbies who download existing cyberattack scripts and attempt to use them to attack systems	Disrupt and/or embarrass a group/organization	• Can exploit unpatched vulnerabilities in the Ground Segment IT infrastructure that supports the Space Segment
II	Hackers for Hire	Established individual hackers who know how to write scripts to exploit systems; small scale	Disrupt and/or embarrass a group/organization	• Can exploit unpatched vulnerabilities in the Ground Segment IT infrastructure that supports the Space Segment
III	Small Hacker Teams	Groups of hackers who know how to write scripts and tools to exploit systems; medium scale	Obtain information and/or impede an organization's capability	• Can exploit vulnerabilities in the Ground Segment IT infrastructure • Can exploit vulnerabilities in the communication between the Ground Segment and Space Segment
IV	Insider Threats	Can be intentional (disgruntled employees) or unintentional (accidental); people who have access to systems' internal workings and can expose them or compromise them	Obtain critical information and/or impede an organization's capability	• Can exploit vulnerabilities in the Ground Segment IT infrastructure • Can exploit vulnerabilities in communication systems between Ground Segment facilities • Can exploit vulnerabilities in the communication between the Ground Segment and Space Segment • Can exploit vulnerabilities in the communication between friendly satellites • Can exploit vulnerabilities at the external interfaces of the satellites • Can exploit vulnerabilities within the satellites
V	Large, Well-organized teams, criminal	Well-organized group of hackers who can compromise systems by exploiting known and unknown vulnerabilities; large scale i.e., ransomware actors	Undermine and/or impede an organization's capability. Obtain sensitive information	• Exploit vulnerabilities in the Ground Segment IT infrastructure • Exploit vulnerabilities in communication systems between Ground Segment facilities • Exploit vulnerabilities in the communication between the Ground Segment and Space Segment • Exploit vulnerabilities in the communication between friendly satellites • Exploit vulnerabilities at the external interfaces of the satellites • Exploit vulnerabilities within the satellites
VI	Highly-Capable Actors	Organized groups of people whose main focus is on determining vulnerabilities in systems and exploiting them	Undermine and/or destroy an organization's capability	• Exploit vulnerabilities in the Ground Segment IT infrastructure • Exploit vulnerabilities in communication systems between Ground Segment facilities • Exploit vulnerabilities in the communication between the Ground Segment and Space Segment • Exploit vulnerabilities in the communication between friendly satellites • Exploit vulnerabilities at the external interfaces of the satellites • Exploit vulnerabilities within the satellites

Current Space Systems are typically built with common cybersecurity architectures to mitigate the Tier I, II, and III Threat Agents through the use of access-controlled environments, encrypted communication between the Space Segment and Ground Segment, robust flight software, and secured system architectures [33] [34]. The vulnerabilities in Space Systems usually come from the three remaining threat agent groups. The Tier IV actors are more difficult to mitigate since they are trusted personnel working within the design and operation of a Space System who can compromise the system either intentionally or unintentionally. The Tier V and VI threat agents are challenging to protect against because, as dedicated adverse organizations, they focus on ways to compromise the vulnerabilities in Space Systems. The common cybersecurity architectures which are currently used today are somewhat effective at preventing the Tier IV, V, and VI threat agents from compromising the Space System. However, these threat agents can potentially still exploit vulnerabilities in a Space System. To prevent the exploitation, it is therefore important to determine and mitigate the potential vulnerabilities.

IT systems existed before modern OT systems. Cyber threats to systems originated from IT systems. However, the rise of OT systems has led threat actors to adapt their methods of attack to OT systems [35]. For an OT system, such as the Space Segment, the IT cyber threats need to be evaluated to determine their applicability to an OT system. An OT Space Segment is typically a deployed system with onboard software that controls the satellite functionality. The typical satellite architecture does not have user input fields, webpages, or video screens to access a Graphical User Interface. There are many commonly known cybersecurity threats that can compromise vulnerabilities in an IT infrastructure.

Table 2 shows some of the most common ones. These threats have been analyzed and annotated with their applicability to an explicit OT system such as the Space Segment.

Table 2: Common IT Cyber Threat Categories [25] [29] [36]

Threat Type	Summary Description	Applicability to OT Space Segment
Adware	Advertising software, can contain Malware/PITM/Spyware	Yes. This can be categorized as Software Threats. It may be possible to upload Malware onto a Satellite
Backdoor	Vulnerable external interface to software that allows bypassing security measures	Yes. This can be categorized as Insider Threats, or someone that knows the encryption of the communication
Bots and Botnets	Compromised IT device / devices that enable threat actor control and are connected to a network	Yes. This can be categorized as Software Threats
Code Injection	Taking advantage of vulnerabilities in software code	Yes. This can be categorized as Insider Threats, or someone that knows the encryption of the communication
Drive-By Exploit and Watering Hole	Malicious code that has compromised a website and the systems of visitors	No. This is a website threat
Denial of Service (DoS)	Overload server capabilities, sends large amounts of queries	Yes. If someone knows how to communicate with the satellite, they could perform a DoS attack
Exploits And Exploit Kits	Compromising unpatched vulnerabilities in software systems	Yes. This can be categorized as Software Threats
Formjacking	Compromising form fields on webpages to steal information	No. This is a website threat
Insider Threat	Insiders either intentionally or unintentionally circumventing system controls or taking advantage of their accesses	Yes. This can be categorized as Insider Threats
Password Cracking	Cracking passwords to log into user accounts	No. This is more applicable to a Ground Segment for user authentication to access the systems that control the Space Segment
Person-In-The-Middle (PITM)	Interception of communication data between devices/data centers	Yes. This can be categorized as a Replay Attack, where someone is intercepting communication data
Potentially Unwanted Program or Application	Unnecessary software on a computer that can be avenues of vulnerabilities	Yes. This can be categorized as Software Threats
Ransomware	Malicious code that compromises an IT system and prevents access by the users	Yes. This can be categorized as Software Threats
Replay Attack	Intercept data and use to replay later	Yes. This can be categorized as a Replay Attack, where someone is intercepting communication data
Rootkit	Malicious code that gives a threat actor administrative privilege	No. This is more applicable to a Ground Segment
Spoofing	Trick target into thinking you're a trusted network	Yes. It's possible to trick sensors into thinking they are collecting good data
Spyware	Malicious code that tracks user behavior and information	Yes. This can be categorized as Software Threats
SSL Hijacking	Threat Actor compromising unsecure connections between users and data centers	No. This is more applicable to a Ground Segment

These well-known cyber threats, shown in **Table 2**, can compromise the IT infrastructure in the Ground Segment which supports the Space System. However, depending on how the cyberattack is initiated, these categories of cyber threats are also applicable to OT systems such as a Space Segment. These IT cyber threats inform the focus of this research on the vulnerabilities and gaps in the OT aspects of a space system. However, the IT cyber threats mostly pertain to compromising common webpages, user computers, or IT infrastructure networks. They need to be tailored to the specific OT application due to the various OT architectures (i.e. user consoles, operating systems, and access control schemes). Due to the diverse nature of OT systems, the way a cyber threat is carried out is tailored to that specific OT system. As a result, the avenues through which Space Systems are vulnerable to a cyber threat are not directly the same as a Power Distribution Plant or a Water Treatment Plant, but the general method could be leveraged to tailor the cyber threat.

Historically, industry has been focused on IT threats such as computer networks, websites, and databases, with far less focus on threats to OT systems. It was believed that OT systems such as pipelines and satellites were inherently immune to cyber threats due to their remote locations and their disconnectedness from remote IT networks. However, the previously discussed attacks on OT systems like the Colonial Pipeline which occurred in 2021 prove this assumption of OT safety is no longer true, and because terrestrial OT systems are no longer safe, the same assumption should prudently be made regarding satellites. Despite often being interconnected and reliant upon each other for total system operation, IT systems are not synonymous with OT systems. For commercial applications especially, these IT cybersecurity "best practices" can be very costly to implement in Space where resources are limited and

manufacturers are relying on older technologies that aren't necessarily compatible with terrestrial "best practices" based on newer IT technologies. Therefore, as goal of Research Question 1 Task 1 and Task 2 is to understand the field and synthesize what makes sense for commercial space.

An analysis of the IT cyber threats in **Table 2** determined that some of them can compromise an OT satellite. The cyber threats were applicable to an OT system if they could compromise an interface or the flight software on a satellite. Focusing on an OT perspective of the Space Segment, the internal and external interfaces of a Satellite must be determined to identify the cybersecurity vulnerabilities of the system. Combining the IT cyber threats of **Table 2** with additional research tailored to cyber threats against satellites yields seven common categories of cyber threats when applied to an OT Space Segment, which are defined in **Table 3**.

Table 3: Common OT Space Segment Cyber Threat Categories [17] [37] [38] [39] [40]

Threat	Space Segment Description	Potential Space Segment Impact
Denial-of-Service (DoS)	DoS is when a threat actor has broken into the Space System communication channels to the Satellite and sends numerous commands to its various interfaces in an attempt to lock up the flight computers and/or various components of the vehicle.	• Loss of entire mission • Loss of partial mission (if recovered)
Masquerade	Masquerade is when a threat actor is pretending to be a friendly asset. The threat actor can masquerade as a friendly cross-linked satellite or a friendly ground station. Masquerade is possible when a threat actor has broken the security and access controls of a satellite and knows how it operates.	• Loss of entire mission • Loss of partial mission (if recovered) • Loss of sensitive data
Unauthorized Access	Unauthorized Access is when a threat actor (intentional) or a ground operator (accidental) compromises the physical security of a friendly ground station to access the control systems, or has masqueraded into a friendly satellite cross-linked to the satellite of interest. The threat actor has broken the security and access controls of a satellite and knows how it operates.	• Loss of entire mission • Loss of partial mission (if recovered) • Loss of sensitive data
Replay	Replay is when a threat actor intercepts the communication paths either between Satellites in a constellation or between a Satellite and a ground station. The threat actor can then "replay" that intercepted data in an attempt to compromise the commanding to the satellite, or give a ground operator a false state of being for the satellite.	• Loss of entire mission • Loss of partial mission (if recovered) • Confusion
Software Threats	Software Threats come in two categories. The first category is by the people who build the satellite. They could have inadvertently missed a software flaw which can cause the satellite to act in a way that is unintentional. The second category is Malware. If a threat actor has broken the security and access controls (either by Masquerade or Unauthorized Access) it may be possible for them to upload malicious code to the satellite. The malicious code could be obviously adverse or simply benign and undetected.	• Loss of entire mission • Loss of partial mission (if recovered) • Loss of sensitive data
Tainted Hardware Components	Tainted Hardware is when a threat actor has compromised the components of a satellite during their procurement life cycle. Due to the use of COTS components, vulnerabilities in COTS parts are becoming well known and threat actors could hide malicious hardware/software inside the hardware which could compromise the integrated system.	• Loss of entire mission • Loss of partial mission (if recovered) • Loss of sensitive data
Jamming	Jamming is when a threat actor overcomes the external interfaces of a satellite preventing it from communication with friendly assets. Jamming can be thought of like a DoS at the RF level.	• Loss of entire mission • Loss of partial mission (if recovered)

As shown in **Figure 4**, there are various ways a cyber threat can be initiated on a Space System. Combining these cyber threats with the Threat Agents defined in **Table 1** and the system

partitioning in **Figure 2** reveals an understanding of where in the Space System architecture these threats occur and the sophistication of the cyber threat. **Figure 4** shows the combined view of the Space System. As shown in **Figure 4**, the Primary Spacecraft has multiple external interfaces with other spacecraft and ground stations. In addition, the Primary Spacecraft is vulnerable from its internal interfaces. The seven cyber threats in **Table 3** are all applicable to these various interfaces. In addition, the Threat Actors that could be conducting these cyber threats are highly sophisticated and knowledgeable in the operation of the Space Segment.

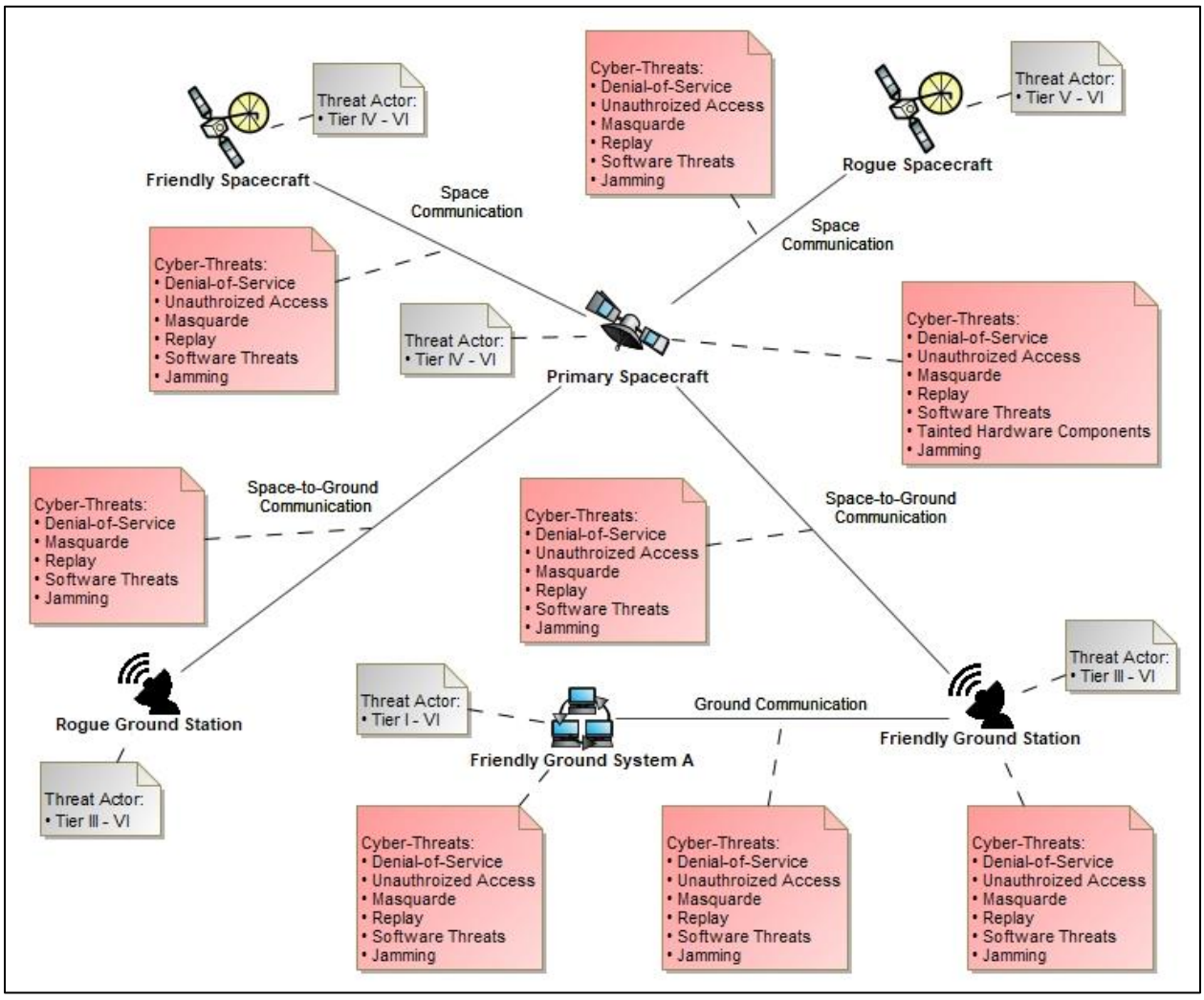

Figure 4: Space System Threat Landscape [31] [37]

For many of these cyber threats, the test phase of a Space System typically has a lot of threat mitigation checks and balances. This process typically catches and attempts to prevent the vulnerabilities from occurring on the deployed system. For the purpose of this research, these threats are being looked at through the lens that a vulnerability was not caught during the build process and could theoretically be exploited.

3.3 A survey of space industry-related open-source material to characterize the architecture of a candidate commercial LEO satellite

LEO satellites typically have common architectures composed of standard subsystems to control the satellite bus and to host a mission-specific payload. Satellites can be thought of as a system of systems which have external interfaces that could compromise the rest of the satellite. External interfaces are all possible entry points for cybersecurity threats which can compromise the mission. In addition, depending on how well the flight software to fly the satellite was written or how well the internal components were vetted, there could be internal entry points for cybersecurity threats. This task will review a common LEO satellite architecture which could be part of a proliferated LEO constellation and determine vulnerabilities in the design where malicious actors could intrude.

3.3.1 Defining the top level architecture of a LEO Satellite

The focus of this research is on the Space Segment and will discuss elements of the other systems within the enterprise in regards to external interfaces. The Space Segment pertains to the spacecraft which is composed of various interdependent subsystems. Each subsystem controls mission-critical functionalities. The overall subsystem architecture is leveraged from the space industry as almost all space missions have a similar subsystem architectural organization. The

system architecture is typically composed of the subsystems shown in **Figure 5**, which was developed from an analysis of a variety of smaller satellite space mission architectures.

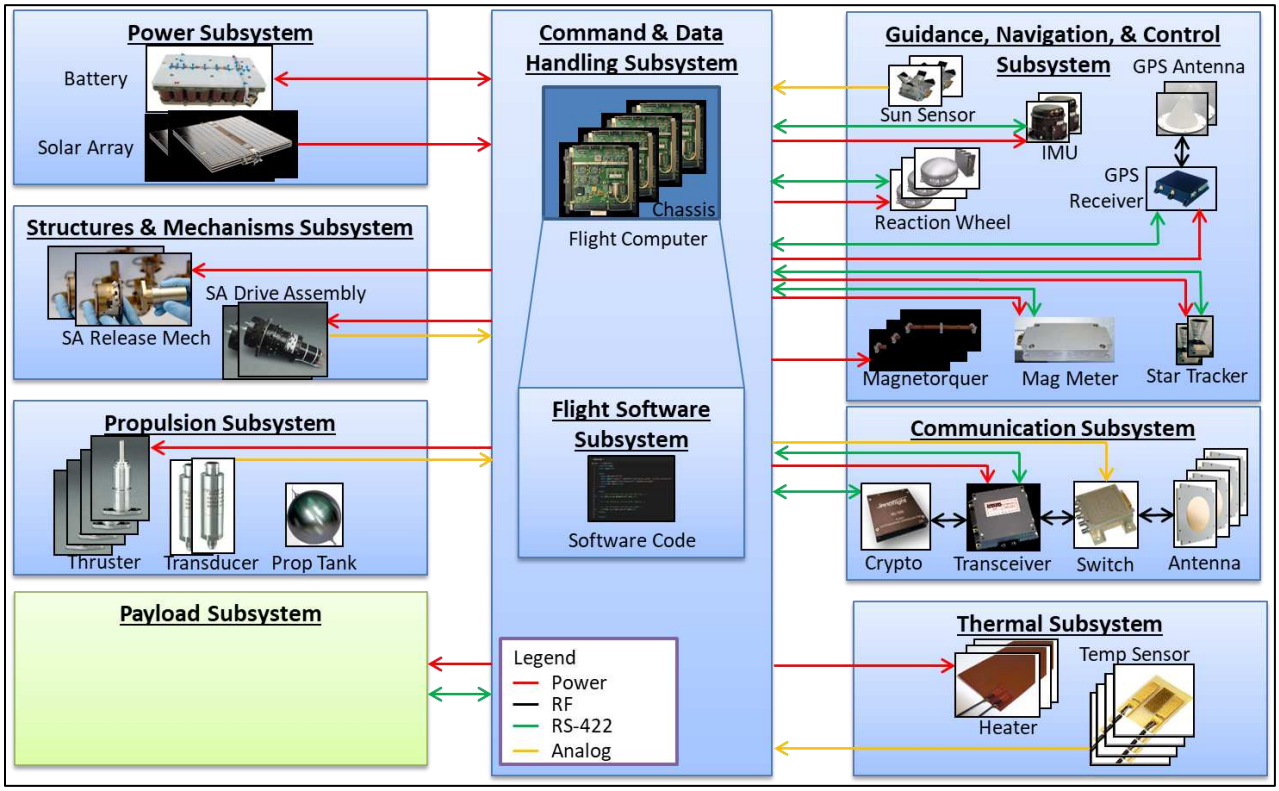

Figure 5: Generic LEO Satellite Architecture [41] [42] [43] [44] [45] [46]

A Spacecraft is typically made up of the following nine subsystems, as shown in **Figure 5.**

- **Command & Data Handling Subsystem:** The command and data handling subsystem is the heart of the spacecraft. All subsystems will be routed through an avionics component for interface and control. The flight software and the avionics components are the glue that holds the spacecraft subsystems together.
- **Power Subsystem:** The power subsystem is responsible for generating, storing, and controlling power on the spacecraft. This is accomplished through solar array panels, batteries, relays, and power control logic.

- **Communication Subsystem:** The communication subsystem is responsible for communication between the spacecraft and the ground system on Earth. This is accomplished through a transceiver and gimbaled antennas which maintain a constant communication link.
- **Guidance, Navigation, & Control Subsystem:** The guidance and navigation subsystem is responsible for guiding the spacecraft and its fine controls. This is accomplished through inertial measurement units, reaction wheels, and optical sensors.
- **Thermal Subsystem:** The thermal subsystem is responsible for heating & cooling the spacecraft. This is accomplished through thermistors and heaters.
- **Propulsion Subsystem**: The propulsion subsystem is for propelling the spacecraft through space. This is accomplished through altitude control thrusters and orbit control thrusters.
- **Structures & Mechanisms Subsystem**: The structures subsystem is the structure of the spacecraft and any structural mechanisms.
- **Flight Software Subsystem:** The flight software is all the software on the spacecraft which acts as the logic controlling and linking all subsystems.
- **Payload Subsystem**: The payload is the overall purpose of the satellite and the mission. It can be composed of many of the components within the subsystems noted above.

3.3.2 Evaluating the industry components that make up each LEO satellite subsystem

Examining the components that make up the subsystems of the satellite exposes the latent vulnerabilities in the overall satellite. Analog and digital components both have unique vulnerabilities which must be addressed. Digital components can be compromised through their data buses, by malware, or via embedded nefarious components. Analog components, such as

thrusters, valves, or switches, are more difficult to compromise from a software perspective, but well-vetted supplier chains ensure they do not have physical bugs or intentionally added nefarious parts. Many of the suppliers hold their product sheets at proprietary levels. By leveraging commonly known components with common interfaces utilized today, as shown in **Table 4**, the components that make up this concept LEO Small Satellite were determined in **Figure 5**.

Table 4: Satellite Component Interfaces Exemplified

Component	Supplier	Part Number	I/O Interface	Reference
Command & Data Handling				
Flight Computer				
	DDC	SCS750G4	SpaceWire, 1553, LVDS, RS422, GPIO/LVCMOS	[47]
	SwRI	HP-SBC	SpaceWire, LVDS, RS422, GPIO, Ethernet	[48]
	CAES	DS4350272-X00	CAN, SpaceWire, 1553, RS422, LVDS, Ethernet, GPIO	[49]
	Aitech	SP0-S	Ethernet, RS422, GPIO (TTL)	[50]
	BAE	RAD5545	SRIO, SpaceWire, 1553, RS422, LVDS	[51]
	Innoflight	CFC-400	RS422, LVDS, SRIO, 1553, SPI	[52]
	SEAKR	Medusa	1553, SpaceWire, RS422, Ethernet, GPIO	[53]
Power				
Battery				
	Space Inventor ApS	BAT100-P3	CAN	[54]
	Ibeos	B28-135	Analog	[54]
	SAFT	VES16 8s4p battery	Analog	[55]
Solar Array				
	SparkWing	SparkWing	Analog	[56]
	EnduroSat	6U Deployable Solar Array	Analog	[57]
	Blue Canyon	3U Solar Array	Analog	[58]
Communications				
Transceiver				
	L3Harris Technologies	MSX-765 Transceiver	RS-422	[59]
	IQ Spacecom	SLink-PHY Transceiver	RS-422	[60]
	Tethers Unlimited	SWIFT-SLX Transceiver	RS-422, LVDS, SpaceWire, Ethernet	[61]
	Honeywell	STC-MS03	RS-422	[62]
	Innoflight	SCR-106	RS-422, LVDS, LVCMOS	[63]

		Blue Canyon	SDR	LVDS	[64]
	Switch				
		Radiall	Low Power Coaxial DP3T Switch	LVCMOS	[65]
		Teledyne	33SDC	LVCMOS	[66]
		Renaissance Electronics Corporation	SW-316	LVCMOS	[67]
	Crypto				
		Innoflight	KI-103	RS-422, LVDS, LVCMOS	[68]
		L3Harris Technologies	MCU-110C	RS-422	[69]
	Antenna				
		EnduroSat	S-Band Antenna Commercial	RF	[70]
		SpaceQuest Ltd.	AC-2000	RF	[71]
	GPS Antenna				
		L3Harris Technologies	AS-48917	RF	[72]
		Innovative Solutions In Space B.V. (ISIS)	S-band Patch Antenna	RF	[73]
		ANYWAVES	S-band Antenna	RF	[74]
Guidance, Navigation, and Control					
	Star Tracker				
		Blue Canyon	Full Extension NST	RS-422, RS-485	[75]
		Terma	T1 (45 Deg Baffle)	SpaceWire, RS-422	[76]
		Space Micro	µSTAR-200M	SpaceWire	[77]
		Sodern	Hydra-TC	1553, RS-422	[78]
	IMU				
		L3Harris Technologies	CIRUS-EX	1553, RS-422	[79]
		Sensonor AS	STIM202	RS-422	[80]
		Northrop Grumman	LN-200S	RS-422, RS-485	[81]
		Honeywell	MIMU	1553, RS-422	[82]
	Reaction Wheel				
		Blue Canyon	RWP100	RS-422, RS-485	[83]
		L3Harris Technologies	RWA-15	Analog Voltages	[84]
		Microsat Systems Canada	MicroWheel 1000	RS-422, RS-485	[85]
		Hyperion Technologies	RW210-6.0	Analog Voltages	[86]
		Millennium Space Systems	RWA-1000	CAN, RS-485	[87]
		Bradford Space	W45	1553, Analog	[88]
	Magnetometer				
		ZARM Technik AG	AMR Magnetometer	RS-422	[89]
		Magson GmbH	MACM Fluxgate	RS-422, RS-485	[90]
		NewSpace Systems	NMRM-Bn25o485	RS-422, RS-485	[91]
	Magnetorquer				
		ZARM Technik AG	MT70-2	Analog	[92]
		Sinclair Interplanetary	TQ-15	Analog	[93]
		NewSpace Systems	NCTR-M012	Analog	[94]
		Chang Guang Satellite Technology	Magnetic Torquer	Analog	[95]
	Sun Sensor				
		Adcole Maryland Aerospace	Coarse Sun Sensor	Analog	[96]
		Solar MEMS Technologies	SSOC-A60	Analog	[97]

		Hyperion Technologies	SS200	Analog	[98]
		Bradford Space	Fine Sun Sensor	Analog	[99]
	GPS Receiver				
		RUAG	LEORIX GNSS Receiver	1553, RS-422, SpaceWire, PPS	[100]
		Surrey Satellite Technology	SGR-Axio	CAN, RS-422	[101]
		MOOG	NavSBR	RS-422, LVDS, 1553	[102]
		General Dynamics	Viceroy-4 GPS Spaceborne Receiver	RS-422	[103]
		Airbus	LION 1100Neo GNSS Receiver	1553, RS-422, SpaceWire	[104]
Thermal					
	Heater				
		MINCO	4009/003	Analog	[105]
		Tayco Engineering	Flexible Heater	Analog	[106]
		All Flex	Thermofoil Heaters	Analog	[107]
	Temperature Sensor				
		Tayco Engineering	Surface Temperature Sensor	Analog	[108]
		Variohm	ESA/ESCC Space Qualified NTC Thermistors with Leads	Analog	[109]
		Renesas	ISL71590SEH	Analog	[110]
Propulsion					
	Thruster				
		MOOG	Thrust Single Seat	Analog	[111]
		Bradford Space	1N HPGP	Analog	[112]
		Aerojet	MR-103G	Analog	[113]
	Latch Valve				
		ArianeGroup	Low Pressure Latch Valve	Analog	[114]
		VACCO	V1E10537-01	Analog	[115]
	Pressure Transducer				
		Stellar Technology	ST1300	Analog	[116]
		Taber Industries	2211	Analog	[117]
	Propulsion Tank				
		ArianeGroup	177 L Hydrazine Tank - OST 31-1	N/A	[118]
		MT Aerospace AG	E3000-590	N/A	[119]
Structures & Mechanisms					
	SA Gimbal				
		MOOG	Type 2 Solar Array Drive Assembly	Analog	[120]
		SNC	C14 Bi-Axis Gimbal	Analog	[121]
		RUAG	SEPTA® 24	Analog	[122]
	Solar Array Release Mech				
		SNC	Hold Down Release Mechanism (HDRM) 400 W Articulated Array	Analog	[123]
		RUAG	Hold Down and Release Mechanism	Analog	[124]

3.3.3 Evaluating component interfaces and potential internal vulnerabilities

From the prior section, our candidate satellite architecture only has one external interface with the ground segment: the communication subsystem radio. The majority of the threats from **Table 3** (Denial-of-Service, Masquerade, Unauthorized Access, Replay, and Jamming) pertain to cybersecurity threats external to the satellite trying to get in and compromise the system. The remaining 2 threats (Software Threats and Tainted Hardware Components) pertain to cybersecurity threats internal to the satellite. Cybersecurity threats on board the spacecraft can come in the form of monitoring sensitive data and/or injecting false data to corrupt the system [125].

From **Section 3.3.2** we have determined that in common satellite architectures, the component interfaces are typically composed of either point-to-point interfaces (such as Analog, RS-422, RS-485, LVDS, or SpaceWire), or data buses (such as CAN, MIL-STD-1553). The point-to-point interfaces can be perceived to be more secure since all information is routed to the flight computer and the flight computer acts as the source of truth for all spacecraft operation. Some satellite architectures utilize common data buses, such as MIL-STD-1553 or Controller Area Network (CAN), as the backbone of their architecture. These protocols are typically unsecure and don't provide any cybersecurity features to authenticate the source of commanding.

One of the basic mitigations to these kinds of threats is that satellites are complex systems and compromising them requires specific knowledge of the system and its internal interfaces [126]. Using CAN as an example, the benefits of this style of architecture allows the design to attach all components to a common harness interface. This allows for each of the attached components to send and receive commands from the CAN bus [127]. This allows for integration flexibility and redundancy architectures by having multiple CAN Nodes that have the capability

to command the bus [128]. For example, for a spacecraft the architecture can have multiple flight computers on the same CAN bus so in the event that there is a hardware failure the other flight computer(s) can seamlessly take over commanding. However, on the flip side, because the CAN bus essentially interlinks all components on the satellite, cyber threats can gain access to all interconnected components [129]. As satellites become more modernized with multiple communication points on this vital backbone, the number of possible threat access points increases as well [129]. Some key vulnerabilities of data buses are:

- Replay attacks where communication is replayed over the bus [130].
- Denial of service attacks where the bus becomes unusable [130].
- Unrestricted monitoring and/or prevention of fuzzing of the data to identify cyber vulnerabilities [131].
- Lack of supervision to identify cyber threats [132].

These cyber threats are also relevant for point-to-point interfaces which do not have a data bus. These threats would be isolated between each component and the flight computer, and unable to travel to other components due to lack of a data bus. Based on this evaluation we have determined a few key aspects that satellites should monitor to help detect and mitigate these cybersecurity threats:

- Have the capability to detect anomalous behavior of components.
- Have the capability to determine if component commanding is out of bounds or not in accordance with expected communication.
- Have the capability to determine if component telemetry is out of bounds or not in accordance with expected communication.

3.4 Research Question 1 Conclusion

Upon completion of this research, we can now answer Research Question 1: Define the cybersecurity threats that are applicable to commercial LEO satellites and associated vulnerabilities for the system.

Our literature review allowed us to evaluate the current state of the field and synthesize an understanding of common cybersecurity threats with particular attention to those threats which endanger satellites. We then used our industry experience and completed additional research into common satellite components, allowing us to define a cyber-secure architecture which a LEO satellite program could use as a baseline. This research showed that many of the common terrestrial cybersecurity threats are also applicable to space systems, and that satellites are inherently susceptible to modern cyberattacks despite their remote locations and physical accessibility barriers.

Research Question 2 will leverage the threats and vulnerabilities to a satellite as defined in Research Question 1 as guidance to define a baseline cyber-secure architecture.

Chapter 4.

Research Question 2: Defining a Cyber-Secure Architecture for a Candidate LEO Satellite

4.1 Research Question 2 Introduction

This chapter addresses Research Question 2, which is restated as follows: Define a cyber-secure architecture for a candidate LEO satellite. To answer this research question, two tasks were laid out.

Task 2.1: Evaluate industry documentation and guidance to characterize the definition of a cyber-secure system. To answer Task 2.1, we evaluated literature to characterize the sources of satellite vulnerabilities and challenges satellite manufactures are facing in building cybersecurity systems. There is not currently a central organization which governs all cyber best practices. Instead, each program generates its own processes and guidance depending on the customer organization, the risk class of the program, and its requirements. Although NIST is typically an instrumental source of information due to its terrestrial efforts at mitigating risk, NIST is not the single source of truth. Even the government has many divisions and branches that specialize in weapon system cybersecurity, but these organizations are disconnected from the design process of each program.

Task 2.2: Analyze the candidate LEO satellite architecture against the industry defined cyber-secure architecture to define gaps and areas of improvement to mitigate cybersecurity vulnerabilities. To answer Task 2.2, we leveraged our industry experience to

emphasis the importance of building cybersecurity into the system architecture. In addition, we highlighted the challenges that satellite engineers and cybersecurity engineers are facing when they build satellites and proposed a solution to help reduce the functional siloing which is inherent to the space industry.

The research seeks to address this research question by leveraging our industry experience and an evaluation of industry generated sources to provide a starting point for satellite designers to integrate cybersecurity into the requirements process and enable the design of a cyber-secure architecture from the beginning of a program lifecycle.

This Research Question was written concisely to be submitted to Space Force Journal, and therefore the following content is written with the article format in mind. However, due to the timeline of completing this Dissertation, we were unable to submit the article to the journal.

4.2 Research Question 2 Abstract

The purpose of this article is to raise awareness of the lack of Mission Cybersecurity consideration for Class C/D satellites across the space industry, to describe why cybersecurity must be considered, and to encourage the aerospace companies to integrate cybersecurity requirements early into the design process to enable them to evolve alongside the cybersecurity threat landscape.

To defend this assertion, we analyzed the way that cybersecurity is conventionally addressed in the development of satellite programs; discussed how mission cybersecurity is currently being designed and implemented during the development process of Class C/D satellite missions and the pitfalls of the current application of cybersecurity on these programs; and compared the current methodology with a more traditional, requirements-based design approach

driven by systems engineering processes and procedures. We also made suggestions for cybersecurity-specific requirements to aid in the implementation of a more traditional systems engineering design process on Class C/D satellite missions.

If industry can adopt a more disciplined and systems engineering centric viewpoint of cybersecurity for Class C/D satellite missions by building in the requirements for the implementation of cybersecurity throughout the satellite system including the satellite from the beginning of the design process, these companies will reduce the risk of cybersecurity vulnerabilities in their missions. This will increase the success rate of these missions while providing the added benefit of a streamlined cybersecurity design process and therefore a reduction in cost and schedule.

4.3 Cybersecurity Vulnerabilities in Satellites

Historically, space system manufacturers have relied on proven tools and spacecraft architectures. As a result of their ongoing success, they have continued to pass those tools and spacecraft architectures down onto the next projects. This means that the architectures and processes for satellite design and development in use today originated from long-lived large-scale government satellites, and may have not been significantly updated or improved in the decades since their inception. This has established a development environment that is slow to change and has not adapted to the modern mission cybersecurity threats which directly threaten the satellite itself [133]. Due to the specialized considerations of satellite operations, space hardware has been designed and built for selective applications and was only available to the specific government contractors who had worked closely with government organizations to pay for their development [15]. Specialized parts design and development provided an unappreciated and exclusionary form of security protection which worked at the time, since the knowledge of

how to interact with and operate satellites was not well known outside space manufacturers and governments [16]. At present, the predominant mindset towards cybersecurity in satellite missions relies solely on encrypted communication to the satellite: Because the communications and ground stations are encrypted and protected, the satellite itself is therefore also secure from external cybersecurity threats.

Until recently, the concept of a non-state actor accessing and communicating with a satellite was not a prospect worthy of consideration [15]. There have been limited case studies into the cybersecurity exploitation of satellites because historically the academic community has not been involved in the detailed development and operation of satellites. The few publicly-available examples only describe how threat agents compromised the command-and-control interface to the satellite, and in all of these cases the operators were able to successfully recover the satellites [14] [134] [135] [136], without long-term mission impact. Because of this lack of reporting, and because of the added expense associated with protecting systems up-front, the Class C/D satellite developers are beginning to make the same mistakes that were made by terrestrial system designers. They trust the protective power of the "air gap" between Earth and Space, and they are therefore not protecting their developing space infrastructure, instead choosing to focus their efforts and expense on the protection of ground systems [137].

The growing cybersecurity vulnerabilities come among broad changes in the infrastructure of space communications and infrastructure. The government has been pushing industry to build faster, cheaper, and smaller satellites [138]. This shift has reinvigorated the space industry and opened access to both new innovators and malicious actors. With this shift in industry comes a shift in the supply chain for satellite components. Rather than being proprietary, purpose-built, and obscure to procure, many vendors have developed commoditized Commercial-off-the-Shelf

(COTS) hardware that space manufacturers can utilize to design and build their satellites. This public availability of COTS components enables threat actors to purchase them for leisurely evaluation and/or create their own COTS-based hacking devices [16] [139]. These hackers can learn how the COTS products work and determine the specific cybersecurity vulnerabilities for each COTS product which could be exploited once the parts are integrated into a satellite [17]. Similarly, digitization of design artifacts has led to increased accessibility for many of the specification sheets and user guides so threat actors can easily learn how to operate the hardware. These factors have combined to create a significant challenge in modern satellite development since there is a customer desire to minimize the growing cost to build a satellite while providing a robust product. The affordability of COTS components drives the industry to use them, which could embed potential vulnerabilities into a system [18] [19].

These challenges of space system development have resulted in a lag in the advancement of cybersecurity protection requirements and methodologies for the satellites themselves. Until recently there has even been active resistance to adopting cybersecurity requirements and testing: "DOT&E and service test agencies said that prior to around 2014, program offices tried to avoid undergoing cybersecurity assessments because they did not have cybersecurity requirements and therefore thought they should not be evaluated" [140]. DOD's weapon systems acquisition process has also struggled to deliver weapons that are cyber resilient [26]. The United States government can no longer afford to think of cybersecurity for space systems as an afterthought to meeting general requirements [21]. Satellite designers must start thinking of cybersecurity as an integral and emergent part of the system which must be planned into the design process of a space system [22], rather than an expensive afterthought to be avoided in the interest of keeping costs low.

Modern efforts to provide general design guidance to satellite designers have resulted in a 4-class scale, with satellites being classified as Class A, B, C, or D. The most prominent guidance adopted by government customers is TOR-2011(8591)-21: *Mission Assurance Guidelines for A-D Mission Risk Class* [141]. This guidance was developed by the Aerospace Corporation with input from numerous industry aerospace contractors, and was intended to provide a standard and simplistic way for industry to equate design rigor against program design life / cost / schedule. The Aerospace Corporation evaluated all aspects of the life cycle of a program and provided a scale for design robustness in regards to program execution, design, test, execution. The document details how Class A/B and C/D missions should be designed. The key differences between Class A/B and Class C/D lie in the size, mission life, and design complexity: Class A/B programs are the large "cannot fail," highly complex missions of significant national security which produce a satellite about the size of a school bus and are designed to last 15+ years, while Class C/D programs are more experimental, less complex, lower cost, more tolerant of risk-taking, and last 6 months up to 5 years.

Cost and Schedule are the main reasons a customer will select a Class C/D mission, because the customer doesn't have funding for a billion-dollar program and as a result they want to focus on a specific, smaller-scale mission. The Class C/D programs are more open to utilizing COTS components that in some cases have never flown in space previously. TOR-2011(8591)-21 was released in 2011 and was written to provide guidance on how to build a system to meet a specific mission to help ensure mission success, but it hasn't been updated since its release and therefore doesn't contain sufficient guidance for how to make the system cyber-secure, particularly with the modern proliferation of COTS products and specifically COTS software which are used to bring down the costs on these Class C/D missions. Lastly, large established

companies are struggling to build cheap Class C/D satellites because their decades of experience have entrenched costly robust processes tailored for Class A missions. To fill the market gap left by the large companies, many entrepreneurs are starting their own satellite companies and designing their satellite architecture from the ground up, and generally cybersecurity is not emphasized in the concerns of starting a new business. If Class C/D satellites and constellations are designed today with present-day cyberattack capabilities as well as protections against what the near- and far-reaching future of cyberattacks might hold, then their operators and customers can mitigate the expense and interruption of future cyberattacks, and the expense of retro-fitting space-based systems with security features once attacks on satellites become more widespread.

4.4 Current Industry Approach to Addressing Cybersecurity Vulnerabilities

Guidance for addressing cybersecurity vulnerabilities in satellite systems is generally limited to documents which highlight the need for more awareness [NIST SP 800-37, DoDI 5000.90, DoDI 8510.01] while other guidance is specific to widely-adopted IT systems [NPR 2810.1F, DoDI 8500.01] or certain terrestrial operational technology systems [CNSSI 1253 Space Overlay, NIST Commercial Space]. Relative to these applications, however, we have observed that the cybersecurity guidance for space is much less specific, making cybersecurity in space for Class C/D programs an emerging field due to the increasing and evolving importance of this domain [16].

The focus of Class C/D programs is on agility, and on quickly developing a functional system for a specific customer desired capability [142]. As a result, it is atypical for a C/D-type program to have strict cybersecurity requirements as part of their baseline requirements, because the customer is focused on the completion of the mission rather than on spending time considering what could go wrong relative to becoming the victim of a cyberattack in space [142].

We have observed a spectrum of approaches that industry has used to address cybersecurity, none of which are ideal in building a Class C/D cyber-secure system:

- **"Do Nothing" Approach:** Typically, we have observed that companies (many of which are the Class C/D satellite producers) do not consider provisions for a cyber-secure satellite beyond very basic standards such as communication encryption. Sometimes these companies are developing their own spacecraft and have no external stimulus driving them to address cybersecurity in their design. When the focus is on building a functional system for a specific purpose, where sometimes the mere act of launching is considered "mission success," the thought of protecting it from malicious actors isn't worth the time or effort [141].
- **Customer Defined / Requirement-Based Approach:** Customers can and do outline their own program-specific cybersecurity requirements. This is an improvement over the "Do Nothing" approach since the customer at least recognizes the importance of cyber. This is the ideal approach for a Class C/D mission since it defines cybersecurity requirements, without the perceived cost and schedule burden of a Class A/B process. However, to generalize, not all customers know what is important in building a cyber-secure system so this could provide a false sense of security by having a single requirement requiring communication encryption. i.e. "I did something so I'm protected." In many cases, even when they consider cyber-risk, companies are compromising on cybersecurity at the expedience of cost and schedule.
- **Standardized Framework Approach:** Some customers prefer to treat Class C/D missions as if they are just cheaper Class A/B missions. This leads to the implementation of costly and time-intensive cybersecurity assurance frameworks like Risk Management

Framework (RMF) on all satellite programs regardless of type classification. RMF and other prominent systematic frameworks aim to manage security risks effectively while maintaining an acceptable level of risk for both the system and its users.

The most thorough approach to addressing cybersecurity vulnerabilities for satellite programs is executing the NIST Risk Management Framework (RMF). RMF is a popular standardized approach to risk management which is available to industry, and is in fact considered the gold standard of risk management and mitigation. RMF is a comprehensive methodology that provides a framework for designers to address a wide spectrum of cybersecurity threats and determine mitigations on minimizing and/or accepting these risks. It offers a repeatable process for evaluating the security of satellite systems, ensuring compliance with established security standards and regulations like NIST 800-53, FIPS 199, or CNSSI No. 1253.

- NIST 800-53 contains comprehensive guidance for information technology systems, including over 1000 controls that cover all facets of IT system usage and operation. If one is not familiar with this standard it can be overwhelming to review and determine how to apply it to a program. This is where the RMF process comes into play: The framework provides these controls and allows security engineers to down-select them for the specific systems applicability [143].
- FIPS 199 provides a methodology to categorize Information Technology systems based on their Confidentiality, Integrity, and Availability [144]. Where TOR-2011(8591)-21 uses an A, B, C, or D scale to classify satellites, FIPS 199 classifies on a scale of Low,

Medium, or High Complexity. FIPS 199 is paired with NIST 800-53 to help tailor the 1000+ controls down to what is necessary for the IT system.

- CNSSI No. 1253 merges NIST 800-53 and FIPS 199 together. It provides a reference to help determine the appropriate controls in accordance with a system classification.

By assessing threats, vulnerabilities, and impacts, RMF seeks to enhance the overall security posture of satellite systems. In satellite design, RMF offers a robust process to evaluate security comprehensively. By considering a wide spectrum of threats and vulnerabilities, RMF helps minimize risks associated with satellite operations. Its systematic approach helps ensure satellites meet cybersecurity standards and regulatory requirements. RMF is composed of seven steps [145]:

- **Step 1: Prepare.** This step determines the business strategy to utilize RMF.
- **Step 2: Categorize Information Systems.** This step utilizes categorization derived from FIPS 199 to categorize the system somewhere between LLL to HHH.
- **Step 3: Select Security Controls.** This step is the time-intensive step to determine what controls to apply. Many modern Class C/D companies run lean. A challenge these companies encounter is how to review and determine what to apply from a control database which contains over 1000 potential controls, before it's too late to get the necessary controls incorporated in the satellite. In addition, the 1000+ controls were written for terrestrial IT systems and are not specific to satellite vehicles themselves. This step can run throughout the SRR, PDR, and CDR phases of a program [146].
- **Step 4: Implement Security Controls.** This step can run in parallel with Step 3. This is where the controls are applied to the systems.

- **Step 5: Assess Security Controls.** This step evaluates how well the controls are being implemented.
- **Step 6: Authorize Information System.** This step involves a buy-off from the customer or operating originations.
- **Step 7: Monitor Security Controls.** As the system is operated, the continuous monitoring of the control effectiveness is implemented.

While RMF is a valuable tool to manage perceived cybersecurity risk for Class A/B programs, it has some limitations when it comes to addressing the complex and evolving threat landscape for Class C/D satellites.

The traditional RMF process is a lengthy and compartmentalized process that does not enable interdisciplinary training and crossover between traditionally-trained satellite designers and cybersecurity engineers. Due to the traditional system compartmentalization in spacecraft design, most spacecraft engineers are experts in the specific aspects of their respective subsystems. However, few of these engineers have been trained to incorporate cybersecurity into their subsystem design. The converse is true of professionals entering into the still-emerging discipline of cybersecurity. These engineers have been trained to consider the cybersecurity aspects of system design without having been taught the specifics and difficulties of operating these systems in space. This lack of interdisciplinary training lends itself to an RMF process which, while looking complete on paper, is often the result of engineers who are inexperienced in cybersecurity checking off RMF boxes related to cybersecurity without truly understanding their implications to the specific program. Our observations of industry implementation of RMF at various companies indicates a level of "checkbox fatigue" in the engineers who are charged with implementing RMF, leading to an attitude towards system design which is driven more by

satisfying checklists and process implementation requirements, rather than a deep understanding of the satellite architecture and the relevant vulnerabilities.

RMF's comprehensiveness is also a weakness when it comes to the low-magnitude budget and design effort associated with Class C/D spacecraft. These companies are forced to either use overly complex processes such as RMF, or invent their own processes for satellite design. Because RMF is costly, time consuming, and complex, many companies opt to invent their own frameworks, or else they largely ignore cybersecurity, resulting in a wild-west satellite design ecosystem in which the physical satellite is vulnerable to cyberattacks. Designing cybersecurity into a system such as a satellite is complex because cybersecurity functions span the entire satellite architecture [33].

In practice, RMF usually results in two parallel efforts of cybersecurity implementation which end up competing with each other rather than complementing each other. The first effort involves the company's attempt to build a Class C/D satellite to satisfy a specific mission objective, wherein the design team is focused on building a functional product to satisfy the mission. The second effort is the security team putting together the check list on how the program plans to be complaint to RMF. Due to the competitive nature of Class C/D missions, by the time the security team gets to Step 3 in defining the specific security requirements, more often than not the satellite design has progressed to a point where changes will start to impact cost and schedule. This feeds back into the RMF for evaluating risk, taking into account potential impact for the mission. Since Class C/D programs are typically lower-cost and shorter-duration, the team is willing to accept only applying minimal protection on the satellite, i.e. encryption, and building a compliant ground system. When faced with 100 checkboxes, only one or two of which might actually be applicable to this particular Class C/D satellite, it is easy to gloss over or

miss the important 1-2% in favor of turning in a list that states that everything was “considered” – especially when an engineer has only been allotted a couple of minutes per checkbox for the analysis. This approach therefore focuses heavily on compliance requirements rather than a comprehensive understanding of the system's vulnerabilities, leading to compliance from a checklist perspective, while gaps still exist in compartmentalized portions of the system. This used to be a non-issue because satellites and their ground stations operated in isolation. Now that satellites are able to communicate with one another, however, the threat have multiple avenues that they can use to compromise the satellites.

4.4.1 Case Study – Investigating How NASA Defines Cybersecurity

Even prominent agencies such as NASA are still overcoming the rising challenge posed by proliferated LEO and their existing satellite systems. There has been an increase in cyberattacks on NASA’s IT systems alone, most of which are not set up appropriately to optimally mitigate the attacks (i.e. lack of agency-wide RMF, weak internal security controls, security POCs lack visibility/authority across programs) [147]. In 2019 a NASA Inspector General report determined that over the years NASA has experienced numerous cyber intrusions, including cases where adversaries have either stolen mass amounts of sensitive data or penetrated key servers required for satellite operation [148].

NASA has limited agency-wide design standards available for developing spacecraft cybersecurity. Because it is difficult to analyze how NASA does business behind closed doors, we reviewed the standards which NASA has made publicly available on https://standards.nasa.gov/, as well a publicly available Request for Proposal (RFP). During this

review, we found only a few standards that provide design recommendations which are related to a spacecraft's cybersecurity.

- **NPR 2810.1F NASA Information Security Policy** [149]**:**
    - Written for IT system security.
    - References NIST SP 800-37 RMF process, NIST SP 800-53 security controls, and various other NIST & FIPS documentation.
    - References NASA-STD-1006 [150] and NPR 7150.2 [151] for secure system and software engineering.
- **NASA-HDBK-1005- NASA Space Mission Architecture Framework (SMAF) Handbook For Uncrewed Space Missions** [152]**:**
    - Recommends a Cybersecurity design viewpoint for sensitive missions.
    - Reference NASA-STD-1006 [150] and NPR 7150.2 [151] for cyber guidance/requirements.
- **NPR 7150.2D NASA Software Engineering Requirements** [151]**:**
    - Section 3.11 defines software cybersecurity requirements / practices.
    - Software defects create security vulnerabilities. Need high quality "secure" coding methods and test practices to minimize bugs/deployment errors.
    - Need the ability to triage intrusions, rapidly assess threats, and update software.
    - COTS should be evaluated for vulnerabilities.
    - Points to NASA-STD-1006 [150] for preventing unauthorized access.
    - Software needs to be tested for cyber vulnerabilities.
- **NASA-STD-1006 Space System Protection Standard** [150]**:**

- Created to start standardizing cyber protection across NASA programs so they are more resilient to potential threats. Focuses on three key aspects:
  - Maintaining Command Authority
    - Systems need to prevent unauthorized access to ensure data integrity/positive control.
    - Space system should use encryption.
    - Recommends a "defense-in-depth" with both encryption and authentication. This might depend on a backup link that is more open, depending on anomalies.
  - Ensuring GPS Resilience
    - If a system requires GPS to operate, it should have backup systems that can fly the system in the event of GPS interference, and/or ways to detect / circumvent a spoofed signal.
  - Reporting Unknown Interference
    - If there is an attempt to access without authorization or to disrupt Guidance/Navigations, the Spacecraft should log the attempt, downlink, and then the program should alert other programs.

- **NASA-STD-8739.8B Software Assurance and Software Safety Standard** [153]**:**
  - Compliments NPR 7150.2D [151] by highlighting the assurance process with the software requirements.
  - Appendix A discusses additional considerations for common software errors. One of the software errors is listed as "Security and Virus Errors" and any of the

following could be considered the cause of the error. However, this document does not define how to detect or mitigate the causes [153]:

- 1. Denial or interruption of service
- 2. Spoofed or jammed inputs
- 3. Missing capabilities to detect insider threat activities
- 4. Inadvertent or intentional memory modification
- 5. Inadvertent or unplanned mode transition
- 6. Missing software error handling or detect handling
- 7. Unsolicited command
- 8. Stack-based buffer overflows
- 9. Heap-based attacks
- 10. Cybersecurity vulnerability or computer virus
- 11. Inadvertent access to ground system software
- 12. Destruct commands incorrectly allowed in a hands-off zone
- 13. Communication to/from an unexpected system on the network

Having evaluated the available NASA guidance documentation, we searched on SAM.GOV, a popular website for government RFPs, to see if we could evaluate proposal documentation. We found an inactive RFP for Geostationary Extended Observations Spacecraft Solicitation [154]. The program is a follow-on to the popular Geostationary Operational Environmental Satellites (GOES) [155]. Below in **Table 5** we evaluated each of the proposal documents and determined if they contained guidance for cybersecurity.

Table 5: Geostationary Extended Observations Spacecraft Proposal Cybersecurity Evaluation

Proposal Document [155]	**Cybersecurity Guidance**
RFP 80GSFC22R009 Final	• Contract documentation. • Points to Attachment L as a contractual deliverable for IT Security. Doesn't define cybersecurity for the satellite.
Attachment A- SOW • Program Statement of Work.	• Doesn't include anything for cybersecurity or security. • Applicable Documents don't reference any cybersecurity documents.
Attachment B- PSPEC • Functional and Performance Specification.	• Section 3.4.2.3 defines command security. Only discusses command encryption and FIPS 140-3. • Sections 3.7.1.0-11 and 3.8.1.0-9: Each section contains one requirement pointing to NIST-SP-800-53 for ground IT security. • Applicable Documents reference: o Federal Information Processing Standards Publication 140-3, Security Requirements for Cryptographic Modules, March 22, 2019. o NIST Special Publication 800-53, Rev. 5, Security and Privacy Controls for Information Systems and Organizations, September 2020.
Attachment C – GIRD • General Interface Requirements Document.	• Doesn't include anything for cybersecurity or security. • Applicable Documents don't reference any cybersecurity documents.
Attachment D – PRAD • Payload Resources Allocation Document.	• Doesn't include anything for cybersecurity or security. • Applicable Documents don't reference any cybersecurity documents.
Attachment E-1 GXIUIID • Imager Unique Instrument Interface Document	• Doesn't include anything for cybersecurity or security. • Applicable Documents don't reference any cybersecurity documents.
Attachment E-2 LMXUIID • Lightning Mapper Unique Instrument Interface Document	• Doesn't include anything for cybersecurity or security. • Applicable Documents don't reference any cybersecurity documents.
Attachment E-3 GXSUIID • Sounder Unique Instrument Interface Document	• Doesn't include anything for cybersecurity or security. • Applicable Documents don't reference any cybersecurity documents.
Attachment E-4 OCXUIID • Ocean Color Instrument Unique Instrument Interface Document	• Doesn't include anything for cybersecurity or security. • Applicable Documents don't reference any cybersecurity documents.
Attachment E-5 ACXUIID • Atmospheric Color Instrument Unique Instrument Interface Document	• Doesn't include anything for cybersecurity or security. • Applicable Documents don't reference any cybersecurity documents.
Attachment E-6 -ABIUIID • Advanced Baseline Imager Unique Instrument Interface Document	• Doesn't include anything for cybersecurity or security. • Applicable Documents don't reference any cybersecurity documents.
Attachment F- IRD-SS-DCS • SS to Data Collection System Interface Requirements Document	• Doesn't include anything for cybersecurity or security. • Applicable Documents don't reference any cybersecurity documents.
Attachment G- IRD- SS-C3S	• One requirement in Section 5.3.11.0-1 for the command receiver to comply with NASA-STD-1006.

• SS to Ground Located Command, Control, & Communications Interface Requirements Document	• Applicable Documents reference: o Space System Protection Standard, NASA-STD-1006, Baseline, October 29, 2019.
Attachment H -SCMAR • Mission Assurance Requirements	• Section 5 defines Software Assurance requirements. Mostly pertains to how the spacecraft software will be developed and tested to minimize bugs. Section 5.2.0-1 has a requirement pointing to NASA-STD-8739.8A as guidance to develop a software development plan. • Applicable Documents reference: o NASA-STD-8739.8A NASA Standard for Software Assurance.
Attachment I- Radiation Environment for Electronic Devices • Radiation Environment for Electronic Devices	• Doesn't include anything for cybersecurity or security. • Applicable Documents don't reference any cybersecurity documents.
Attachment J- GRDDP • Reliable Data Delivery Protocol	• Doesn't include anything for cybersecurity or security. • Applicable Documents don't reference any cybersecurity documents.
Attachment K-CONOPS-0004 draft • Concept of Operations	• Doesn't include anything for cybersecurity or security. • Applicable Documents don't reference any cybersecurity documents.
Attachment L- IT Security Management Plan	• 1 Page TBD document. Contractor is supposed to submit this as a deliverable to NASA.
Attachment M- App Doc List Final	• Summary of applicable documents for IT Security. • Doesn't define how each document is applicable to the IT system. Additionally, nothing specific to the satellite design.
Attachment N-TAM_Table_R	• Doesn't include anything for cybersecurity or security. • Contains a table for environmental testing.
Attachment O MEL	• Doesn't include anything for cybersecurity or security. • Contains a table for a Master Equipment List template.
Enclosure 1-IT Security Management Plan Template (1)	• Template for the Security Management Plan to be used for Attachment L. • References many industry standards for IT security. Doesn't define how a program should use or implement the referenced IT security standards. Additionally, many of the reference documents are part of the contractual documents so they might not truly be applicable for the company that won the contract.

Having reviewed the NASA proposal documents in **Table *5*** we can now draw some conclusions as to how NASA is defining cybersecurity for large programs.

- The various documents point to a list of IT Security documents for the IT systems that will be used to design and operate the system.
- The only cybersecurity requirements directly applicable to the spacecraft are:
  - Command Encryption

- NASA-STD-1006 [150]
    - Maintain command authority to prevent unauthorized access
    - Resiliency to operate if GPS is jammed or spoofed
    - Log unknown interference to disrupt guidance systems
- NASA-STD-8739.8 [153]
    - Defines considerations to help identify software hazards, some of which could be attributed to cybersecurity vulnerabilities

Largely as we have previously stated, the focus of cybersecurity is on IT systems and not on the spacecraft itself, outside of having requirements for generalized encryption.

4.5 New Strategies for Cybersecurity of C/D missions

Based on the understanding of the field, this research question asserts that space system manufacturers need to emphasize the importance of "Mission Cybersecurity" when designing their systems. Mission Cybersecurity focuses on the application of cybersecurity protection directly on the OT mission system, i.e. satellite, instead of following the traditional cybersecurity approach of securing the (typically terrestrial) computer IT infrastructure. This is an important delineation because traditional cybersecurity protections have been scarcely applied directly to Class C/D satellites, essentially leaving a significant cyber-vulnerability that threat actors could compromise. To support this assertion, we will contrast the current industrial approach to addressing cybersecurity vulnerabilities with our proposed modernized and systems-engineering-driven approach to satellite system cybersecurity. This research then discusses the strengths and weaknesses of the proposed approach, and draws conclusions about the potential for implementation in the technical and policy environment of the present.

A holistic strategy for Class C/D programs that utilize an RMF-like cybersecurity assurance framework is necessary for engineers to comprehend and recognize the cybersecurity interfaces of the system during the design phase. Engineers and stakeholders need to be enabled to proactively identify and understand the cybersecurity interfaces, dependencies, and potential threats associated with each component early in the life cycle. Mission Cybersecurity needs to be broken down in such a way that satellite designers can easily understand what it means. This would allow for a better integration of cybersecurity principles and a more profound understanding of potential vulnerabilities. The most effective method is to define design guidance and design requirements. Up until now and particularly under RMF, the cybersecurity defense of a satellite has been treated as a subset of flight software, when in fact it permeates the entire system and cannot be divided out into any other subsystem. Designers will need to pivot to thinking of cybersecurity as a spacecraft subsystem. This will force thinking of subsystem-to-subsystem interfaces and the overall impact of cybersecurity on the system as a whole. Potential attack behaviors must instead be the drivers behind hardening and securing the product. For these smaller satellites, there is not a "one size fits all" approach to risk mitigation. There are many approaches to implementing cybersecurity, and this research is recommending an approach to raise awareness and incorporate minimal best practices.

Modern small satellite constellations with hundreds of satellites that work together but are relatively interchangeable are becoming more and more common. The operator of such a constellation is going to be less concerned about a few individuals getting hacked than they would be if every single satellite had a unique job to perform to make the constellation operate. Likewise, the loss of a few small cheap satellites with a one-year lifespan would be less important to a company than the loss of a 15-year satellite. As Class C/D satellite constellations

grow and diversify, the guidelines for protecting them from cyberattacks must evolve as well. The industry needs levels of protection that correspond with mission class / cost / design life. Industry needs to consider cybersecurity requirements upfront in their design prior to SRR. In addition, these requirements need to be achievable and must be easily interpreted by spacecraft designers. By starting with a defined baseline of cybersecurity design principles and requirements, the specific programs can still tailor the process to specific mission parameters.

4.6 Defining a Cyber-Secure Satellite Architecture through Requirements

Satellite manufacturers for Class C/D satellites will benefit from a simplified set of cybersecurity requirements to help guide their design. These principles need to be simplified such that typical spacecraft designers, who generally do not have cybersecurity expertise, can relate to them and understand their significance to the satellite architecture. There are numerous guides and cybersecurity principles available in publications, but they have been largely written for IT systems and may not be applicable to satellite and space systems. In addition, as determined by the Government Accountability Office (GAO) during one of their own audits, Cybersecurity requirements need to be defined as part of the original contract, or they will not be incorporated [26].

In response to the challenges described above, this research seeks to synthesize knowledge from industry experts, lessons learned we have observed from multiple satellite programs, and industry guidance(such as CNSSI 1253 and its associated overlays) to present the following requirements to achieve a cyber-secure architecture for a Class C/D LEO satellite. We propose the set of requirements defined in **Table 6** to provide guidance to satellite companies so that cybersecurity is built up-front into the architecture of the satellite.

Table 6: Class C/D Cyber-Secure Architecture Requirements

Identifier	Requirements	Rationale
L2-SS-01	The Satellite **shall** utilize a secure boot methodology for loading flight software.	The satellite should utilize modern avionics. Most modern satellite avionics utilize modern processors that have secure boot capability. Secure boot involves digitally signing the Flight Software with a cryptographic key built into the avionics [156].
L2-SS-02	The Satellite **shall** utilize commercial encryption, or equivalent, for all external communication data links.	The satellite should utilize commercial encryption. Many commercial satellite encryptors are readily available, and it is also possible to build encryption into their operating software. CNSA is the current state-of-the-art encryption for commercial systems [157]. It has superseded the widely adopted NSA Suite B; however, this requirement includes "or equivalent" to enable affordable options.
L2-SS-03	The Satellite **shall** utilize a communications protocol that includes built-in cybersecurity protections.	CCSDS is one of the most popular communications protocols for satellites. Modern CCSDS includes many cybersecurity features as part of the protocol, such as integrity validation of the data transmitted, authenticity validation of the data source, and anti-replay [158]. This requirement calls for the use of these defined features. In addition, depending on the communication protocol, it is possible to have encryption built into the protocol [159].
L2-SS-04	The Satellite **shall** incorporate a cyber-secure "gold copy" of flight software that can be loaded in the event of a critical cyber fault.	On most satellites, it is standard practice to have multiple boot copies of flight software. Under this requirement, one of these boot copies should be cyber-secure and can be defaulted to in the event of a critical cyber fault. This will result in the ability for the satellite to load itself into a known cyber-safe state.
L2-SS-05	The Satellite **shall** validate the authenticity of uploaded software prior to accepting the uploaded data into on-board storage.	The satellite should check that all software is authentic and from the authorized operator sending the data prior to allowing it to load onto the satellite.
L2-SS-06	The Satellite **shall** validate the integrity of uploaded software prior to loading.	The satellite should check that all software is properly formatted prior to allowing it to execute onto the satellite.
L2-SS-07	The Satellite **shall** have an adaptable cybersecurity policy for all mission modes.	Many satellites have different operational modes (i.e. Nominal Ops, Initialization, Safe Mode, End of Life). The cyber solution needs to account for these system operational mode changes. In addition, this will ensure the satellite incorporates cyber-safe counter measures if integrated with a fault management response.
L2-SS-08	The Satellite **shall** detect cyber events on the satellite.	This requirement is intentionally open-ended. Satellite manufacturers should conduct threat analysis to determine the appropriate threats to protect against. As part of this requirement, the satellite manufacturer should determine specific commands or telemetry points that could be used to determine if or when a cyber event has happened. Examples are jamming attempts, loss of signal, unauthorized access, replay attempts, command accepts/rejects, and planned ground contact windows [37].
L2-SS-09	The Satellite **shall** report cyber events on-board the satellite to ground operators.	When a cyber event is detected, the satellite needs to alert operators with the event and forensics.

The requirements in **Table 6** show the set of requirements that, if implemented, can guide the systems engineering design process for Class C/D type satellites to effectively incorporate cybersecurity. These requirements encompass the satellite as a whole from the top level, providing guidance for requirements decomposition. From a system perspective, this drives what components are selected throughout the architecture, the external satellite interfaces, and satellite cybersecurity behavior. **Table 7** shows how the requirements from **Table 6** help detect, mitigate, and report the cybersecurity threats defined in Research Question 1.

Table 7: Cyber-Secure Requirements to Cybersecurity Threats

Cyber-Secure Requirement Identifier	Cybersecurity Threats from RQ 1 Mitigated							Rationale
	Denial-of-Service (DoS)	Masquerade	Unauthorized Access	Replay	Software Threats	Tainted Hardware Components	Jamming	
L2-SS-01			X		X			Secure boot requires a trusted key paired between hardware/software. Only trusted users would have access to this key to create, modify, & upload operating software.
L2-SS-02	X	X	X	X				By utilizing encryption, the satellite will only be able to communicate with other trusted satellites, or ground stations, that have matching encryption keys.
L2-SS-03	X	X	X	X				Similar to encryption, modern communication protocols have built in cybersecurity checks to ensure they are only able to communicate with other trusted satellites or ground stations.
L2-SS-04					X	X		The intent of a cyber-secure version of flight software is for the satellite to recover from a cybersecurity attack into a known safe state.
L2-SS-05		X	X		X			By validating the authenticity of uploaded software, we can ensure it is authentic and from the authorized operator sending the data prior to allowing it to load onto the satellite.
L2-SS-06		X	X		X			By validating the integrity of uploaded software, we can ensure all software is properly formatted prior to allowing it to execute onto the satellite.
L2-SS-07	X	X	X	X	X	X	X	This requirement is for an adaptable cybersecurity policy that could be designed to mitigate all these threats.
L2-SS-08	X	X	X	X	X	X	X	This requirement is for the ability to detect cybersecurity threat that could be designed to mitigate all these threats.
L2-SS-09	X	X	X	X	X	X	X	This requirement is for the ability to downlink cybersecurity logs that could be designed to encompass all these threats.

4.7 Cyber-Secure Satellite Architecture Requirements Discussion

We now reflect on the importance of Mission Cybersecurity and the requirements identified in **Table 6** and discuss the future impacts of those requirements on the systems engineering design process for satellite programs.

4.7.1 Mission Cybersecurity Viewpoint

A key component/attribute of the requirements detailed in **Table 6** is that they provide a Mission Cybersecurity viewpoint for satellite cybersecurity. These Mission Cybersecurity requirements are important for modern satellite development programs because they emphasize a holistic and overarching view of the importance of cybersecurity protections from the inception of the program through the design process and culminating with the operational lifespan of the satellite. An important mindset is that the satellite is the primary system for accomplishing the mission, and is the primary reason the company or customer is spending money. Typically, cyber protections are placed on ground systems because this is easier to accomplish, which leads to potential vulnerabilities on the satellite itself.

In our experience cybersecurity protections directly on the satellite are usually lacking. Incorporating cybersecurity practices from the earliest stages of an acquisition is easier, less costly, and more effective than trying to add, or "bolt on," cybersecurity protections late in the development cycle or after a system is fielded [26]. Ground systems on Class C/D missions typically rely on established ground components such as antennas services purchased through long-trusted vendors such as KSAT [160] and their design and software code writing timeline is therefore more flexible. The architectural design of the satellite vehicle itself must be finalized much earlier on in the design timeline to enable the sourcing and purchasing of specialized

components. Specialized satellite components must be ordered much earlier on in the process to allow for enough time for extensive integration and test of the space vehicle. Unlike ground station components, the satellite components are not easily repaired or replaced after mission launch. The requirements in **Table 6** have been developed to draw companies into a mindset of full Mission Cybersecurity from the inception of their satellite programs, including the satellite and the ground station.

The concept of the mission cybersecurity viewpoint is to change how we view satellites and their cybersecurity vulnerabilities. As explained in Research Question 1, similar to terrestrial Operational Technology (OT) systems such as an electric power plant, these systems that were once secure through obscurity are now becoming targets. It is no long a matter of "if" but "when" a critical satellite or constellation will be compromised. In addition, proliferated LEO and proliferated ground stations (i.e. commercial ground) has led to thinly-stretched security which malicious actors take advantage of, gaining new threats just by the fact that there are more avenues through which to compromise vulnerable systems.

4.7.2 Incorporation into a Systems Engineering Process

The set of proposed requirements in **Table 6** are the basis on which a Mission Cybersecurity cyber-secure systems engineering and development process can be built, so that cybersecurity is integrated into the design process upfront during the design process. By recommending cybersecurity requirements for the satellite, we are ensuring that cyberdefense is applied at the beginning of the design process and will also be tested and verified before flight, regardless of whether a company is following an RMF process or not. Widespread adoption of the requirements in **Table 6** will help ensure that satellite manufacturers are poised to address cybersecurity vulnerabilities and common concerns from the beginning of the design process,

thus building proactive protections into the design rather than relying on reactive post-threat patches or recovery attempts. In regards to the standard Systems Engineering Vee [161] these requirements would be injected at the start of the program for a requirement baseline at the program SRR. This is important because adding cybersecurity later in the program lifecycle acts as a band aid when it's too late to impose effective and significant change on the program and gives a false sense of security, as demonstrated by GAO during some of their routine DOD evaluations [26].

These discrete requirements raise awareness to the systems engineering and design teams that these capabilities are necessary and helps to minimize the siloing of the teams as discussed **Section 4.4.** This also drives the need for programs to conduct more cybersecurity testing as part of the deliverable design process to test against the negatives which the GAO also found to be lacking [26]. One of the most powerful benefits of including cybersecurity in an integrated systems engineering architecture is that these requirements become part of the design baseline which will ensure that the ability of the satellite to withstand cyber threats, penetration, and other forms of cyberattacks will be tested and verified as part of the final verification process and as part of completing the Systems Engineering Vee alongside all of the other non-cybersecurity design requirements [162]. The temptation to merely "check the cybersecurity box" will be averted, and the test team will have to include experts in cybersecurity to ensure the passage of the cyber requirements, thus strengthening the overall program and the likelihood of mission success. Finally, this will champion incorporating a cybersecurity test and evaluation mindset into the penetration testing of the overall system. Cyber T&E adds the testing viewpoint that the satellite cannot be compromised from cyber hacking [163].

4.7.3 Road Map to Incorporation

When it comes to cybersecurity, it is possible to design much more complicated protections and interventions, but for Class C/D satellites the goal is to simply start thinking and incorporate a new mindset, from which expansion is possible. Therefore, the requirements in **Table 6** are a recommended list of requirements which are deliberately written to be unintimidating, achievable, generic, and easy to comprehend for a Class C/D program. As a result, and to provide a stepping stone for companies to adopt cybersecurity, these requirements intentionally do not dictate the specifics for implementation. Instead, companies will customize these roadmap requirements to serve their specific purposes and needs by adding additional cybersecurity protections such as user accounts, access levels, data restriction, multifactor authentication, data retention policies, zero trust, and COTS firmware vulnerability assessments. The specifics for performance will be left up to programs as they evaluate threats.

As part of expanding on the requirements in **Table *6*,** manufacturers need to adopt a roadmap to continually improve their cybersecurity posture. The requirements in **Table 6** were intentionally written to be achievable and we recommend the following roadmap:

- Step 1) Incorporate the requirements from **Table 6** into the beginning of the design process for an SRR requirements baseline.
- Step 2) Conduct threat analysis to define specific threats to incorporate into lower level requirements for a PDR requirements baseline. Initially this does not need to be exhaustive, and will enable the company to learn about potential threats.
- Step 3) Design the satellite and plan how to verify the satellite and subsystem requirements for a CDR requirements baseline.

- Step 4) Conduct Cyber penetration testing to verify the requirements. Start first on satellite simulators or Hardware In The Loop systems prior to the programs TRR. Then, as part of the test campaign, plan penetration testing on the system.
- Step 5) During operation or on future missions, improve the requirements and threat mitigation strategies by adding advanced protection capabilities such as:
    - Zero Trust Architecture [164].
    - Intrusion Prevention Systems [165].

4.7.4 Example of Expansion of Satellite Requirements

The intent of the requirements in **Table 6** is to be a first step in having the capability to detect and protect the satellite's cyberattack surface. These requirements focus on the ability to authenticate information first, then detect potential intrusion, and lastly alert the operators of such intrusion.

Whether they intentionally define these as requirements or not, satellite manufacturers typically build their systems to utilize communication encryption (L2-SS-02), gold copies of flight software (L2-SS-04), and software CRC integrity checking (L2-SS-06). Modern hardware and software practices incorporating techniques such as secure boot (L2-SS-01), cyber-secure communication protocols (L2-SS-03), software authenticity validation (L2-SS-05) should be easy to incorporate into their architecture. The remaining requirements: to have an adaptable cyber policy (L2-SS-07), ability to detect defined cyber events (L2-SS-08), and report detected events (L2-SS-09) are novel and define a gap inherent to industry as explained throughout this section. The key requirements L2-SS-07 and L2-SS-08 are intentionally left open-ended since as part of the integration and decomposition effort the specific cyber threats need to be evaluated by

the specific satellite manufacturer to determine what is considered a cyber threat and merits the design and test effort to mitigate. However, incorporating these as requirements from the beginning will encourage the conversation to determine what is a cyber threat and encourage the resolution on how to appropriately cyber-test the satellite to demonstrate that it is cyber-secure. As companies incorporate these baseline cyber-secure requirements into their design, they can further expand and decompose them as they perform threat analysis to define the specific threats to their system(s).

The set of requirements in **Table 6** can be expanded to include attributes that complete a requirements verification cross matrix (RVCM). They can be further decomposed to the subsystem level to define how a subsystem needs to function to meet the satellite level requirement. An example of a requirement with several key attributes and potential decomposition is shown below in **Table 8**.

Table 8: Example RVCM & Decomposition

ID	Short Text	Requirement	Verification Method	Verification Approach
L2-SS-08	Cyber Event Detection	The Satellite shall detect cyber events on the satellite.	Test	The satellite will undergo cyber penetration testing. The cyber-policy for the various mission modes will be exercised against various cyber threats.
L3-CYB-01	False Command Alert	The Cybersecurity Subsystem shall alert when satellite commands are not identified in the on-board command/telemetry database.	Test	The satellite will undergo cyber penetration testing. The cyber-policy for the various mission modes will be exercised against various cyber threats.
L3-CYB-02	Malicious Command Alert	The Cybersecurity Subsystem shall alert when satellite commands are determined to be malicious in accordance with the cybersecurity policy.	Test	The satellite will undergo cyber penetration testing. The cyber-policy for the various mission modes will be exercised against various cyber threats.
L3-CYB-03	Invalid Command Alert	The Cybersecurity Subsystem shall alert when satellite commands are determined to be invalid in accordance with the cybersecurity policy.	Test	The satellite will undergo cyber penetration testing. The cyber-policy for the various mission modes will be exercised against various cyber threats.
L3-CYB-04	Malicious On-Board Telemetry Alerting	The Cybersecurity Subsystem shall alert when on-board component commanding is invalid in accordance with the cybersecurity policy.	Test	The satellite will undergo cyber penetration testing. The cyber-policy for the various mission modes will be exercised against various cyber threats.
L3-CYB-05	Anomalous On-Board Telemetry Alerting	The Cybersecurity Subsystem shall alert when satellite telemetry is not as identified in the on-board command/telemetry database.	Test	The satellite will undergo cyber penetration testing. The cyber-policy for the various mission modes will be exercised against various cyber threats.
L3-CYB-06	Invalid On-Board Telemetry Alerting	The Cybersecurity Subsystem shall alert when on-board component telemetry is invalid in accordance with the cybersecurity policy.	Test	The satellite will undergo cyber penetration testing. The cyber-policy for the various mission modes will be exercised against various cyber threats.

4.8 Research Question 2 Conclusion

Upon completion of this research, we have answered Research Question 2: Define a cyber-secure architecture for a candidate LEO satellite.

This research emphasized that the threat of cyberattacks against satellites grows in tandem with the increasing use of satellites. Cybersecurity vulnerabilities in satellites can result in the loss of critical data, disruption of services, failure to complete the mission, and even loss of the satellite itself. Therefore, manufacturers and operators must incorporate cybersecurity practices into satellite designs from the beginning, starting with requirements development. To define a cyber-secure architecture, we defined the importance of Mission Cybersecurity for successfully completing a satellite mission, recommended high-level requirements that meet the extensible need for Class C/D space systems and described the rationale behind these recommendations, and encouraged companies building Class C/D satellites to tailor these requirements to their own needs in order to best ensure the mission successes of their individual programs and satellites. Through the use of the requirements defined by this research question a Class C/D satellite can start off with a cyber-secure architecture, and as companies continue to improve their cybersecurity hygiene, they can continue to improve their cyber-secure architecture.

Research Question 3 will demonstrate how an MBSE cyber-secure satellite architecting process preserves the benefits of using MBSE as a systems engineering design process, as well as generating a template MBSE cyber-secure architectural methodology which could be referenced as a starting point by a company as it embarks on a Class C/D satellite program.

Chapter 5.

Research Question 3: An Evaluation on how an MBSE Cyber-Secure Satellite Architecting Process Preserves the Benefits of Utilizing MBSE

5.1 Research Question 3 Introduction

This chapter addresses Research Question 3, which is restated as follows: Evaluate how an MBSE cyber-secure satellite architecting process preserves the benefits of utilizing MBSE. To answer this research question, two tasks were laid out.

Task 3.1: Perform a literature review to characterize how the industry has applied MBSE to model cyber-secure architectures. To answer Task 3.1, we evaluated literature to define how MBSE has previously been applied to IT and OT system architectures. In addition, we evaluated common cybersecurity industry tools to determine commonalties and how they are integrated into a system architecture process. Based on this research we were able to gain an understanding for how MBSE and cyber tools have been applied to model a cyber-secure system.

Task 3.2: Conduct MBSE modeling to define a candidate cyber-secure architecting process for a LEO satellite and evaluate if the model preserves the cost/schedule benefits of utilizing MBSE. To answer Task 3.2, we leveraged our industry experience to build an MBSE model of a LEO satellite, proposing a methodology for how cybersecurity could be integrated into an MBSE architecture process to synthesize a cyber-secure architecture process. This task

integrated the findings from Research Questions 1 and 2. Finally, this task examined industry research to evaluate the cost benefits of applying MBSE.

The research seeks to address this question by leveraging our industry experience and an evaluation of industry generated sources.

5.2 The Benefits of an MBSE Architecture Approach

The traditional systems engineering design process, where information is housed in separate comprehensive documents, has proven to be not effective in developing towards emergent properties of a total system, such as integrating cybersecurity protection into a satellite [166]. The traditional process can introduce errors (of omission, of specification, or of integration) into the overall system design process. These errors can lead to a complex and time-consuming design process that may result in designers lacking the ability to fully envision the big picture or the vulnerabilities embedded in the overall system [167]. Traditional systems engineering processes pose challenges to being migrated into a digital application. Due to the growing complexity of space system projects, the traditional (non-MBSE) design approach and tools are having a harder and harder time supporting the growing customer demand for faster schedules and cheaper products.

The U.S. Government Accountability Office (GAO) routinely audits government programs for accountability, integrity, and reliability [168]. As part of a recent annual audit of the defense industry, the GAO determined that “programs’ average delay in delivering initial capabilities has increased by over 27 months since their first full estimates. Further, MDAPs [Major Defense Acquisition Programs] show, on average, 51 percent total acquisition cost increase since their first full estimates” [169]. Model-Based Systems Engineering (MBSE) may be a solution to help

alleviate some of the cost increases and inaccurate schedule planning that has so far been inherent to space industry. As system design complexity continues to grow, the ability of traditional tools & methods to design these systems is getting weaker. MBSE can be used upfront in the design process to conceptualize the system and then throughout the lifecycle to test, verify, and validate that the system meets the needs of the stakeholders [170]. MBSE is a different way to approach designing a system, one which enables designers to model and design systems of higher complexity and integration through the structured application of digital models and tools. MBSE makes an integrated system model the primary Single Source of Truth (SSOT) artifact for systems engineering activities. This model is then utilized to model a complex system and its interfaces digitally to perform early architecting, verification, and validation of a system [24]. MBSE enables designers to have greater architecture visualization, reusability, agility, and information sharing when applied to a system architecting process. By interconnecting these engineering documents digitally, all of the interdependencies and interfaces can be logically linked in one engineering location as an SSOT [171]. As a result, modeling the system digitally can enable short development cycles to rapidly develop products which can result in a cost and schedule savings for the company using MBSE. As such MBSE is the future for satellite engineering design process, and the notion that it can decrease design complexity can potentially enable space system companies to adopt MBSE to meet the customer demand for faster and cheaper programs.

When applied to cybersecurity, the MBSE model can act as an SSOT to provide a complete picture of all the system interfaces and functionality in one location. This enables cybersecurity designers to create security-centric viewpoints across the entire design to understand where the vulnerabilities are in the design and to minimize the chance of missing

critical interface information housed in separate disconnected artifacts. In addition, an MBSE model enables cybersecurity engineers to define cybersecurity centric requirements, identify where in the architecture requirements are verified, outline the security boundaries, explain the security controls necessary within each security boundary, and model defense-in-depth or zero trust techniques [25]. Due to the novel nature of both MBSE and cyber threats to satellites, there are limited publicly available examples of the application of MBSE to resolve cybersecurity challenges during the satellite design process.

On the basis of the above reflections on the state of the field, there is a need to evaluate the approach of adopting and tailoring MBSE to space systems cybersecurity engineering practices and determine how to maximize the return on investment when designing a cyber-secure satellite architecture. An examination of the industry application of MBSE and how industry has measured its effectiveness will lead to a more practical understanding of the benefits of MBSE. This research seeks to guide the development of cybersecurity for the emerging Class C/D satellite market by defining the core architecture artifacts so that these smaller projects know what they need to model and how to apply it to cybersecurity. The Class C/D space manufacturers can then leverage this agile framework as a starting point for generating their own MBSE models for their respective satellite programs and cybersecurity solutions, leading them to be able to integrate cybersecurity into their design process to ensure their own specific cybersecurity concerns are also validated and addressed throughout the design process. In short, this research aims to help space system manufacturers apply MBSE to their cybersecurity challenges for satellites.

5.3 The Challenges to Adopting an MBSE Architecture Approach

In an idealized implementation, a space system program would strictly follow the philosophy behind MBSE and all of the program engineers would utilize the same MBSE model as an SSOT. However, space system companies encounter significant challenges to attempts to change the traditional industry established process and to adopt MBSE instead [172]. The concept of utilizing MBSE as an SSOT is attractive in concept, but its implementation usually results in a drastic pivot in that every technical and management employee, even those who specialize in unique engineering tools, would need to learn a new tool and develop new processes for how to use and apply MBSE. A swift change would significantly disrupt business profit and schedule, so the consensus view of digital transformation is that a gradual shift must be encouraged [173].

Modern MBSE relies on a visual modeling language such as System Modeling Language (SysML). The MBSE systems engineers use blocks and logic expressions to model a system in addition to lists of written requirements, in the same way that some software programming languages use graphical blocks, icons, and symbols rather than alphanumeric words to code a program. Without a structured MBSE process, tool, and framework every MBSE systems engineer has their own unique way of implementing MBSE, making it difficult to standardize MBSE implementation. This impacts a model's extensibility and drives the need to create program-wide design guidance to instill modeling consistency. In the application of space system engineering, the customer is generally familiar with seeing certain traditional document-based engineering common products and design processes throughout a program life cycle and presented at design reviews (i.e. GANTT charts, detailed analysis reports, standalone interface control documents), most of which are not inherent to an MBSE model without significant

modeling effort. Lastly, we have observed that employees at space system companies often resist the MBSE culture change because they do not have a clear understanding or example of the value MBSE can provide over their traditional ways to stimulate their need to change.

Large and well-established space system companies, due to how they are organized and governed by their rigorous corporate guidance and contracted processes, have had a hard time adopting an MBSE approach because they are stuck in their traditional document-focused enterprises, processes, and contracts [174]. As a result, many programs continually run into the SE problems that MBSE is designed to alleviate. In our space systems engineering context MBSE can help in implementing Agile Systems Engineering methods [175]. MBSE models can also be used in spacecraft engineering processes to assess new components and evaluate behavioral and mission effects across products.

We have observed that typically space system companies keep traditional processes, procedures, and deliverables until they are forced to change due to customer direction. However, space system customers are beginning to hear about the benefits of MBSE and are interested in applying those benefits to the programs they are managing [176]. In order to maintain their competitive edge, it therefore behooves satellite manufacturing companies to learn to strategically implement MBSE at their leisure and under their own experimentation, without a contractual force [169] and subsequent scramble to adopt unfamiliar MBSE processes which may result in an unworkable and cumbersome MBSE process rather than a premeditated, elegant, and proposal-ready MBSE process [177].

5.4 Methods of MBSE which have previously been Applied to Engineering

Researchers have been developing roadmaps and real examples of MBSE application and assessment to demonstrate the costs and benefits of MBSE. MBSE research exists which demonstrates the value of replacing conventional SE tools [178] [179] [180], such as VISIO for creating diagrams and Microsoft Excel for creating analytic spreadsheets like a Mass Budget or Power Analysis. We have observed that during the process of adopting MBSE tools companies can mangle their adoption: Instead of creating a good working model, the organization creates a "model" that is used only to generate conventional document-based SE examples, such as an Excel list output [181] [182]. Of course, without stronger evidence of product or process improvement, companies do not understand the benefit of transitioning to MBSE, and instead often choose to continue under document-based SE paradigms.

The examples in the previous paragraph do not demonstrate the full extent of the benefits and capabilities of MBSE. For instance, MBSE not only allows for the recreation of a Microsoft Excel analysis – it also allows for the full integration of that analysis into all aspects of the modeling and product architecture. MBSE allows a systems engineer to model a mass budget by rolling up the weight of individual model blocks that represent each hardware component on the spacecraft and then feeding the compiled mass budget into other analyses such as the CONOPS, interfaces, and the software behavior. MBSE also allows for the automated rollout of component changes and property refinements. Continuing the mass budget example, if the actual mass of a component is different from the initial estimate or if a different component is inserted, the model needs to be updated in only one location – the original component block – and then the changes (and their overall effects) automatically cascade throughout the rest of the model [183].

To investigate MBSE's improvements over the legacy tools space system companies currently use, the engineering community has applied MBSE to system design in the following ways:

- Developing the concept of operations of a satellite to enhance design coordination and stakeholder feedback by modeling use cases, context, and system behavior [184].
- Functional validation of a spacecraft Command & Data Handling subsystem by modeling use cases, context, and system behavior [185].
- Mapping requirements and test cases against a system design [186].
- Managing and identifying system requirements [187].
- Conducting program technical reviews in lieu of a document-centric approach [188].
- Creating a digital twin of the physical system [189].
- Integrating test planning and artifacts to promote the organization and structure of test artifacts [190].

We have observed that all of these methods are important and a beneficial use when designing a space system in MBSE. This research question looks to demonstrate these processes and extend them with the novel application of modeling a cyber-secure space system architecture.

5.5 Current Landscape of Cyber Modeling Tools

There are many methodologies and tools available for conducting cybersecurity modeling and threat identification. These tools are typically used to model IT computer systems and their

cyber-boundaries. The following tools and methodologies can be leveraged to model cybersecurity for a satellite by treating the satellite like an IT computer system; however, these tools may be difficult for new cybersecurity practitioners to learn and use [191] and the modeling is disconnected from the satellite architecture and design artifacts used to build the satellite (as explained in **Section 4.4**.

Having surveyed the industry, a few popular methodologies and tools that were evaluated are:

- **Microsoft STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege).** The Microsoft STRIDE tool is free to use and great at modeling cyber-physical interfaces. The tool is similar to VISIO with a cybersecurity tool kit overlay where the user can model IT/OT networks and their interconnects, allowing the definition of the system users, security boundaries, and vulnerabilities [192].
- **PASTA (Process of Attack Simulation and Threat Analysis).** Developed by VerSprite security. PASTA provides a methodologic approach to evaluating the system and then determining the cybersecurity threats and analyzing the risk impact to the system. [193].
- **Attack Trees.** A formal methodology to evaluate cybersecurity threats and the associated risks. Attack trees are structure similarly to "fish bone diagrams" used in root cause analysis. [194]**.**
- **MITRE ATT&CK** (**Adversarial Tactics, Techniques, and Common Knowledge**). A data base of real-world cybersecurity vulnerabilities for IT and terrestrial OT systems. [195].

- **Aerospace Corporation SPARTA (Space Attack Research and Tactic Analysis).** A new cybersecurity vulnerability database similar to ATT&CK but tailored specifically to space systems. This is the most applicable source to this research for defining the attack surface of a satellite. [196].

SPARTA, the newest of the above frameworks, is a reference for cybersecurity engineers when designing a satellite, but non-cyber engineers can find it difficult to assimilate SPARTA due to the depth and complexity of the framework. SPARTA is a comprehensive framework for cybersecurity but does not include cost, schedule, architecture, or implementation considerations. In general, SPARTA may be too comprehensive for small-scale satellite programs to utilize effectively, and would therefore be subject to the same pitfalls and extra tailoring work as RMF (as was described in Research Question 2).

The successful integration of cybersecurity and satellite architecture hinges on the definition and creation of design requirements which can be built into the architecting process. The selection and integration of the correct tools and methodologies are a crucial component to the success of such overall design integration.

In summary, this background research into cyber modeling tools has identified many tools and frameworks to conduct cybersecurity threat modeling which are commercially available and have been built and refined for terrestrial IT and OT systems. These tools are specialized to cybersecurity, however, and do not currently integrate well with space system architecture development tools. This observation shows that, there exists a barrier between the integration of cybersecurity and satellite modeling and requirements definition. This dissertation therefore

proposes a process to define a typical satellite architecture process and iteratively incorporating cybersecurity into the process.

5.6 How to Apply MBSE to a define a Cyber-Secure Satellite Architecting Process

This research leverages a synthesis of various methodologies of applying MBSE to a space program to define a cyber-secure satellite architecting process based on MBSE by defining the architecture definition through the use of a capabilities-driven approach. Capabilities-driven architecture processes start the architecture definition by defining what the system needs to do to accomplish the customer objectives [197]. The process is tailorable and can be used to define a great depth of information covering all the minute behaviors and interfaces of a system.

Companies have typically already established document-based architecture methodologies and tools to design satellites (i.e. excel calculators, external disconnected analysis software, Microsoft Word Interface Control Document (ICD) and specifications, etc…) before they make the decision to pursue a Class C/D satellite contract, so adopting a purely MBSE approach to architecture definition becomes cost prohibitive [198]. Therefore, for Class C/D satellite programs, the use of MBSE becomes a cost-benefit decision of where to apply the tools to maximize the return on investment and minimize risk when compared to traditional document-based approaches.

There are many ways to apply an MBSE architecture process [199] [200], but MBSE software applications do not explicitly define this process [201]. Through working on multiple satellite programs with numerous other space engineers, we have observed that the following approach to MBSE, as outlined in the proceeding sections of Chapter 5, will be most effective for implementing MBSE design on Class C/D satellites. This approach may have commonalities

with various industry approaches to using the MBSE tools to define model elements, but the proposed process is a novel approach to defining a space system architecture with MBSE. It is differentiated in that, here we use an architecture-driven approach to define a cyber-secure system rather than the traditional method of implementing an architecture-disconnected cybersecurity approach using the tools described in Section 5.5. As a result, we believe this approach is an appropriate and beneficial step to applying an MBSE systems engineering to develop commercial Class C/D satellite systems and enable engagement with cyber engineers and space designer engineers to define the cybersecurity interfaces of the satellite.

5.6.1 Defining The Model Organization

To define the model on which we will implement this approach, we adopt a modern set of MBSE tools such as Cameo Systems Modeler from Dassault Systemes. A well-organized containment tree is imperative to make the model effective, consistent, and navigable, and MBSE generally encourages the creation of both a Logical Containment Tree and a Physical Containment Tree [25]. However, creating and tracking two trees adds complexity and expense – resources which Class C/D satellite programs are often short on. Therefore, our proposed Class C/D MBSE approach simplifies the containment trees into a single overarching tree, as pictured in **Figure 6**. The containment tree should be organized in such a way that it drives the different aspects of modeling and helps to partition different modeling artifacts in their own packages. The core six key packages that we propose to be the foundation of any MBSE model are shown in **Figure 6**.

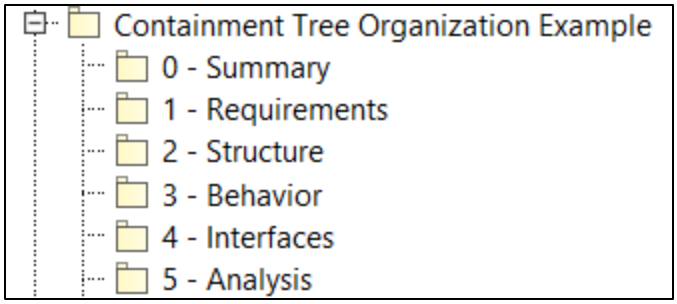

Figure 6: MBSE Model Packages

- **0 - Summary:** This package houses top level model information such as change logs, TBXs, acronyms, glossaries, programmatic documentation, design review check lists, model landing pages, etc. The Summary Package is a miscellaneous package for information that provides context or background details for the specific package level.
- **1 - Requirements:** This package contains requirements and verifications. For example, at the segment level this will contain segment requirements, and at the subsystem level this will contain subsystem requirements. Additionally, once a program reaches the verification stage, this package will contain the verification artifacts and trace them back to the requirements.
- **2 - Structure:** This package is one of the main model packages. It contains the structural model elements like the Block Definition Diagrams (BDD) and Internal Block Diagrams (IBD) that define the system architecture and define the logical and physical composition of the system.
- **3 - Behavior:** This package contains all the behavioral diagrams such as activity diagrams, use case diagrams, and other operational behaviors.
- **4 - Interfaces:** This package contains specifics on system interfaces such as a space-to-ground ICD, hardware electrical/mechanical/software ICDs, and software exchange items for modeling purposes. This package will contain interface specific BDDs and IBDs.

- **5 - Analysis:** This package contains model-specific analyses or externally referenced analytical model that are necessary as defined per programmatic requirements and is therefore crucial to space programs. If the model is being used to convey requirement verification and system performance to tools external to the MBSE software application, this package could link to the external analytical reports and external tool outputs.

There are numerous ways to setup an MBSE model which are typically predicated on organization and context that is important to the specific MBSE practitioner. For example one effective way is defined in the book *Effective Model-Based Systems Engineering* [25], and is similar to our proposed method, addressing most of the same content, as compared below (see Figure **7**). Our recommendation is simplified to focus on the minimal key packages for Class C/D Space Programs and is specifically based on our experience implementing MBSE on several Class C/D space programs.

Our Recommended Packages	[25] Recommended Packages
0 – Summary	(f) – General Information
1 – Requirements	(g) - Requirements
2 – Structure	(b) – Structure (e) - Context
3 – Behavior	(a) – Use Cases (c) - Behavior
4 – Interfaces	(d) - Data
5 - Analysis	

Figure 7: Recommendation Compared to Alternative Industry Recommendation

Taking this model structure a step further, we recommend replicating this package structure for all levels of the model. For example, each of the segments defined in Research Question 1 (Ground, Launch, and Space segments) has its own subsystems and components. Applying the model structure defined above in **Figure 6** produces a structure similar to the prototype example shown in **Figure 8.**

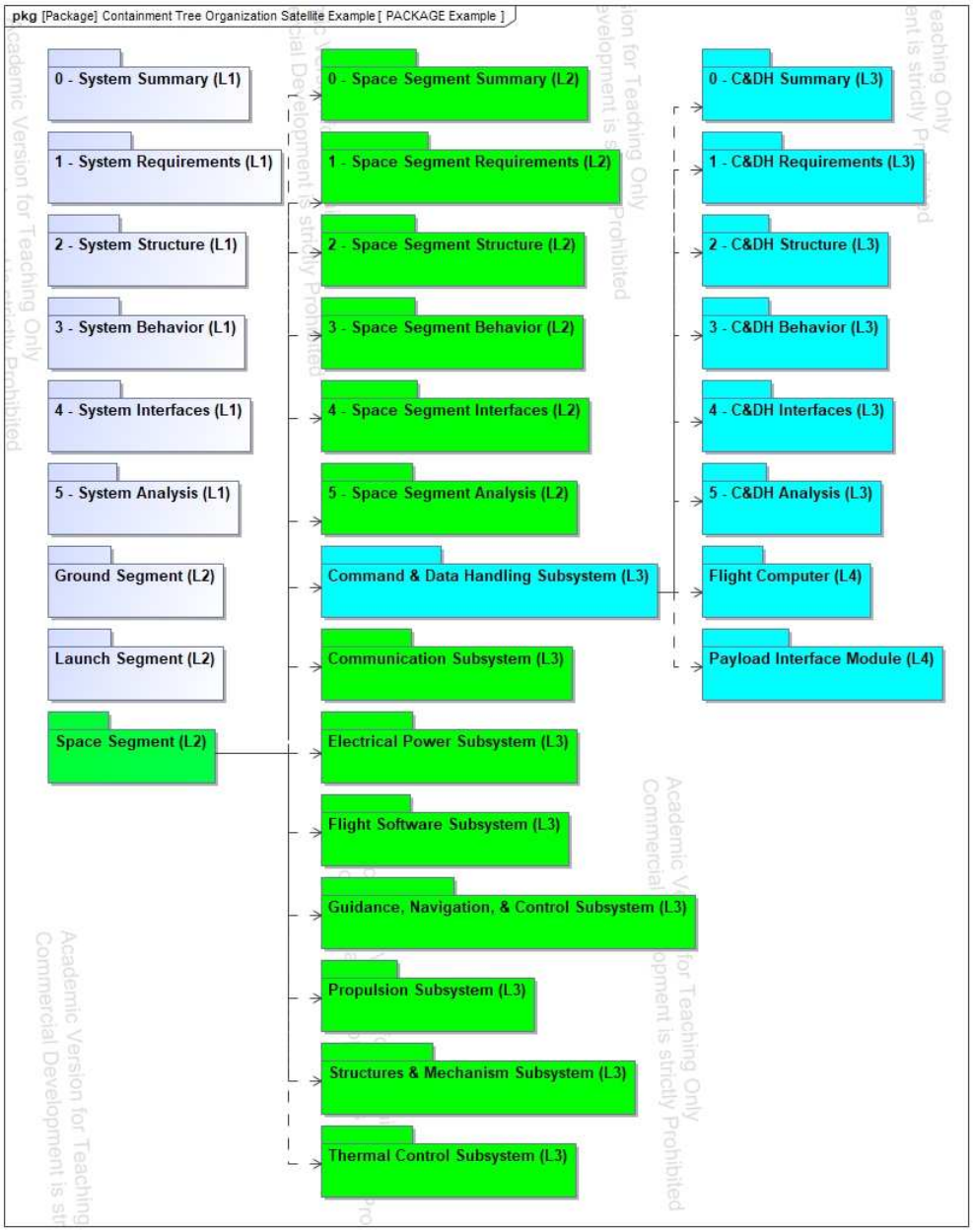

Figure 8: Example MBSE Containment Tree

Figure 8 illustrates that the prototype MBSE space system has the MBSE packages defined in **Figure 6** and also contains the Ground Segment, Launch Segment, and Space Segment which are the basic elements of a total Space System as defined in **Research Question 1** (see gray boxes on the left column). One of the top-level segment boxes, the Space Segment, is opened to show that the segments also comprise the same prescribed MBSE

packages in addition to containing the further Subsystems of that particular segment (see green boxes in the center column). Finally, the Command and Data Handling (C&DH) Subsystem of the Space Segment has been opened to exhibit the same MBSE packages contained in each subsystem as well as its individual Subsystem Components (see teal boxes on the right; in the case of the C&DH subsystem, these components are the Flight Computer and Payload Interface Module).

5.6.2 Defining the Modeling Approach

After setting up the containment tree for a Class C/D satellite program as described in **Section 5.6.1**, a general flow to generate the model contents should be followed for building the capability-driven architecture. Our recommended flow is pictured below in **Figure 9**.

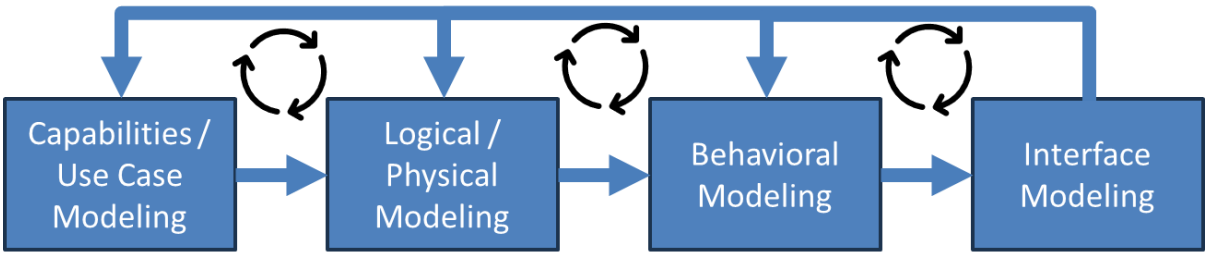

Figure 9: MBSE Model Creation Flow

This process enables incremental model development and integration and is demonstrated in detail in **Section 5.7** of this dissertation. A brief overview is as follows:

- The first step to creating the MBSE model is to define the system capabilities. To simplify this first step, we will construct natural language model elements called Use Case Diagrams which describe what the satellite system needs to do to meet its objectives. These diagrams will be iteratively developed as the system requirements are defined and further refined.

- The second step is to conduct logical and physical modeling of the BDD and IBD system elements, starting with modeling the overall system, then breaking it into smaller pieces and modeling the segments, next moving into the subsystems, and finishing with the individual components. These diagrams will also be iteratively developed as the logical elements are refined through various working groups and as physical elements such as individual components are down selected.
- The third step is to conduct behavioral modeling. The behavioral modeling consists of further refinement for how the system will operate. This includes creating activity diagrams and state diagrams. For example, this refinement would pertain to the various states of each component and then how those states form the various operational modes of the spacecraft. The bulk of the flight software development happens during this stage. This stage could also include activities such as modeling and predicting how the system will respond in various fault scenarios. These diagrams will also be iteratively developed as the logical and physical elements are refined.
- The fourth step is to conduct interface modeling. Once a preliminary architecture has been defined, the various interfaces can start to be modeled. For example, this could be looking at hardware ICDs or defining a Space-to-Ground ICD and then allocating the specific commanding or telemetry to those interfaces.

One of the powerful benefits of applying a structured MBSE process, as outlined above, is that such a process promotes teamwork and model iteration. Teams can use a process similar to a weekly MBSE Review Board to review and control what gets added to or modified within the model. This iterative architecting process allows the satellite program to integrate engineers from multiple disciplines, including systems engineers, subsystem design engineers, test

engineers, and manufacturing engineers, throughout the entire design process. It also allows the multidisciplinary team to work through the functional decomposition of the capabilities while balancing the feedback and perspectives of all team members. It is through this process that a program can push top level system behaviors and CONOPS to the lowest system levels of component selection and behaviors to satisfy the mission.

As explained in **Research Question 2**, space companies need a time-efficient process which their team can easily comprehend to increase their competitiveness and thus increase the number of contracts they can win and their company revenue. Based on our experience with industry and through research observations, this model creation flow is our recommendation of a simplified process for implementing MBSE to design a Class C/D space system. This tactical process strives to maximize the benefits of MBSE across the multidisciplinary team while minimizing the overall complexity of the MBSE architecting approach. By leveraging the process described herein, we can augment the general MBSE satellite design process with cybersecurity capabilities to create an MBSE cyber-secure satellite architecting process, as demonstrated in **Section 5.7**.

5.7 Example of Applying MBSE to Define a Cyber-Secure Satellite Architecting Process

5.7.1 Defining System Threads, and Top-Level Use Case Diagrams

For the purpose of this dissertation, we will focus on the satellite architecture process and providing a case study of how one could integrate cybersecurity by implementing MBSE. For the purpose of this dissertation, the satellite mission will be a simple Class D Earth Weather Observation Satellite in a LEO orbit. Our MBSE architecture process starts with defining System Threads and portraying them on a Use Case diagram to describe the capabilities which the

satellite needs to perform to satisfy the mission CONOPS. The Use Case diagrams will cover the major satellite design phases and the various Design Reference Missions (DRM) which the satellite will accomplish during its lifecycle. The System Threads on a Use Case diagram (simplified for the purposes of this dissertation) is shown below in **Figure 10** and illustrates the various threads the Class D satellite could have.

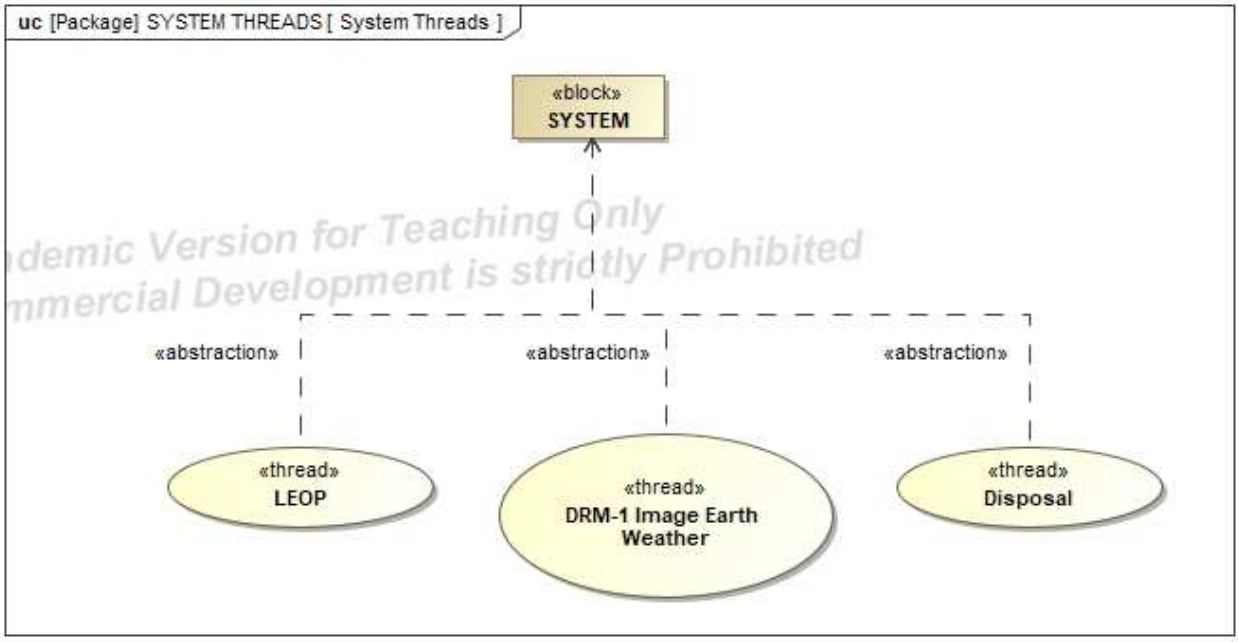

Figure 10: Satellite Threads Summary

- **Launch and Early Operations Phase (LEOP):** This thread will contain the unique system capabilities that are utilized during satellite separation from the launch vehicle, detumbling, and system checkout.
- **DRM-1 Image Earth Weather:** This is the main mission thread and is further elaborated upon throughout this dissertation. Note that the singular DRM-1 in this Use Case diagram is a simplification for the purposes of this dissertation – there could be multiple DRMs on a given satellite.

- **Disposal:** This thread will contain the unique system capabilities to decommission the satellite systems and perform disposal operations.

This dissertation will focus on the DRM-1 mission thread called *Image Earth Weather* of the Class D satellite as the primary system thread and elaborate from the Use Case diagram in **Figure 11** for further decomposition. Within this system thread we define the space segment capabilities as Use Cases. To define the Use Case we utilize a custom MBSE stereotype "SS Capabilities" for the space segment (SS) satellite capabilities. The culmination of the capabilities portrays the primary mission CONOPS which are the core functions the satellite must perform in order to execute this system thread.

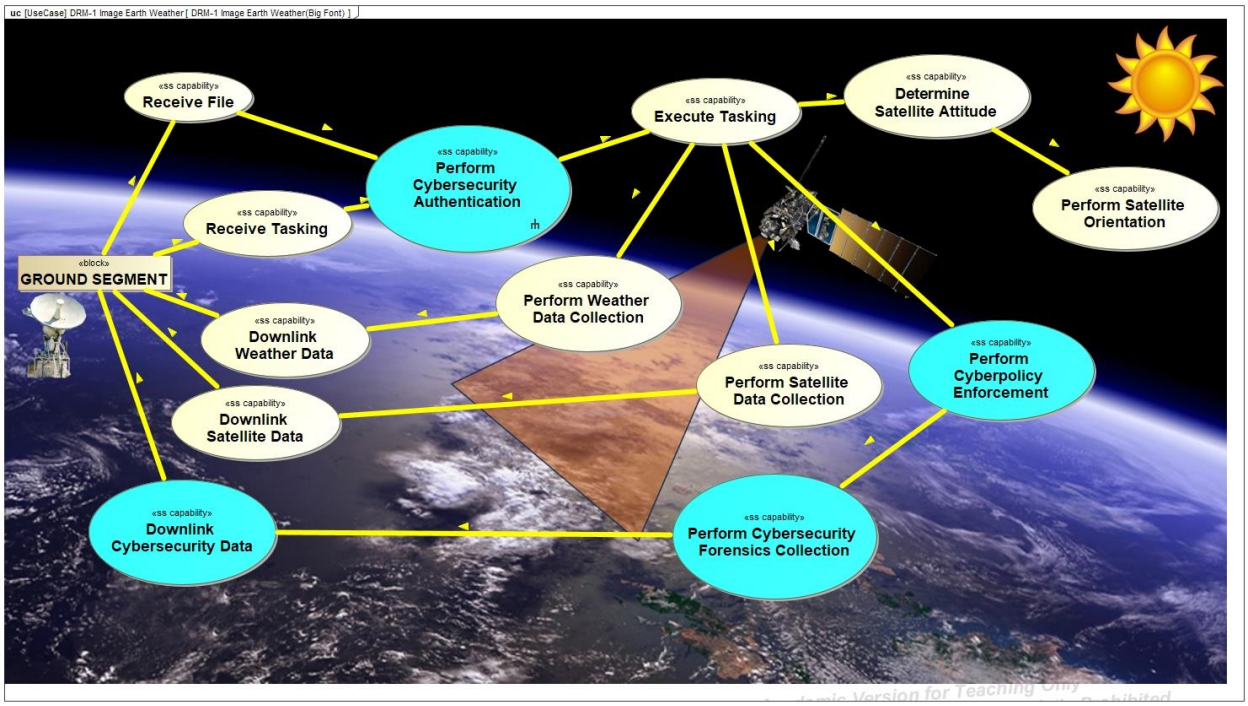

Figure 11: DRM 1 Use Case Diagram

Figure 10 shows the Use Case Diagram for the *DRM-1 Image Earth Weather* thread. This Use Case Diagram depicts the satellite procedure when the Ground Operator sends a

command to the satellite to image a specific portion of the earth's weather (for example, a developing hurricane). Upon receiving a command, the satellite will conduct operations to verify the authenticity of the command through a cybersecurity authentication gate, then orient its position, take a series of high-resolution images, and finally downlink the data to back to the Ground Segment on Earth. To convey the Use Cases, SS Capabilities describe the actions the satellite must conduct to perform the mission. The simplified CONOPS depicted in the Use Case Diagram in **Figure 11** is a representative example for a common satellite development program. However, we have extended this typical CONOPS by modeling a few cybersecurity-specific capabilities such as:

- **Perform Cybersecurity Authentication:** This capability will serve as adjudicating uplinked commanding to the satellite to ensure it is authentic and correctly formatted for how the satellite is intended to operate.
- **Perform Cyberpolicy Enforcement:** This capability is responsible for protecting the satellite from detected cyber threats.
- **Perform Cybersecurity Forensics Collection:** This capability collects and logs any cyber activities on the satellite.
- **Downlink Cybersecurity Data:** This capability serves to downlink cyber-specific telemetry to ground operators.

This Use Case Diagram illustrates the first step in implementing an MBSE approach while including cybersecurity into the architecture derivation process.

5.7.2 Defining Use Case Descriptions

The next step for a capabilities-driven MBSE approach is to define the Use Case description for each capability. These descriptions are used to help teams rationalize about what the specific capability means and how it functions for the system. These Use Cases are going to further drive system decomposition. A Use Case description consists of the following key attributes:

- **Definition:** Defines the purpose and intent of the capability.
- **Initiating Actor & Event:** Defines who or what initiates the capability. This could be external to the satellite, such as the Ground Station.
- **Pre-conditions:** Defines assumptions or required satellite conditions for the capability to execute.
- **Scenarios:** Defines various scenarios in which the capability is exercised. These scenarios can cross multiple capabilities, i.e. uplinking commands to the satellite, and will serve as the basis for the activity diagram modeling below in **Figure 12**.

Figure 12 shows an example description for the ***Perform Cybersecurity Authentication*** Use Case.

#	Name	Documentation
12	Perform Cybersecurity Authentication	DEFINITION: This capability will serve as adjudicating uplinked tasking or files to the satellite to ensure it's authentic and correctly formatted for how the satellite is intended to operate. The sattelite will validate that all commanding and tasking authentic as define in a on board database of allowable commanding, allowable ground stations, and allowable file signatures. INITIATING ACTOR & EVENT: Sattelite either receives a tasking or file from the ground. PRE-CONDITIONS: Sattelite is operational and in a state that it can receive tasking and/or files from the ground. SCENARIOS: 1. Tasking uplinked from ground segment 2. File uplinked from ground segment

Figure 12: Perform Cybersecurity Authentication Use Case

5.7.3 Defining Segment Requirements

After the Use Case Description is defined for an SS Capability, the MBSE practitioner will select one of the following two paths:

1) The MBSE modeler can develop L2 Space Segment requirements from scratch by leveraging the description from the specific SS Capability.
2) The MBSE modeler can link the SS Capability to existing L2 Space Segment requirements (i.e. if the customer flowed space segment requirements) and go through an iterative process in defining mission SS Capabilities and missing L2 Space Segment requirements from the SS Capabilities required to execute the mission.

For the purpose of this dissertation, we will demonstrate the latter path by using the satellite level Cybersecurity Satellite requirements developed in **Research Question 2**. This example will demonstrate a sub-set of requirements that would typically be used to define a satellite. **Figure 13** shows the example cybersecurity requirements organized in an MBSE Requirements Table. There are many attributes that can define a requirement; however, for the

purpose of this Dissertation we are focusing on the key fields to convey what the requirement is and how it would be verified.

#	Id	Name	Text	Rationale	Verify Method	Verification Approach
1	L2-SS-01	Software Secure Boot	The Satellite shall utilize a secure boot methodology for loading flight software.	The satellite should utilize modern avionics. Most modern satellite avionics utilize modern processors that have secure boot capability. Secure boot involves digitally signing the Flight Software with a cryptographic key built into the avionics [25].	Test	During AI&T upload a new image of flight software that has a secure boot signature and verify that the satellite loads the new flight software.
2	L2-SS-02	Communication Encryption	The Satellite shall utilize commercial encryption, or equivalent, for all external communication data links.	The satellite should utilize commercial encryption. Many commercial satellite encryptors are readily available, and it is also possible to build encryption into their operating software. CNSA is the current state-of-the-art encryption for commercial systems [26]. It has superseded the widely adopted NSA Suite B; however, this requirement includes "or equivalent" to enable affordable options.	Test	During AI&T the satellite will utilize a ground modem with encryption/decryption capabilities to demonstrate the satellite can communication with encryption.
3	L2-SS-03	Cybersecure Communication Protocol	The Satellite shall utilize a communications protocol that includes built-in cybersecurity protections.	CCSDS is one of the most popular communications protocols for satellites. Modern CCSDS includes many cybersecurity features as part of the protocol, such as integrity validation of the data transmitted, authenticity validation of the data source, and anti-replay [27]. This requirement calls for the use of these defined features. In addition, depending on the communication protocol, it is possible to have encryption built into the protocol [28].	Inspection	Inspection of communication protocol and security features will show which cybersecurity threats the protocol protects against.
4	L2-SS-04	Cyber-Secure Gold Copy FSW	The Satellite shall incorporate a cyber-secure "gold copy" of flight software that can be loaded in the event of a critical cyber fault.	On most satellites, it is standard practice to have multiple boot copies of flight software. Under this requirement, one of these boot copies should be cyber-secure and can be defaulted to in the event of a critical cyber fault. This will result in the ability for the satellite to load itself into a known cyber-safe state.	Test	During FSW testing the software will demonstrate that in the event that the satellite is compromised it's loaded into a cyber-secure state.
5	L2-SS-05	Uploaded Software Authentication	The Satellite shall validate the authenticity of uploaded software prior to accepting the uploaded data into on-board storage.	The satellite should check that all software is authentic from the operator sending the data prior to allowing it to load onto the satellite.	Test	During AI&T the satellite will show that it authenticates software prior to loading onto the satellite and notify ground operators.
6	L2-SS-06	Uploaded Software Integrity	The Satellite shall validate the integrity of uploaded software prior to loading.	The satellite should check that all software is properly formatted prior to allowing it to execute onto the satellite.	Test	During AI&T the satellite will verify the integrity of software prior to loading onto the satellite and notify ground operators.
7	L2-SS-07	Adaptable Cyber Policy	The Satellite shall have an adaptable cybersecurity policy for all mission modes.	Many satellites have different operational modes (i.e. Nominal Ops, Initialization, Safe Mode, End of Life). The cyber solution needs to account for these system operational mode changes. In addition, this will ensure the satellite incorporates cyber-safe counter measures if integrated with a fault management response.	Test	The satellite will under go cyber penetration testing. The cyberpolicy for the various mission modes will be exercised against various cyber threats.
8	L2-SS-08	Cyber Event Detection	The Satellite shall detect cyber events on the satellite.	This requirement is intentionally open-ended. Satellite manufactures should conduct threat vector analysis to determine the appropriate threats to protect against. As part of this requirement, the satellite manufacturer should determine specific commands or telemetry points that could be used to determine if or when a cyber event has happened. Examples are jamming attempts, loss of signal, unauthorized access, replay attempts, command accepts/rejects, and planned ground contact windows [29].	Test	The satellite will under go cyber penetration testing. The cyberpolicy for the various mission modes will be exercised against various cyber threats.
9	L2-SS-09	Cyber Event Reporting	The Satellite shall report cyber events on-board the satellite to ground operators.	When a cyberevent is detected the satellite needs to alert operators with the event and forensics.	Test	The satellite will under go cyber penetration testing. The cyberpolicy for the various mission modes will be exercised against various cyber threats.

Figure 13: Example Space Segment Requirements

5.7.4 Tracing Use Cases to Segment Requirements

Now that the satellite requirements are integrated into the MBSE tool, the next step is to trace the requirements to the four cybersecurity SS Capabilities which we created above, as shown in **Figure 14**. This dissertation is only demonstrating these four cybersecurity use cases

for the sake of simplicity; a typical satellite program would have many more requirements for each of the various subsystems and top level satellite functionality.

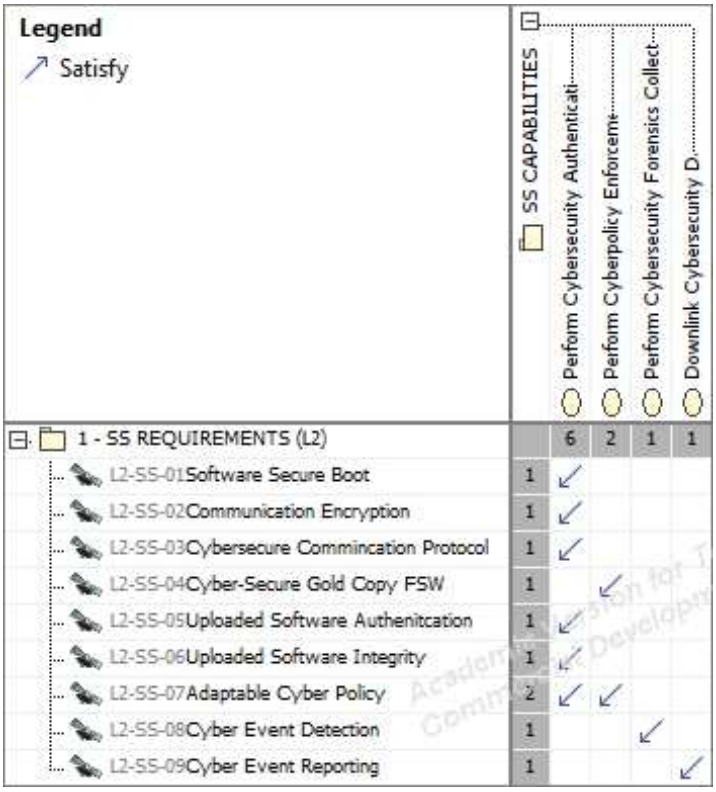

Figure 14: Trace of Space Segment Requirements to SS Capabilities

Determining the tracing between use case and requirement is usually an iterative process which follows one of two possible methods. The first method is to first define capabilities and then derive from the capabilities multiple requirements per use case. The second method is described in Research Question 2 which starts with defining requirements and then iteratively creating capabilities which then are mapped to the requirements. We already defined space segment cybersecurity requirements in Research Question 2, so for this dissertation, we reviewed the requirements and allocated them accordingly to capabilities. The creation of the relationship between these model elements leads to coherency between the Space Segment requirements

verification and the linked capabilities of the CONOPS: When the Space Segment requirements are verified, the CONOPS is also satisfied. This is a key step in demonstrating that the model decomposition satisfies the mission.

5.7.5 Decomposing Segment Requirements into Subsystem Activities and Subsystem Requirements

The next step is to determine how these L2 space segment requirements will decompose into subsystem requirements. The logical architecture MBSE model was defined in **Research Question 1** and is shown in **Figure 15**.

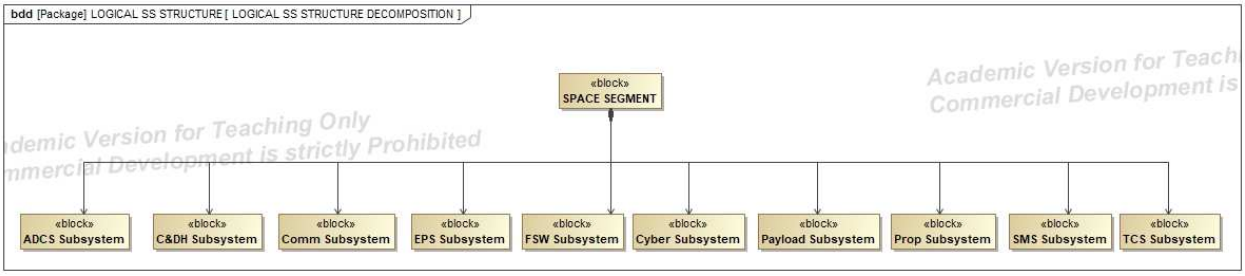

Figure 15: Logical Architecture

In order to add emphasis to the importance of cybersecurity for a satellite, we have extended the logical architecture by modeling an explicit cybersecurity subsystem that will handle the cybersecurity-related features of the satellite. The cybersecurity subsystem will augment the flight software subsystem as an accompanying application layer that processes system information, examines for cyber intrusions, and then logs any intrusions and sends alerts to operators.

After completing this logical architecture, we can trace the L2 space segment requirement to the logical subsystems. This process entails iterative rationalization in determining which

subsystems are required to perform each specific higher-level requirement. The L2 requirements of this example satellite will decompose as shown in **Figure 16**.

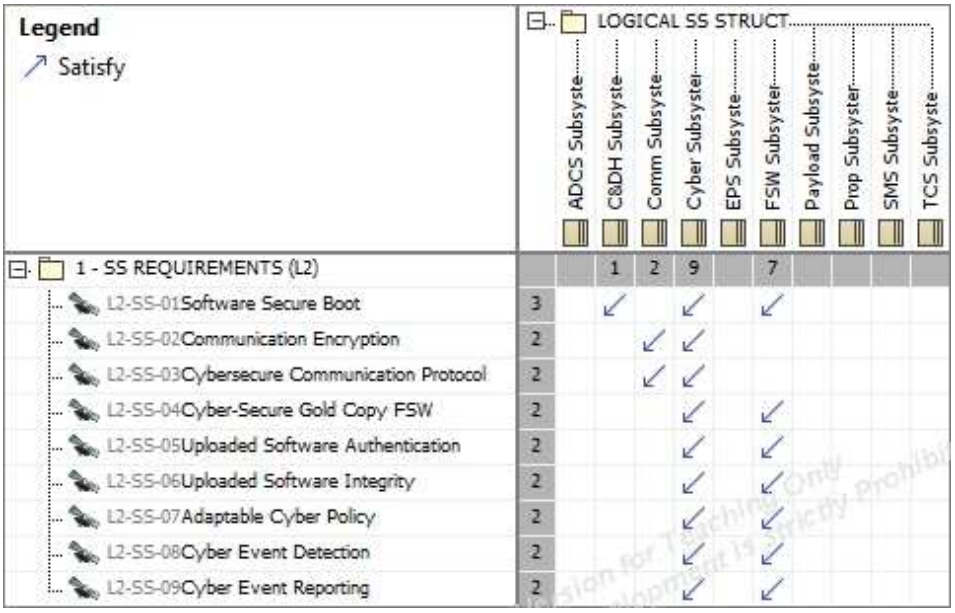

Figure 16 Space Segment Requirements to Logical Architecture

For example, accomplishing the goal of the requirement L2-SS-02 “Communication Encryption” (to send and receive encrypted communications) will require both the software encryption embedded in the cybersecurity subsystem coupled with the communication hardware.

The logical decomposition allocated the requirements to their subsystems, helping with model organization and further refinement. This is considered a logical decomposition because modeling in the physical domain requires a way to allocate subsystem-disconnected physical components to their logical subsystems. Once the requirements are linked to their logical subsystems, the functional decomposition of the SS Capabilities into subsystem activities can be continued. Next, this example leverages the pre-defined Use Cases to begin modeling each scenario as an activity diagram. Before the activity diagram can be modeled, the following characteristics of the specific scenarios must be fully defined:

- **Scenario Description:** Narrative that describes what happens in the specific scenario.
- **Course of Events:** Defines the incremental steps that must occur for this SS Capability to happen.

Figure 17 shows an example of this decomposition for the ***Perform Cybersecurity Authentication*** capability.

#	Name	Documentation
1	Perform Cybersecurity Authentication	SCENARIOS: 1. Tasking uplinked from ground segment 2. File uplinked from ground segment
2	S1 - Tasking uplinked from ground segment	SCENARIO DESCRIPTION: The satellite receives a tasking list from the ground segment and the satellite processes the tasking list for cybersecurity authentication prior to accepting the tasking. COURSE OF EVENTS: 1. Satellite receives a tasking list from the ground segment. 2. The Satellite sends the tasking list from the communication subsystem to the cybersecurity subsystem. 3. The cybersecurity subsystem process the tasking list to ensure all commands are valid and authentic. 4. The cybersecurity subsystem sends the tasking list to flight software for further task execution.
3	S2 - File uplinked from ground segment	SCENARIO DESCRIPTION: The satellite receives a file (either software updates or configuration files) from the ground segment and the satellite processes the files for cybersecurity authentication prior to accepting the file. COURSE OF EVENTS: 1. Satellite receives a file from the ground segment. 2. The Satellite sends the file from the communication subsystem to the cybersecurity subsystem. 3. The cybersecurity subsystem process the file to ensure all commands are valid and authentic. 4. The cybersecurity subsystem sends the file to flight software for further task execution.

Figure 17: Course of Events to Perform Cybersecurity Authentication

The next step is to create an Activity Diagram, the purpose of which is to rationalize the specific behaviors the cybersecurity subsystem must perform to accomplish the SS Capability. The Activity Diagram also provides context for the specifically modeled activities by showing the predecessor / successor activities in the sending / receiving subsystems. Focusing on Scenario 1 “Perform Cybersecurity Authentication,” **Figure 18** depicts the course of events as an activity diagram.

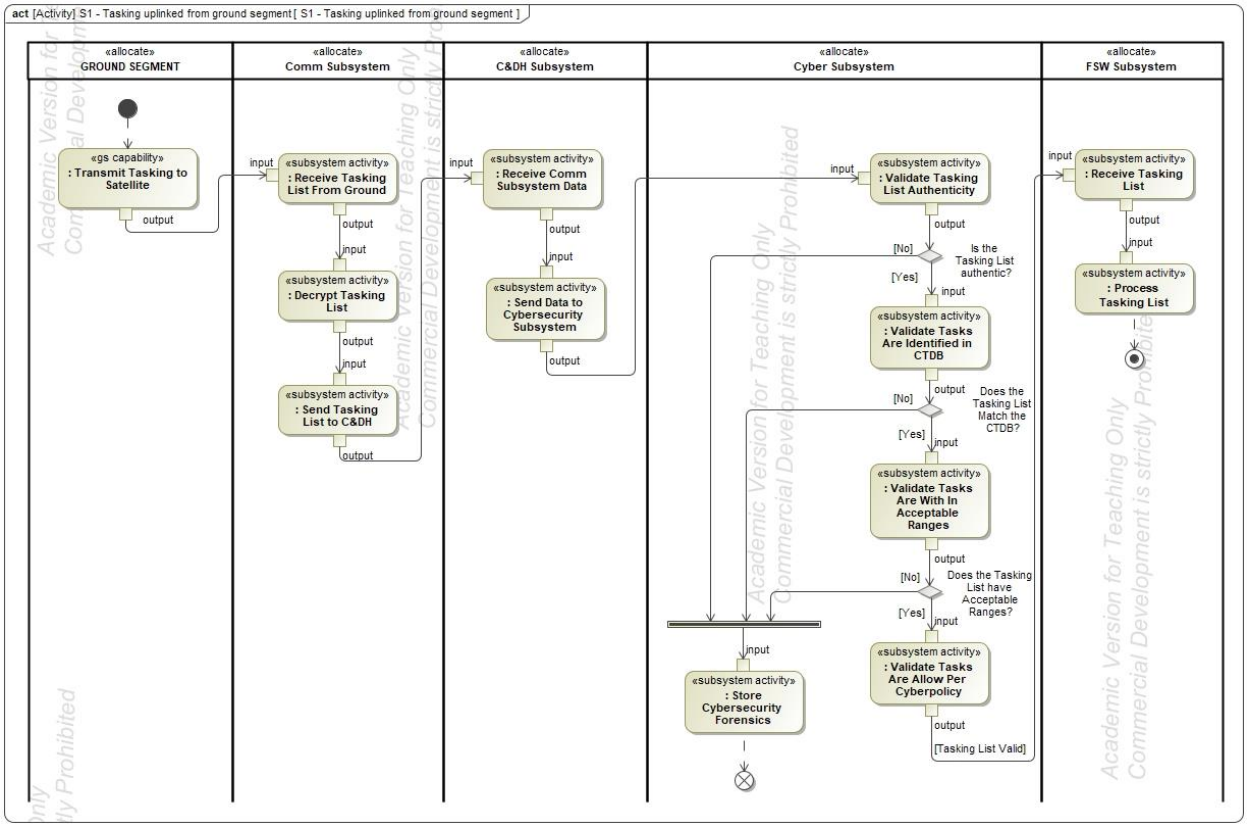

Figure 18: Activity Diagram Depicting Tasking Uplinked From Ground Segment

Figure 18 depicts the following steps:

1. The satellite receives a tasking list from the ground segment.
2. The communication subsystem receives the tasking list, decrypts the data and sends it to the command and data handling subsystem.
3. The command and data handling subsystem receives the data and sends it to the cybersecurity subsystem.
4. The cybersecurity subsystem goes through several checks to ensure the tasking list is valid:
    i. Validates the tasking list authenticity.

ii. Validates that the commands match those as approved in the command and telemetry database.

iii. Validates that the commands are within their approved ranges.

iv. Validates that the tasks are allowed in accordance with the satellite cybersecurity policy.

5. If the tasking list fails one of the above checks, the cybersecurity subsystem will log the failure for downlinking to the ground segment.
6. If the tasking list is valid, then the tasking list is sent to the flight software subsystem for processing.

Now that a subsystem activity diagram has been created, the L3 cybersecurity subsystem requirements can be defined. For the purpose of this dissertation, the L3 cybersecurity requirements derivation was leveraged from **Research Question 2** and was expanded through the above demonstrated capabilities functional decomposition. **Figure 19** shows the resultant L3 cybersecurity requirements in an MBSE requirements table and have been expanded to include rationale, verification method, verification approach, and parent requirement trace.

#	Id	Name	Text	Rationale	Verify Method	Verification Approach	Derived From
1	L3-CYB-01	False Command Alert	The Cybersecurity Subsystem shall alert when satellite commands are not identified in the on-board command/telemetry database.	Satellites typically have on-board command/telemetry databases. This capability should be part of flight software; however, this will encourage a threat vector test approach. In addition, depending on the cyber events, this requirement could encompass nefarious commands.	Test	The satellite will under go cyber penetration testing. The cyberpolicy for the various mission modes will be exercised against various cyber threats.	L2-SS-08 Cyber Event Detection
2	L3-CYB-02	Malicious Command Alert	The Cybersecurity Subsystem shall alert when satellite commands are determined to be malicious in accordance with the cybersecurity policy.	This requires a focused look at the specific commands to the satellite. This would look at things such as commands being outside their valid ranges, or if they are hazardous. The scope of this requirement is scalable depending on the specific identified cyber events.	Test	The satellite will under go cyber penetration testing. The cyberpolicy for the various mission modes will be exercised against various cyber threats.	L2-SS-08 Cyber Event Detection
3	L3-CYB-03	Invalid Command Alert	The Cybersecurity Subsystem shall alert when satellite commands are determined to be invalid in accordance with the cybersecurity policy.	This requires a focused look at the specific commands to the satellite. This would look at things such as commands being valid for their use against the mission modes and states. The scope of this requirement is scalable depending on the specific identified cyber events.	Test	The satellite will under go cyber penetration testing. The cyberpolicy for the various mission modes will be exercised against various cyber threats.	L2-SS-08 Cyber Event Detection
4	L3-CYB-04	Malicious On-Board Telemetry Alerting	The Cybersecurity Subsystem shall alert when on-board component commanding is invalid in accordance with the cybersecurity policy.	This is an extension of the above requirements. This requires a focused look at the internal state of health of the satellite. The scope of this requirement is scalable depending on the specific identified cyber events.	Test	The satellite will under go cyber penetration testing. The cyberpolicy for the various mission modes will be exercised against various cyber threats.	L2-SS-08 Cyber Event Detection
5	L3-CYB-05	Anomalous On-Board Telemetry Alerting	The Cybersecurity Subsystem shall alert when satellite telemetry is not as identified in the on-board command/telemetry database.	Satellites typically have on-board command/telemetry databases. This capability should be part of flight software; however, this will encourage a threat vector test approach. In addition, depending on the cyber events this requirement could encompass nefarious telemetry.	Test	The satellite will under go cyber penetration testing. The cyberpolicy for the various mission modes will be exercised against various cyber threats.	L2-SS-08 Cyber Event Detection
6	L3-CYB-06	Invalid On-Board Telemetry Alerting	The Cybersecurity Subsystem shall alert when on-board component telemetry is invalid in accordance with the cybersecurity policy.	This is an extension of the above requirements. This requires a focused look at the internal state of health of the satellite. The scope of this requirement is scalable depending on the specific identified cyber events.	Test	The satellite will under go cyber penetration testing. The cyberpolicy for the various mission modes will be exercised against various cyber threats.	L2-SS-08 Cyber Event Detection
7	L3-CYB-07	Uploaded Software Authentication	The Cybersecurity Subsystem shall validate the authenticity of uploaded software prior to accepting the uploaded data into on-board storage.	The satellite should check that all software is authentic from the operator sending the data prior to allowing it to load onto the satellite.	Test	During AI&T the satellite will show that it authenticates software prior to loading onto the satellite and notify ground operators.	L2-SS-05 Uploaded Software Authentication
8	L3-CYB-08	Uploaded Software Integrity	The Cybersecurity Subsystem shall validate the integrity of uploaded software prior to loading.	The satellite should check that all software is properly formatted prior to allowing it to execute onto the satellite.	Test	During AI&T the satellite will verify the integrity of software prior to loading onto the satellite and notify ground operators.	L2-SS-06 Uploaded Software Integrity
9	L3-CYB-09	Cyber Event Logging	The Cybersecurity Subsystem shall log attempted cyber events on the SV	The satellite creates a log for anything it detects to alert operators. The specifics of what data gets logged will be determined from threat vectors analysis. Some examples are the type of event, where it occurred, when it occurred, the responses the satellite took, and any relevant data used to detect the threat.	Test	The satellite will under go cyber penetration testing. The cyberpolicy for the various mission modes will be exercised against various cyber threats.	L2-SS-09 Cyber Event Reporting
10	L3-CYB-10	Cyber Log Retention	The Cybersecurity Subsystem shall retain logs for up to [TBD] days.	This is a variable requirement. Depending on how often ground operators' communication with the satellite, or how much data is being logged, can determine how much memory is available.	Test	The satellite will under go cyber penetration testing. The cyberpolicy for the various mission modes will be exercised against various cyber threats.	L2-SS-09 Cyber Event Reporting
11	L3-CYB-11	Cyber Log Downlink	The Cybersecurity Subsystem shall downlink logs every [TBD] days.	This is a variable requirement. Depending on how often ground operators' communication with the satellite, or how much data is being logged, can determine how much memory is available.	Test	The satellite will under go cyber penetration testing. The cyberpolicy for the various mission modes will be exercised against various cyber threats.	L2-SS-09 Cyber Event Reporting
12	L3-CYB-12	Cyber Event Reporting	The Cybersecurity Subsystem shall report cyber events on the SV that can be downlinked by flight software.	When a cyberevent is detected the satellite needs to alert operators with the event and forensics.	Test	The satellite will under go cyber penetration testing. The cyberpolicy for the various mission modes will be exercised against various cyber threats.	L2-SS-09 Cyber Event Reporting
13	L3-CYB-13	Reconfigurable Cyber Policy	The Cybersecurity Subsystem shall have a reconfigurable cybersecurity policy.	The policy can be updated and is not hard-coded. This will ensure that, as threats evolve, the cyber defense protections can be updated.	Test	The satellite will under go cyber penetration testing. The cyberpolicy for the various mission modes will be exercised against various cyber threats.	L2-SS-07 Adaptable Cyber Policy
14	L3-CYB-14	Cyberthreat Detection Policy	The Cybersecurity Subsystem shall have the capability to change cybersecurity policy as a response to a detected cyber event.	The satellite can protect itself in the event that a cyber event or intrusion attempts to compromise the system. The specific policies will be determined as part of the threat vector analysis. This requirement also covers how cybersecurity is integrated with the satellite fault management architecture.	Test	The satellite will under go cyber penetration testing. The cyberpolicy for the various mission modes will be exercised against various cyber threats.	L2-SS-07 Adaptable Cyber Policy

Figure 19: L3 Cybersecurity Subsystem Requirements

Now the L3 cybersecurity subsystem requirements can be traced to the logical subsystems, as shown in **Figure 20**. This allocation can also be accomplished by adding the trace in the requirements table.

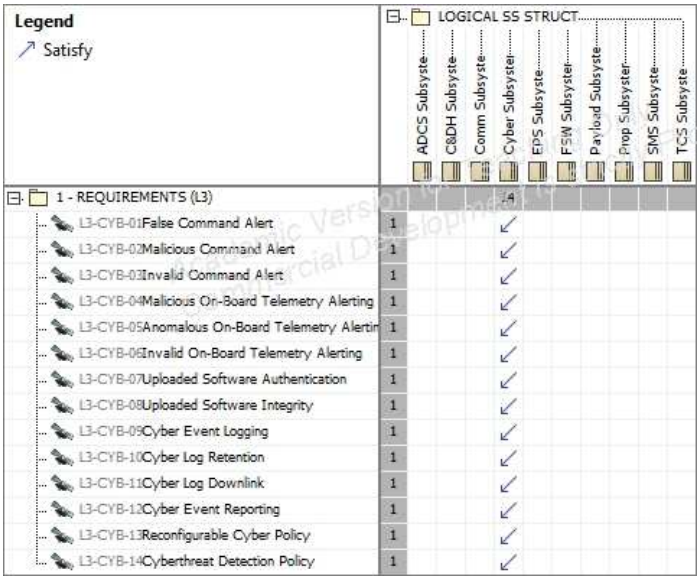

Figure 20: Cybersecurity Subsystem Requirements to Logical Architecture

The final step is to trace the L3 cybersecurity subsystem requirement to their activities as shown in **Figure 21**.

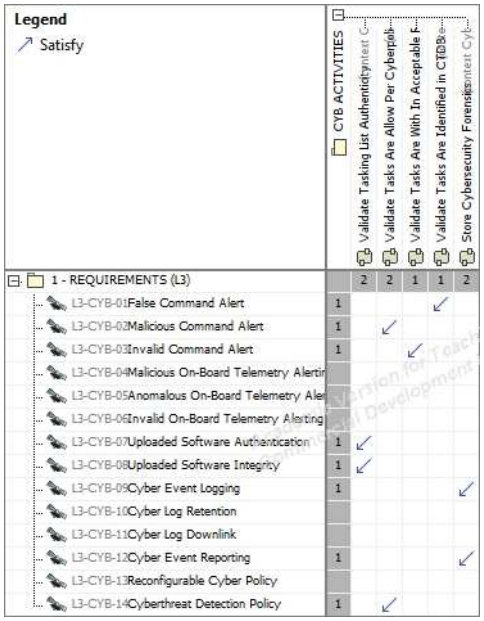

Figure 21: L3 Cybersecurity Subsystem Requirements to Activities

This list summarizes the following requirement-to-activity traces as defined in **Figure 21:**

- L3-CYB-07 and L3-CYB-08 to the *Validate Tasking List Authenticity* activity
- L3-CYB-02 and L3-CYB-14 to the *Validate Tasks Are Allowed Per Cyberpolicy* activity
- L3-CYB-03 to the *Validate Tasks Are Within Acceptable Ranges* activity
- L3-CYB-01 to the *Validate Tasks Are Identified in CTDB* activity
- L3-CYB-09 and L3-CYB-12 to the *Store Cybersecurity Forensics* activity

These subsystem activities can be further decomposed to the component level; however, this example does not carry the decomposition of subsystem activities further than the subsystem level because this is sufficient to demonstrate the purpose of this dissertation which is to convey a methodology to model cybersecurity. This example demonstrates that we have defined the mission CONOPS and have shown what each subsystem (or component) needs to accomplish to meet the mission while demonstrating the traceability of the requirement through its importance and place in the overall mission upon verification.

5.7.6 Logical / Physical Modeling

Once the primary mission CONOPS has been functionally decomposed from system capabilities down to the subsystem level, the Logical / Physical Modeling can proceed. For defining a useful and simplified MBSE process for a Class D mission, logical modeling of each individual component is not necessary and physical modeling for the overall architecture definition suffices. The potential components that are relevant for this mission were evaluated as part of **Research Question 1**. Leveraging the satellite component research conducted as part of **Research Question 1,** we down-selected components for our example satellite. **Figure 22** shows the satellite architecture as defined in **Research Question 1** incorporated into MBSE. The BDD

shows the satellite in the middle of the diagram, and using Directed Composition associations allows the definition of the quantity of each specific component.

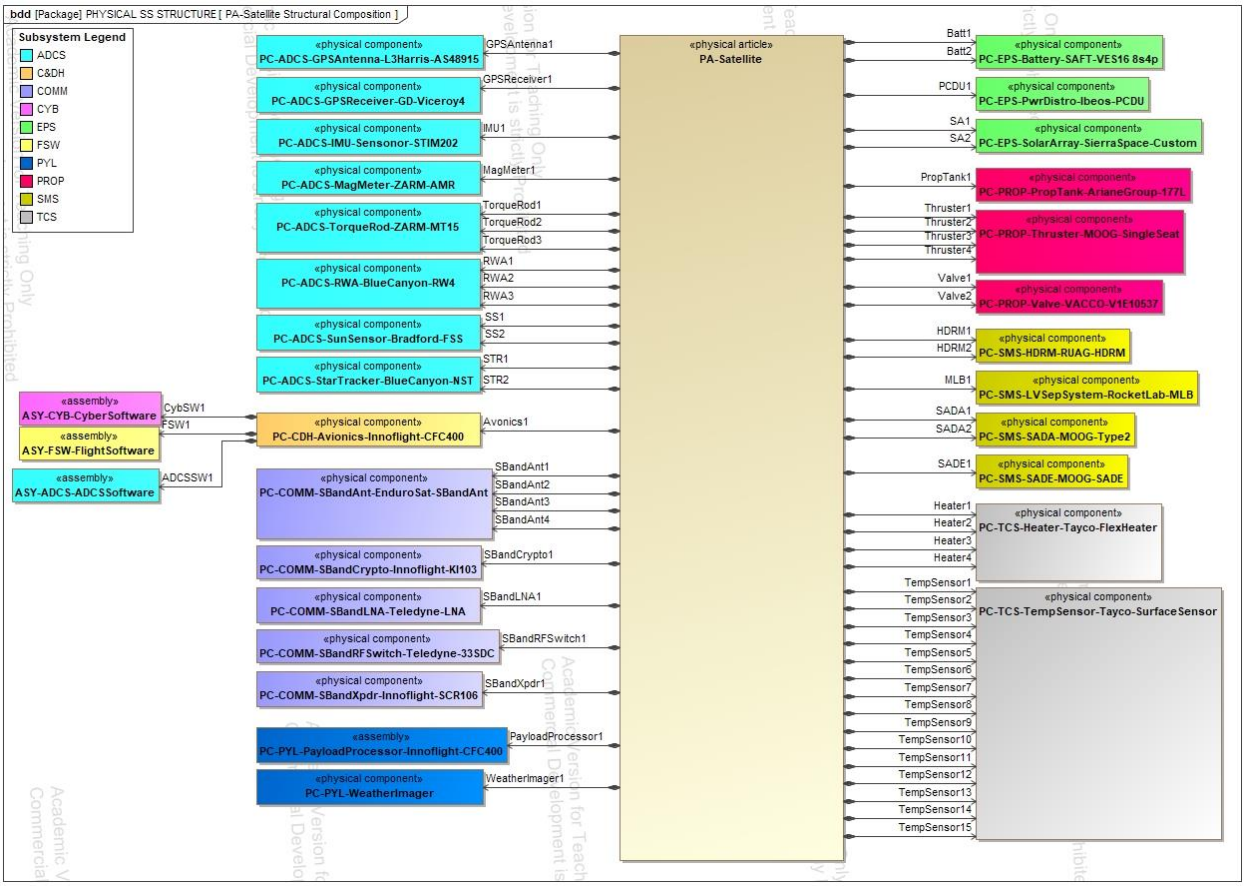

Figure 22: LEO Satellite Physical Components

The Master Equipment List (MEL) is derived from the model of the physical components and includes each component as well as the quantity, vendor, and vendor name for the component. See **Figure 23**.

#	Name	Qty	Vendor	Vendor Name
1	PC-ADCS-IMU-Sensonor-STIM202	1	Sensonor AS	STIM202
2	PC-ADCS-RWA-BlueCanyon-RW4	3	Blue Canyon	RW4
3	PC-ADCS-StarTracker-BlueCanyon-NST	2	Blue Canyon	NST
4	PC-ADCS-SunSensor-Bradford-FSS	2	Bradford	Fine Sun Sensor
5	PC-ADCS-TorqueRod-ZARM-MT15	3	Sinclair Interplanetary	TA-15
6	PC-ADCS-MagMeter-ZARM-AMR	1	ZARM Technik AG	AMR Magnetometer
7	PC-ADCS-GPSReceiver-GD-Viceroy4	1	General Dynamics	Viceroy
8	PC-ADCS-GPSAntenna-L3Harris-AS48915	1	L3Harris	AS-48917
9	ASY-ADCS-ADCSSoftware	1	Custom	Custom
10	PC-CDH-Avionics-Innoflight-CFC400	1	Innoflight	CFC-400
11	PC-COMM-SBandCrypto-Innoflight-KI103	1	Innoflight	KI-103
12	PC-COMM-SBandAnt-EnduroSat-SBandAnt	4	EnduroSat	S-Band Antenna
13	PC-COMM-SBandLNA-Teledyne-LNA	1	Teledyne	LNA
14	PC-COMM-SBandRFSwitch-Teledyne-33SDC	1	Teledyne	33SDC
15	PC-COMM-SBandXpdr-Innoflight-SCR106	1	Innoflight	SCR-106
16	ASY-CYB-CyberSoftware	1	Custom	Custom
17	PC-EPS-Battery-SAFT-VES16 8s4p	1	SAFT	VES16 8s4p
18	PC-EPS-PwrDistro-Ibeos-PCDU	1	IBEOS	Modular PCDU
19	PC-EPS-SolarArray-SierraSpace-Custom	2	Sierra Space	Custom Deployable
20	ASY-FSW-FlightSoftware	1	NASA	cFS
21	PC-PYL-WeatherImager	1	Custom	Custom
22	PC-PYL-PayloadProcessor-Innoflight-CFC400	1	Innoflight	CFC-400
23	PC-PROP-PropTank-ArianeGroup-177L	1	ArianeGroup	177L
24	PC-PROP-Thruster-MOOG-SingleSeat	4	MOOG	SingleSeat
25	PC-PROP-Valve-VACCO-V1E10537	2	VACCO	V1E10537
26	PC-SMS-LVSepSystem-RocketLab-MLB	1	RocketLab	MLB
27	PC-SMS-SADA-MOOG-Type2	2	MOOG	Type 2 SADA
28	PC-SMS-SADE-MOOG-SADE	1	MOOG	SADE

Figure 23: LEO Satellite Master Equipment List

For full satellite programs the MEL can be extended to include additional component properties such as Mass, Power, Thermal, TRL, and various growth margins. The MEL provides an exportable Excel-style reference with an accessible view of the hardware that makes up the satellite which (much like an Excel document) can be re-ordered and manipulated to give desired information (such as adding up a total or partial weight of the components). In addition, the generic MEL table which references all of the parts allows for rapid changes to part properties as the design matures, as well as for roll-ups of potential impacts on the architecture, and provides a holistic view of the specifications of each satellite component.

Now that the physical components have been integrated into the MBSE model, we can create some allocations to link the Requirements (and therefore the CONOPS elements) to the specific components. This is an important step for ensuring traceability between the physical architecture and the requirements. For example, there could be non-functional environmental

requirements which need to be flowed down to specific components that drive the component selection or test program, or functional requirements that are derived based on the components that will determine subsystem behavior. The allocation matrix in **Figure 24** shows that the L3 cybersecurity requirements are satisfied by the Cybersecurity Software which composes the cybersecurity subsystem. If we had decomposed L4 component requirements for the cybersecurity subsystem, we would then further decompose and allocate them to the processes that make up the Cybersecurity Software.

Figure 24: Cybersecurity Subsystem Requirements to Physical Components

The next step is to model the physical interfaces between the components. As shown in **Figure 25**, examining the communication subsystem reveals that only way to communicate with the satellite is through the communication subsystem. The communication subsystem then routes all received data to the C&DH flight computer which hosts the cybersecurity subsystem software.

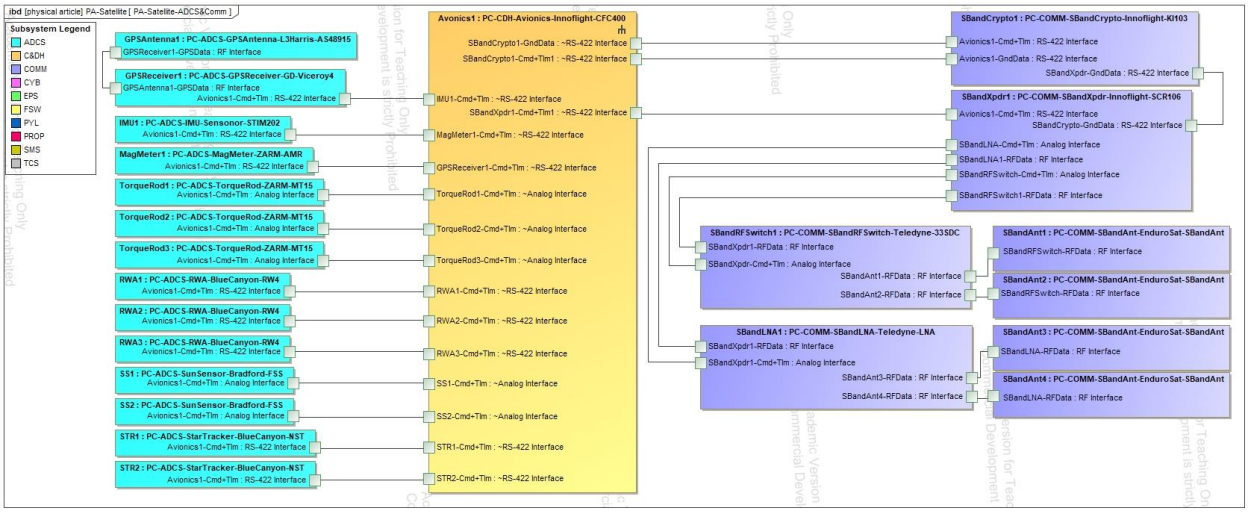

Figure 25: LEO Satellite Physical Architecture

As shown in **Figure 26**, the cybersecurity subsystem is hosted on the C&DH flight computer and interacts with the flight computer physical interfaces and flight software interfaces. This demonstrates that communication external to the satellite goes through the communication subsystem, to the C&DH flight computer, and finally through the cybersecurity subsystem prior to Flight Software (FSW) processing. Additionally, there are multiple interfaces with FSW for the cybersecurity subsystem to process and adjudicate satellite operation.

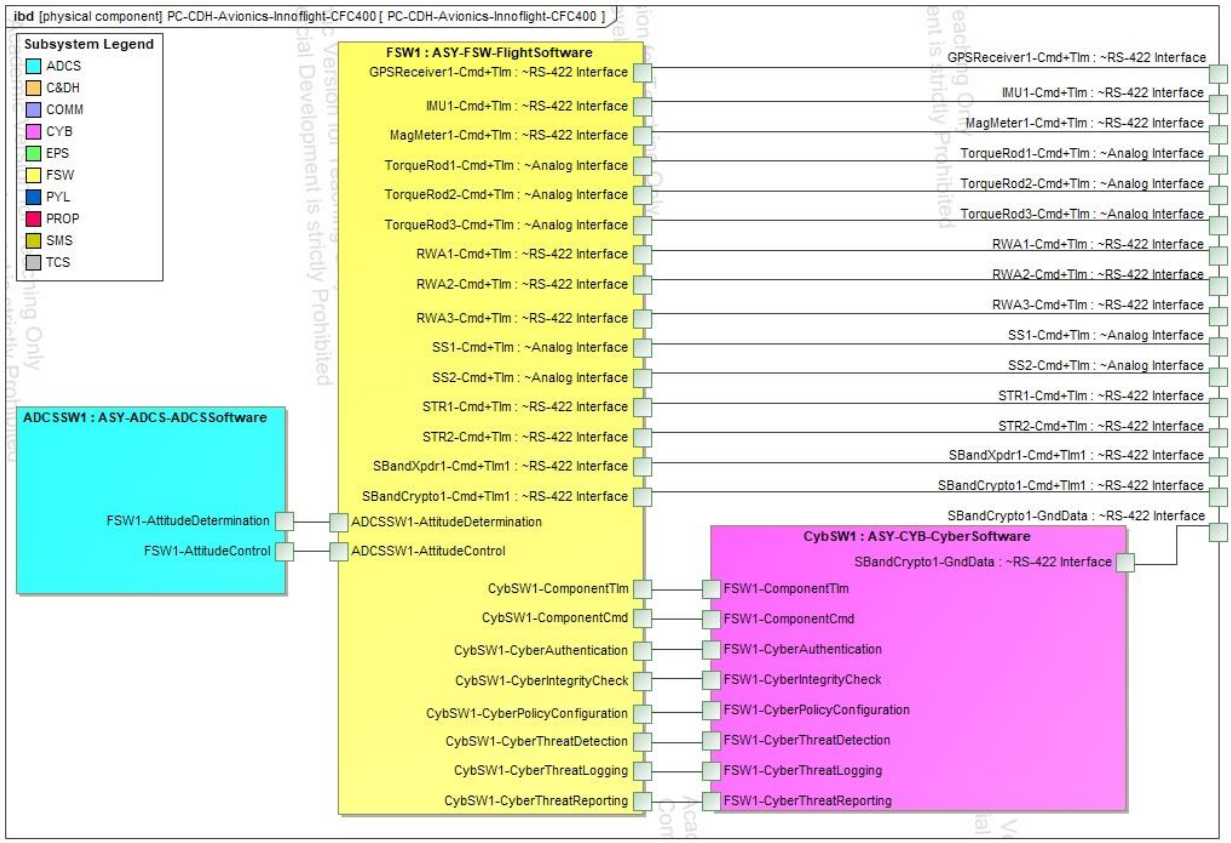

Figure 26: Cybersecurity Software Location

5.7.7 Behavioral Modeling

Several key diagrams must be created at the beginning of the behavioral modeling step. We need to first define the satellite Phases, Modes, and States. The Phases define the major mission lifecycle partitions as a program is designed, built, launched, and operated [202]. **Figure 27** shows the phases for our satellite. These phases were picked because they are relevant to the CONOPS defined in **Section 5.7.1**.

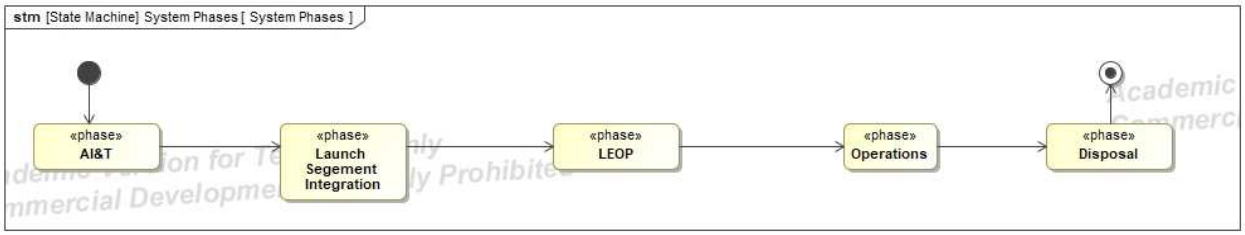

Figure 27: System Phases

These phases can then be allocated under the System Threads defined in **Section 5.7.1**, as shown in **Figure 28**.

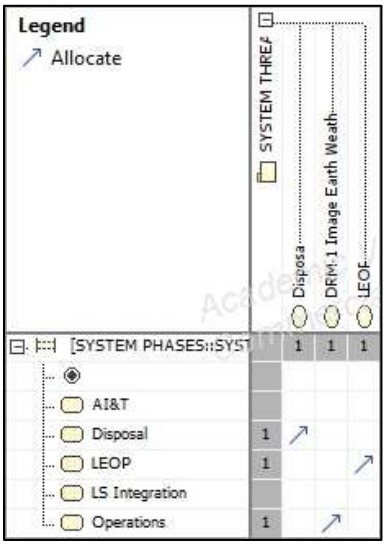

Figure 28 System Phases to System Threads

Note that not all system phases have system threads. There may be additional System Threads for the orphaned system phases, depending on whether or not the MBSE model encompasses the aspects of design planning or only focuses on the operational system. For the purpose of a Class D mission, we recommend only focusing on System Threads that are relevant to how the satellite will function once built.

The next step is to define the Satellite Modes. The Satellite Modes define the major modes of operation of the satellite, and are an abstraction to characterize how the satellite is

functioning for a specific operational configuration [203]. **Figure 29** serves to illustrate a simple Class D satellite Modes Diagram and how the satellite would transition between each mode.

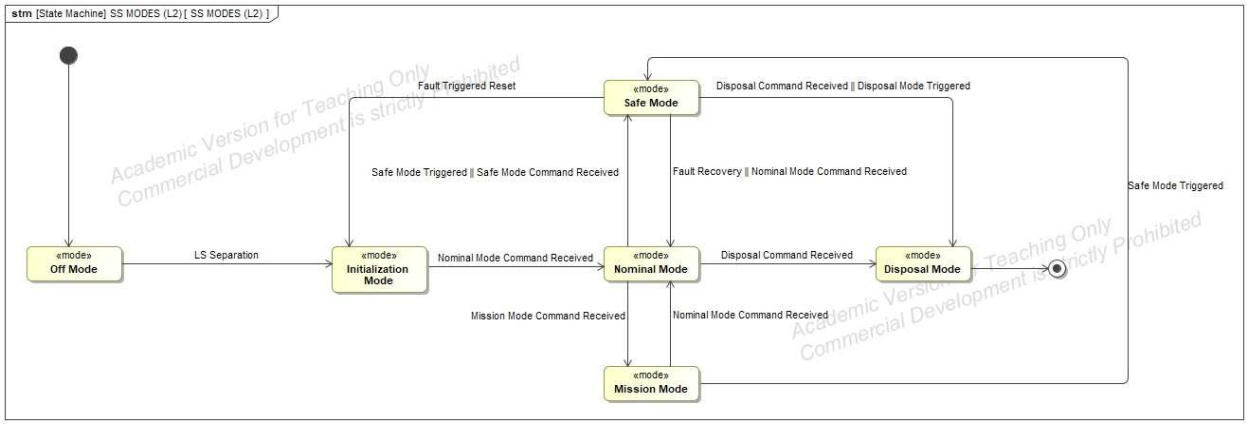

Figure 29: Satellite Modes

As the program is further developed, the specific transitions can be further refined to capture the specifics for how the satellite will operate. The satellite will start in a dormant "launch" mode, then when it detects separation from the launch vehicle it will boot into an Initialization Mode which will be used for satellite checkout activities to confirm the satellite is stable after launch. Once ground operators have checked out the satellite, they put it into a Nominal Mode. The Nominal Mode indicates the satellite is waiting to start the official mission. This is useful for doing orbit adjustments, momentum desaturation, and maintenance. From this mode, the spacecraft can be put into three different modes depending on the next stage of the mission. Mission Mode is where the payload is actively performing its mission, Safe Mode exists in case there is an anomaly the satellite which requires the satellite to be put into a protective mode while troubleshooting and repair activities commence, and lastly Disposal Mode is used at the end of the mission to decommission the satellite.

After their definition is complete, the satellite modes can be allocated to the satellite capabilities, as shown in **Figure 30**, and satellite threads, as shown in **Figure 31**. This tracing shows which satellite modes are applicable for each satellite capability and aids in further decomposition refinement activities as well as in the derivation of the required flight software behavior to satisfy the mission.

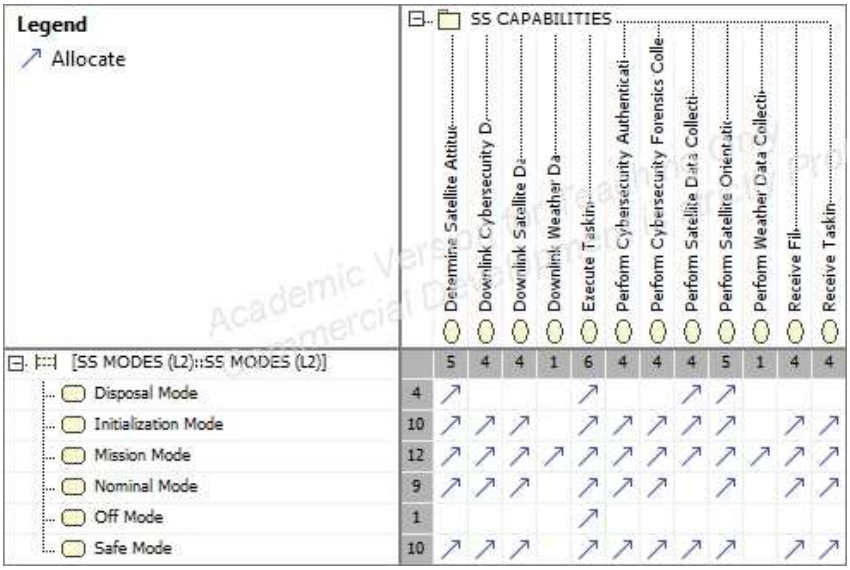

Figure 30: Satellite Modes to SS Capabilities

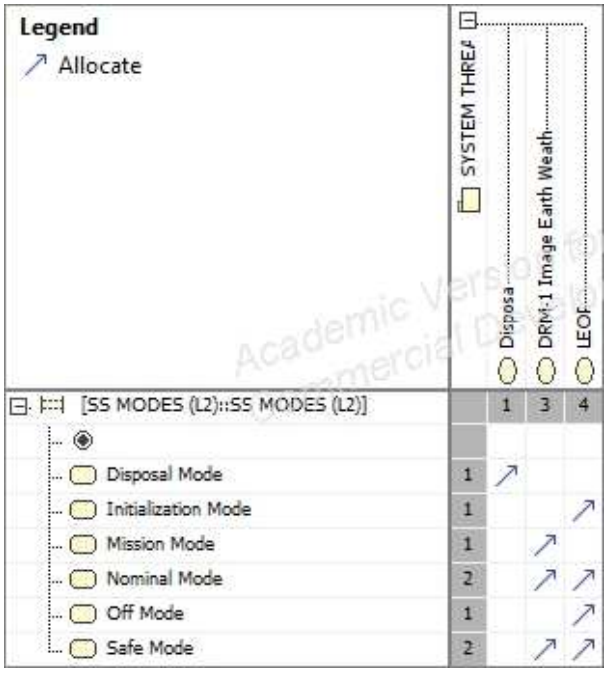

Figure 31: Satellite Modes to System Threads

Finally, the component states are modeled. Because the Modes are at the spacecraft level, they detail how the satellite operates as a whole but do not define how the specific components function to meet each mode. The Satellite States are defined as the operational behavior at the component level [204]. These can be derived from how a supplier programmed the component software (i.e. nominal states, self-test state, error state, etc…) or abstracted out to how the satellite should treat the operation of the component (i.e. on-off-standby, or commanded / not commanded). **Figure 32** depicts a potential state diagram for the cybersecurity subsystem states.

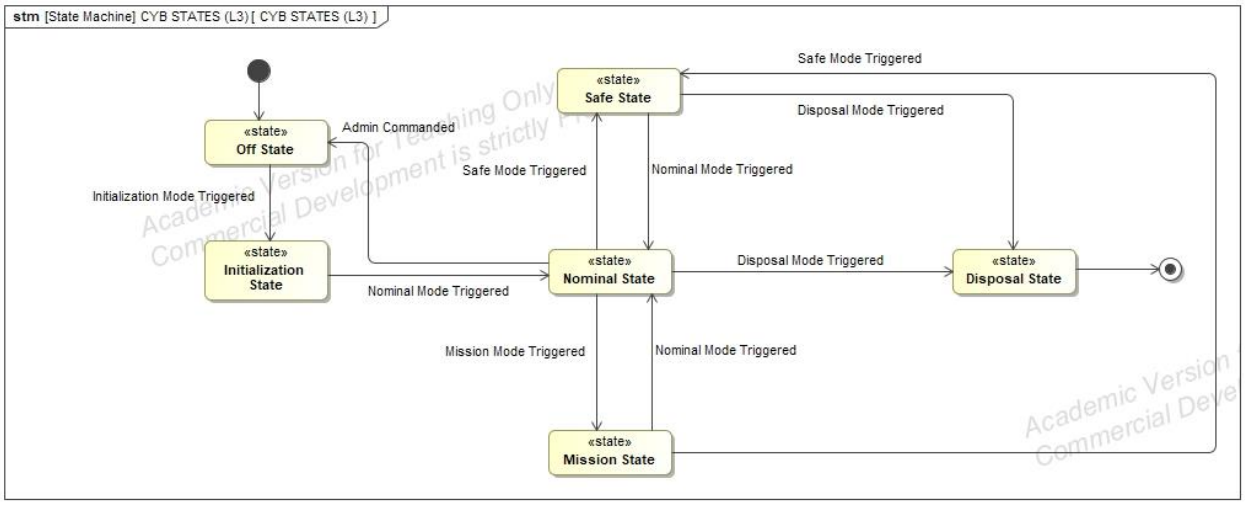

Figure 32: Cybersecurity Subsystem States

This state diagram is an example of how the subsystem operates. As the architecture is further refined these specific states would be extended with Guard conditions to define what triggers Entry and Exit from each state. For this architecture, the cybersecurity subsystem states mimic the satellite modes to enable an adaptable cybersecurity policy. This means that, depending on the satellite mode, there will be a different policy posture that either allows or restricts certain commanding.

Now that we have defined component states, we need to trace them up to the satellite modes, as shown in **Figure 33**. This further ensures that when the satellite is in the Nominal Mode, the Cybersecurity Subsystem is in the Nominal State of protection.

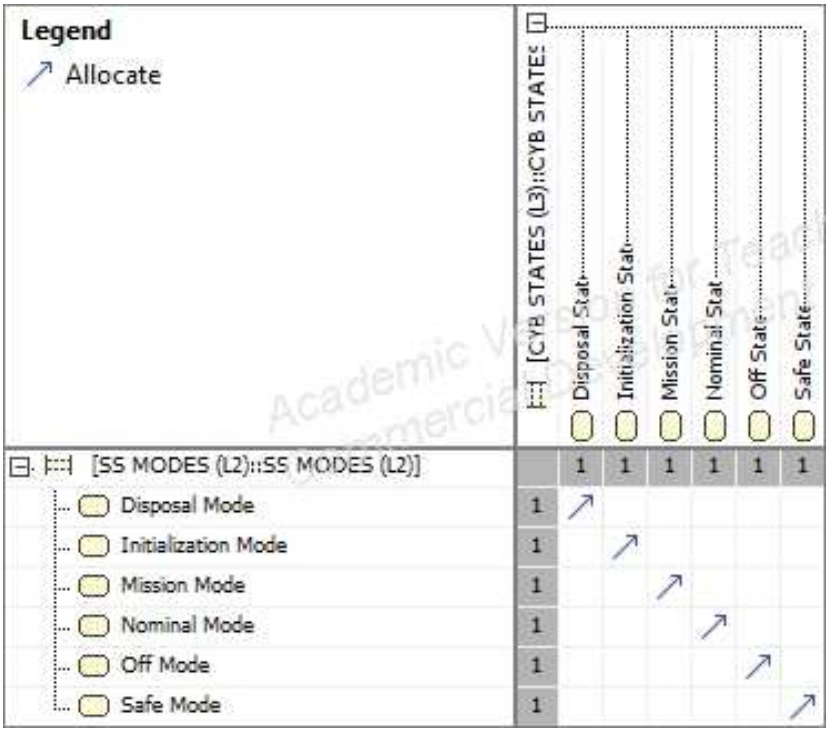

Figure 33: Cybersecurity Subsystem States to Satellite Modes

This completes the final step in modeling the mission phases, satellite modes, and component states, all of which when taken together define the satellite behavior for future decomposition activities for Flight Software to rationalize how Flight Software needs to behave to accomplish each capability through configured modes and states.

The next step for behavioral modeling is to dive down into each subsystem and model how the subsystem functions to meet its activities. This is one of the most important steps for the cybersecurity subsystem because it encompasses the definition of the lower-level behavior and functionality of the cybersecurity subsystem. In order to complete this, we must refer back to **Research Question 1** in Chapter 3 where we defined the cybersecurity threats which are applicable to space systems (see

Table* 3** and **Figure *4), and incorporate this information into an MBSE table, as shown in **Figure 34**.

#	Name	Documentation	Consequence
1	Denial-of-Service	Denial-of-Service (DOS) is when a threat actor has broken into the Space System communication channels to the Satellite and sends numerous commands to its various interfaces in an attempt to lock up the flight computers and/or various components of the vehicle.	• Loss of entire mission • Loss of partial mission (if recovered)
2	Masquerade	Masquerade is when a threat actor is pretending to be a friendly asset. The threat actor can masquerade as a friendly cross-linked satellite or a friendly ground station. Masquerade is possible when a threat actor has broken the security and access controls of a satellite and knows how it operates.	• Loss of entire mission • Loss of partial mission (if recovered) • Loss of sensitive data
3	Unauthorized Access	Unauthorized Access is when a threat actor (intentional) or a ground operator (accidental) compromises the physical security of a friendly ground station to access the control systems, or has masqueraded into a friendly satellite cross-linked to the satellite of interest. The threat actor has broken the security and access controls of a satellite and knows how it operates.	• Loss of entire mission • Loss of partial mission (if recovered) • Loss of sensitive data
4	Replay	Replay is when a threat actor intercepts the communication paths either between Satellites in a constellation or between a Satellite and a ground station. The threat actor can then "replay" that intercepted data in an attempt to compromise the commanding to the satellite, or give a ground operator a false state of being for the satellite.	• Loss of entire mission • Loss of partial mission (if recovered) • Confusion
5	Software Threats	Software Threats come in two categories. The first category is by the people who build the satellite. They could have inadvertently missed a software flaw which can cause the satellite to act in a way that is unintentional. The second category is Malware. If a threat actor has broken the security and access controls (either by Masquerade or Unauthorized Access) it may be possible for them to upload malicious code to the satellite. The malicious code could be obviously adverse or benign and undetected.	• Loss of entire mission • Loss of partial mission (if recovered) • Loss of sensitive data
6	Tainted Hardware Components	Tainted Hardware is when a threat actor has compromised the components of a satellite during their procurement life cycle. Due to the use of COTS components, vulnerabilities in COTS parts are becoming well known and threat actors could hide malicious hardware/software inside the hardware which could compromise the integrated system.	• Loss of entire mission • Loss of partial mission (if recovered) • Loss of sensitive data
7	Jamming	Jamming is when a threat actor overcomes the external interfaces of a satellite preventing it from communication with friendly assets. Jamming can be thought of like a DOS at the RF level.	• Loss of entire mission • Loss of partial mission (if recovered)

Figure 34: Cybersecurity Threats to Satellite

With the cybersecurity threats we defined in **Research Question 1** now integrated into the MBSE model, we can conduct further decomposition within the cybersecurity software. Using the Cybersecurity Subsystem activities defined in ***Figure 18*** we can start modeling the cybersecurity software behavior. **Figure 35** shows the lower-level applications which comprise the Cybersecurity subsystem.

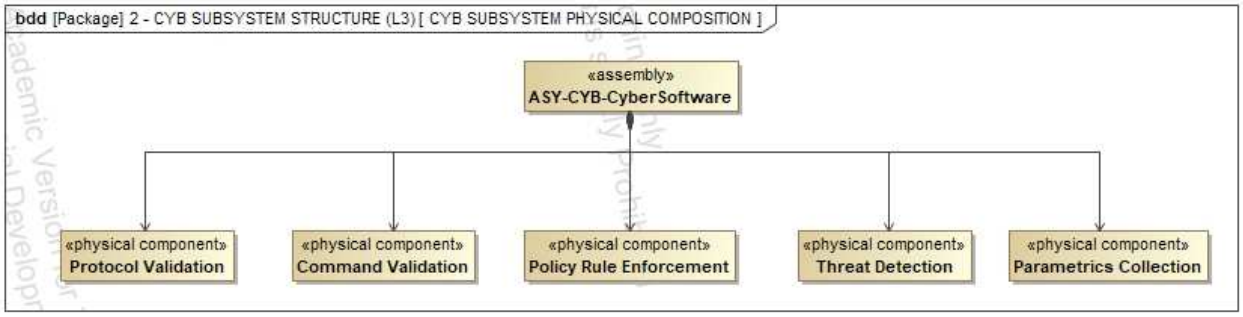

Figure 35: Cybersecurity Subsystem Software Composition

Using this information and the various capability decompositions, we can then construct the overall behavior of the cybersecurity subsystem, as shown in **Figure 36**. This activity diagram can be used to evaluate various threats to the satellite as defined in ***Figure 34***. This will enable team collaboration and further decomposition to determine how each of the Cybersecurity Software components need to operate to achieve their assigned portion of the software mission. This is a simple example of the depth that MBSE can attain in designing a Cyber-secure Architecture. These diagrams can be enhanced with cybersecurity specific viewpoints as the nuances are reviewed and evaluated, enabling a common architecture where engineers can cross-collaborate, deep dive the threats, and model threat specific and use case specific diagrams.

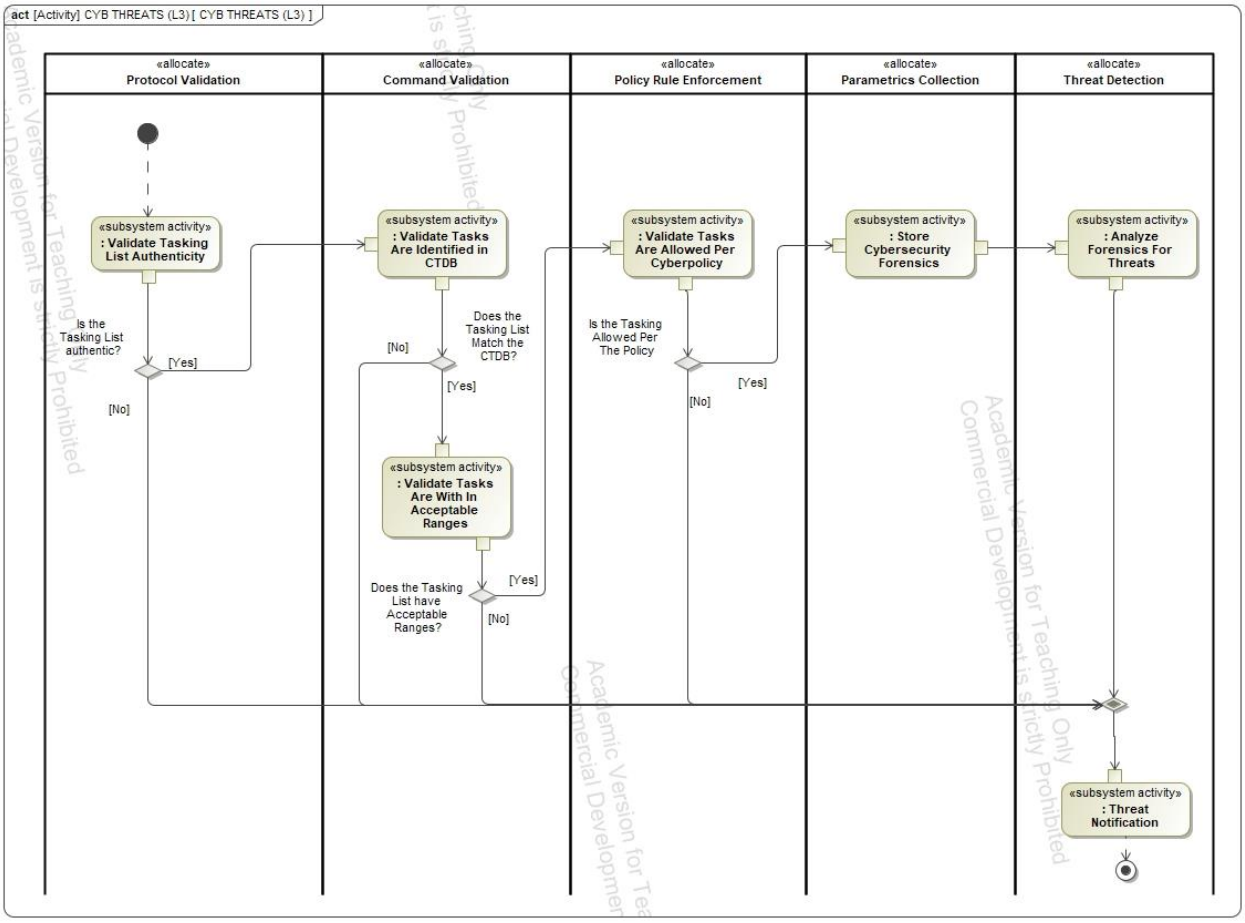

Figure 36: Cybersecurity Subsystem Activity Diagram

5.7.8 Interface Modeling

The last step in this MBSE process is to conduct interface modeling. The Class D approach requires a mix of logical and physical modeling. The logical interfaces are put into the model first. Then, as the logical architecture is refined and the Ground Segment constraints (i.e. leveraging an existing ground segment with defined interfaces) and component software ICDs are further defined, the modeling process transitions to adding the physical interfaces to the model. **Figure 37 i**llustrates how we leverage the architecture from **Figure 25** to determine the specific interfaces, commands, and/or telemetry going across several of the interfaces.

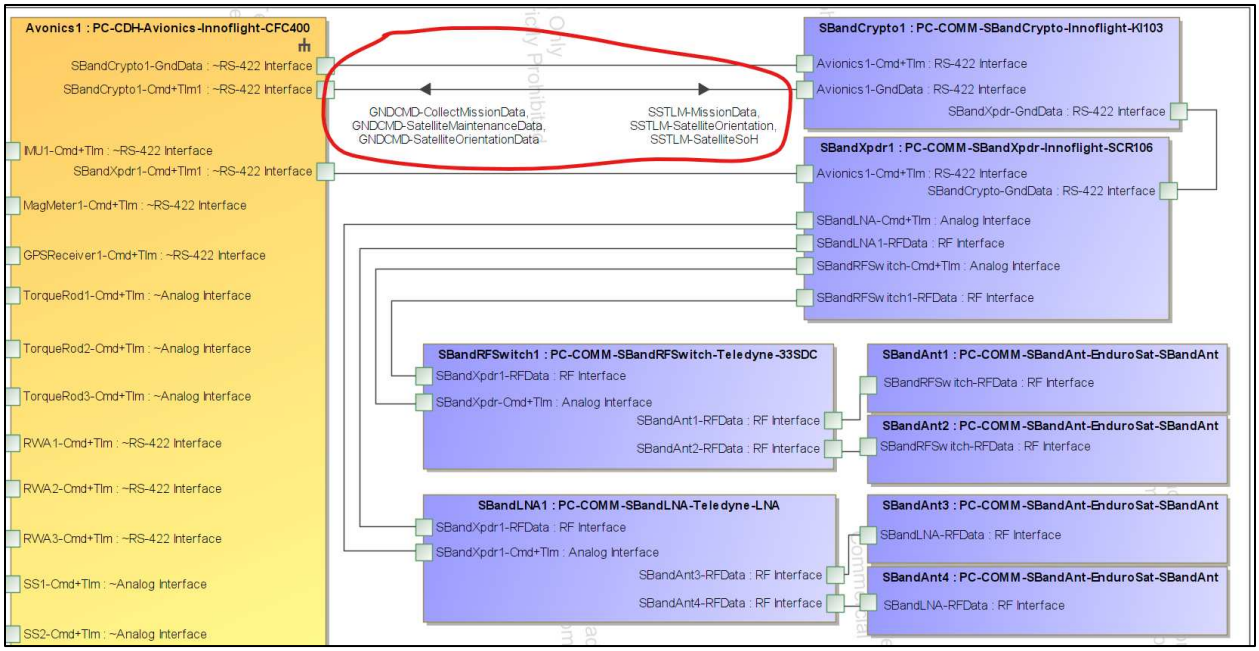

Figure 37: Interface Modeling, With Commanding Identified by The Red Circle

In Figure 37 in the red circle, commanding is received on the satellite and the commands are physically routed from the communication subsystem antenna to the cybersecurity software hosted on the C&DH processor.

Figure 38 demonstrates how commands from the ground are routed through the satellite to the cybersecurity software.

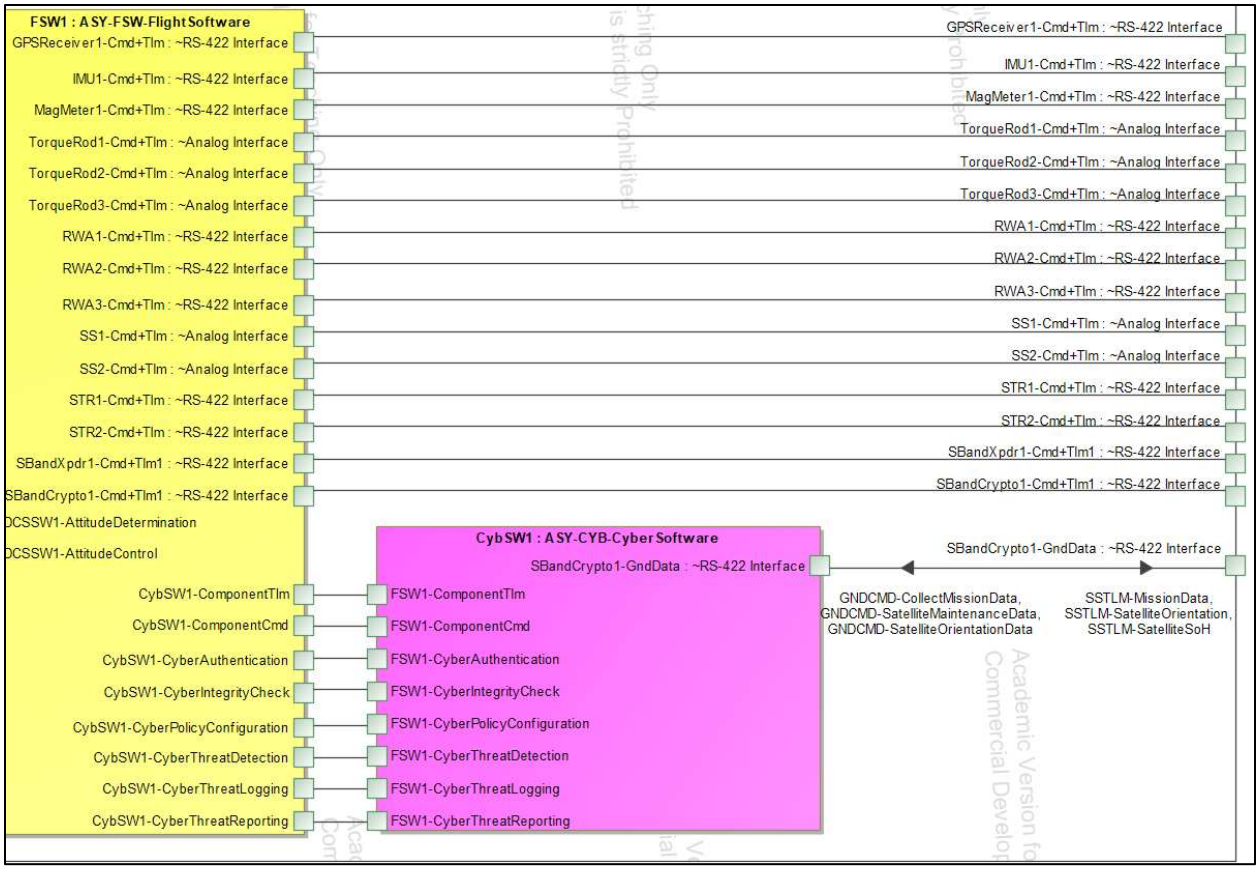

Figure 38: Interface Modeling to Cybersecurity Subsystem Integration

By incrementally repeating this process for all components and subsystems, we begin to create a command and telemetry database as shown in **Figure 39**.

#	Name	Documentation	Command Number	Limit Max	Limit Min	Logical Receiver	Logical Sender	Physical Sender	Physical Receiver
1	GNDCMD-CollectMissionData	This is a ground command to command the satellite to collect mission data with it's payload				SPACE SEGMENT	GROUND SEGMENT		PA-Satellite
2	COMPCTRL-SetRWARPM	This is a component control from FSW to set the reaction wheel speed					FSW Subsystem	ASY-FSW-FlightSoftware	PC-ADCS-RWA-BlueCanyon-RW
3	COMPTLM-RWARPM	This is a component telemetry from the reaction wheel to FSW reporting the reaction wheel speed.							

Figure 39: Command and Telemetry Database

This telemetry in turn can then be applied to the definition of a Space-To-Ground ICD between the Space and Ground segments. Finally, this database can be enhanced with the cybersecurity threats defined in ***Figure 34***.

5.7.9 Model Iteration Throughout A Program Lifecycle

The Cyber-secure MBSE Architecting processes outlined in this dissertation requires incremental iterations throughout the program lifecycle. A designer will not be able to complete all steps in depth in a single pass, but must instead revisit the process as the definition is further refined. When this MBSE approach is applied, it must also be partitioned to match the program lifecycle. For example, a designer will not be able to complete all steps before a program's System Requirements Review (SRR). However, the designer can complete the steps that are important to convey the system architecture necessary for that design review.

Figure 40 shows our recommendation for how a team would create the model and iterate over the lifecycle, including the architecture design milestones which we recommend to complete before the various design reviews that are part of a typical satellite design process. This figure extends **Figure 9** by expanding upon the program's design phases.

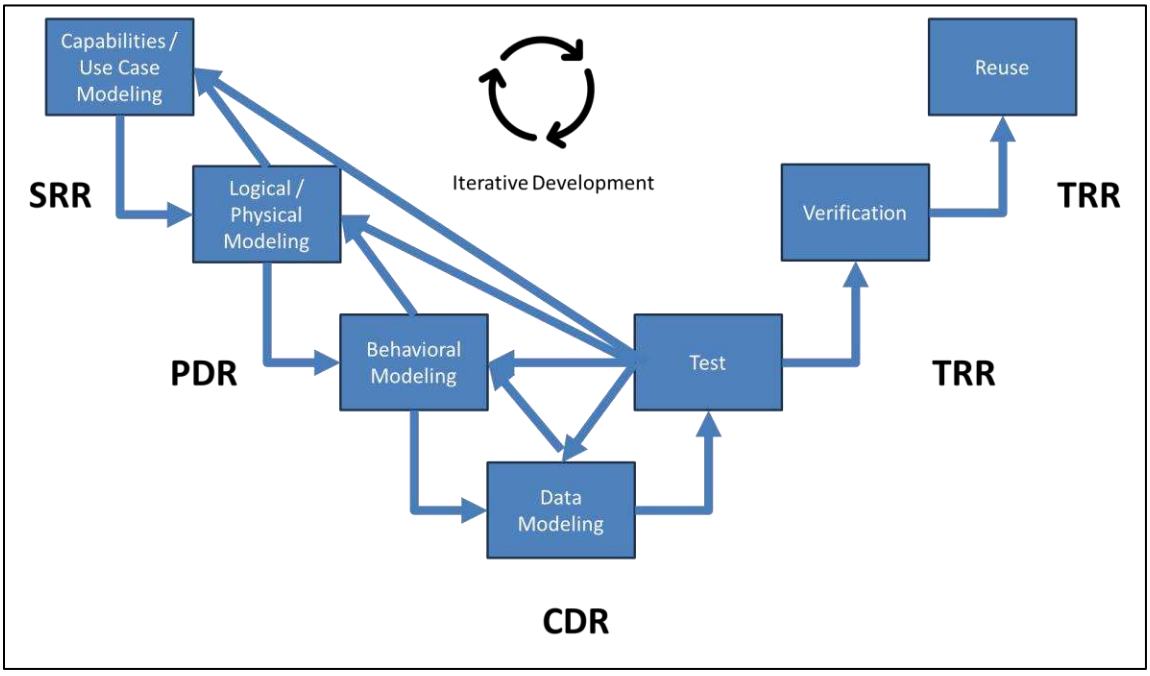

Figure 40: Cyber-secure Architecture Process Iteration

Below are the recommendations for a Class D mission:

- **SRR:** By SRR, the System & Satellite Use Case diagrams will be created to convey the mission. As part of creating these diagrams, the Segment Capabilities will be defined. Lastly, these diagrams and capabilities will be traced to and used to create a baseline for System requirements and draft Segment level requirements.
- **Preliminary Design Review (PDR):** By PDR, the Use Case Diagrams/Capabilities will be further refined to create a baseline of Segment level requirements. Additionally, the initial Logical/Physical architecture will be defined to convey what the PDR architecture is and to enable necessary subsystem analysis. The Segment Capabilities will also be decomposed down to subsystem activities to create a draft of subsystem-level requirements.
- **Critical Design Review (CDR):** By CDR, the Capabilities may be refined as required. However, the build of the Use Case modeling will further refine the subsystem behaviors. The subsystem level requirements will be baselined. The logical/physical architecture will be updated with final component selections and information. Finally, by this phase the component ICDs should be available and can be incorporated into the model. As a result, the initial Segment ICDs can be drafted.
- **Test Readiness Review (TRR):** By TRR, the architecture should be fairly finalized, although there may be minor updates. The bulk of the effort at this point will be finalizing the system ICDs.
- **Pre Ship Review (PSR):** By PSR, the model will be updated with verification artifacts to demonstrate complete traceability from verification artifact to CONOPS at all requirement levels.

As demonstrated with this example we have now defined the CONOPS for the satellite, traced the capabilities down to space segment and subsystem requirements, built the physical architecture of the satellite, and then showed how the physical information transverses the subsystems to arrive at the cybersecurity subsystem. When building the satellite from this architecture, we have clear traceability from a capability to the requirements and components that are responsible for the capabilities. Although this method followed a general architectural decomposition design process, it is possible to further refine the model to more specifics and nuances for how the cybersecurity software will be programmed and operates. This process demonstrates that, similar to software engineers, cybersecurity engineers can follow a process to create cybersecurity software viewpoints in an SSOT that integrates with the system architecture. Lastly, there is crossover between performing modeling and conducting the actual engineering design activities. The model retains the architecture, requirements, and verification, while the engineers use the model to define requirements. The requirements drive what the software needs to do, and finally the verification of the requirements shows that the architecture is satisfied.

5.8 Determining How an MBSE Approach Preserves the Cost/Schedule Benefits of Utilizing MBSE for A Cyber-secure Architecting Process

MBSE is often advertised as a cost-effective / cost-savings approach to defining a system architecture. However, before the savings are realized, there is an initial adoption growth curve at the start of most programs which may see an increase in initial cost on the front end [205]. This curve is due to shifting the organizational processes such as implementing new training for the engineers, establishing model governance, and defining how to coordinate between teams. We have observed that new companies practicing MBSE start to realize efficiency around the

time of a program's PDR when they can demonstrate model artifacts and show how they satisfy the mission requirements.

In the previous section, we conducted MBSE modeling to define a candidate cyber-secure architecting process for a LEO satellite. The last step of this dissertation is to discuss if an MBSE approach, as demonstrated in the prior section, preserves the cost and schedule benefits of utilizing MBSE.

5.8.1 Surveying How Industry Has Measured the Effectiveness of Utilizing MBSE Approaches

When surveying how to measure the effectiveness of utilizing MBSE several questions are brought to mind: How does one define "effectiveness" in regards to a process change? The new process can be something that makes the team's efforts easier – but did it save the company anything in terms of decreasing budget and/or schedule? Or did it save the employees from frustration due to improved coordination at the monetary expense of learning a new tool? Did it save the company in technical competence on subsequent programs at the expense of the initial effort, helping the company to market and propose future designs by being able to leverage a newly-created MBSE model?

To evaluate how MBSE can be integrated into a space system design process and evaluate the cost benefits of applying MBSE, we need to evaluate the state of the field and see how others have tried to answer these questions. This section reviews multiple academic surveys and studies based on recorded industry experiences for this effort.

(Bayer, 2018) [206] at NASA's Jet Propulsion Laboratory (JPL) proposed that a way to measure the effectiveness of MBSE is to measure how well it resolves technical issues. They

studied lessons learned throughout JPL projects and created a list of challenge areas where MBSE could have helped. JPL then had a program where they actively applied MBSE starting with the initial stages of the program. Using hindsight, (**Bayer, 2018**) [206] was able to create a scorecard and objectively evaluate the issue areas to see if and how MBSE impacted this program compared to the old methods on the previous programs. Their study outlined typical systems engineering issues and showing how MBSE can help resolve and enhance the process. However, their results don't translate well to other corporations or programs because they didn't measure schedule or budget. The study shows good lessons learned and can guide how to apply MBSE at other companies but stopped short of stating whether or not other companies could save on cost and schedule. While recording the "lessons learned" is helpful, the omitted cost and schedule benefits information would be essential to understanding where MBSE will bring the most benefits for those who are looking to transition. In addition, it would be beneficial to use a score card that could be applied to the industry with more numerical metrics associated with it, since hard metrics are the crux of the issue of trying to determine the effectiveness of MBSE. The people who are attempting to apply MBSE need to determine a method to quantitatively compare both processes: before utilizing MBSE and after utilizing MBSE.

(**Sanders 2011**) [207] determined a way to measure the effectiveness of MBSE through defects in requirements. Requirements ambiguity is one of the leading causes of program cost overruns and delayed schedules, and **Saunders** was able to show that using MBSE decreased the average defect per "shall" statement in a specification by 68% (**Saunders, 2011**) [207]. MBSE has often been advertised to reduce the impacts of ambiguous requirements by enabling the definition and validation of requirements early in the design life cycle. [161]. The Saunders study proved the existence of an objective metric for evaluating the usefulness of MBSE. In addition,

this can be extended to show that requirements volatility is a source of cost overruns which can be mitigated by MBSE (**Gans, 2018**) [208].

(**Krasner, 2015**) [209] looked at how MBSE impacted the cost and schedule of a project. They used surveys to collect data over six years from thousands of stakeholders. Their research showed that an average cost of delay can be predicted depending on the number of developers on a project, and implementing MBSE can reduce the cost impact on a program by up to 55%. This is a significant finding; unfortunately, the data that went into their study and their paper doesn't go into sufficient details to define exactly what they were evaluating in Systems Engineering and how it could be applied to other companies. For example, the data reveals nothing about the size of the projects: Were the survey respondents working on small projects that could easily adopt MBSE, large programs on which MBSE is more challenging to adopt, or were they examining the industry as a whole and including projects of all sizes? Their study does not also include details on their methodology for translating survey data into cost data. Despite these shortcomings, their study does at least compile and present some research into how cost is impacted, which could serve as a basis for future studies. Their study also brings up an interesting way to measure MBSE effectiveness by looking at the number of people working on a project and the costs associated with risks they encounter throughout the development life cycle.

5.8.2 Surveying Challenges with Adopting MBSE and Demonstrating its Value

These ideas on how to measure the effectiveness of MBSE stimulate the following question: What are the challenges associated with adopting MBSE? Various industry research leads to the answer that it's not a straightforward task to adopt MBSE. (**Chami & Burel, 2018**) [198] performed a study on the challenges associated with adopting MBSE and how to determine

the return on investment. Part of the motivation for their study is partly in common with the questions asked in this doctoral thesis: “Why should I model?” (**Chami & Burel, 2018**) [198]. This question has been getting asked over and over again by different people across the industry, and evidently this question still needs support to be fully and satisfactorily answered. However **Chami & Burel, 2018** [198] point out that, as MBSE has been becoming more widely accepted by the technical community, this question has been shifting to one of “How?” “How should I model?” or “How should I use and manage models efficiently?” (**Chami & Burel, 2018**) [198]. To conduct their study, **Chami and Burel** conducted a survey through the technical community to determine where people have been experiencing MBSE adoption challenges, creating a list by interviewing MBSE experts throughout the community. The results of their study conclude that two factors lead to the most challenges in adopting MBSE: “Purpose and Scope Definition” and “Awareness and Change Resistance,” both of which in our experience are the typical reasons cited for companies experience challenges with adopting MBSE.

Because MBSE requires a different design approach culture, (**Kim et al., 2019**) [210] looked at the human element in the MBSE design process and how the tools and processes influence its implementation. They used a Human-Centric design framework and applied it at JPL to evaluate the implementation of MBSE. They concluded that in order to integrate MBSE practices into an organization, the existing Systems Engineering processes at the company must be examined and the MBSE methodologies tailored such that the Systems Engineers can understand the benefit of their use. This is a great point because it further defines the exact challenges to adopting MBSE. MBSE needs to be crafted around the specifics of the company in order to ease the transition. Otherwise, a company will run the risk of an incomplete transition that loses support and interest of the company leadership. This is why it’s important to also

determine ways to show the value of MBSE and how it can benefit each company for their specific use. Typically, the system is designed using the traditional document-centric tools that system engineers use, and it then becomes increasingly complicated due to the evolution of complex designs across multiple tools (**Kim et al., 2019**) [210]. Like **Kim et al.**, we have also observed that complicated MBSE models which are difficult for the average user to understand lead to maladaptation and even rejection of the whole MBSE process. If a single engineer creates the model, but is also the only person who can decipher the model, then the model is not useful to the program and the overall MBSE process (not just that particular model) is highly likely to be rejected.

5.8.3 Evaluating Our Model and Defining How Our Approach Provides Value

The studies in the preceding sections all demonstrate in unique ways that the industry is struggling to measure the value of an MBSE approach. Marketing techniques and logical defense of the MBSE tools ensure a perception of value will come from their use; however, due to differences in actual adoption and utilization, it is difficult to compare from company to company and assign a quantitative measure of actual value. **Campo et al.** [211] evaluated the perceived value of MBSE approaches by performing a literature review of 60 academic sources to determine if marketed MBSE claims were backed by actual experience. They found that about 47% of the claims were author opinions and the majority of documented value wasn't based on measured metrics. Essentially, they found that a vast majority of literature promotes the perception that MBSE adds value to programs while the factual evidence to back these claims is lacking. This adds emphasis that industry needs more metric studies to measure the success of MBSE methodologies, which is easier said than done because in order to conduct a true MBSE value study, the participating companies must first have an pre-established document-based

approach, collect metrics on this traditional process, and then finally convert to an MBSE approach, collect more metrics on the MBSE implementation, and finally compare the two sets of metrics to measure the ROI. Not many companies fit such a robust definition of using a traditional process and then converting to an MBSE process – most companies (and engineers) who are used to the traditional process resist the change to the new MBSE process which adds another confounding and difficult-to-metricize factor, while new, younger companies with more amenable workforces will jump straight into an MBSE approach without creating a traditional approach even for the sole purpose of collecting metrics. No matter their method experiences, companies will also vary in their definition and collection of metrics, and will not adapt MBSE methodologies in exactly the same manner, making it difficult to compare and contrast the exact investment into MBSE approaches [212]. As a result, companies will experience common and different challenges that would skew the metrics from company to company and make it difficult for an independent researcher to make a determination on the exact success of MBSE vs the traditional approach. A spot of hope, however: **Campo et al.** [211] did find that the very limited academic sources which had measureable metrics to support their claims did actually indicate some added value when using MBSE.

Despite the limited studies which show that MBSE approaches preserve the cost and schedule benefits, there is still a disconnect between company leaders and their employees within industry. Company leaders believe there is a benefit whereas the majority of engineers that are actually completing the work, and learning the new tools, often believe the opposite [213]. This can be attributed to MBSE adoption challenges such as employee entrenchment and resistance to trying new things as well as trend-chasing company executives and customers, which serves to emphasize the importance of creating an adoption roadmap.

We argue that the initial adoption of MBSE does cost more than a traditional systems engineering approach due to training, governance creation, and initial model rationalization. Only in subsequent programs does the real value begin to shine. In our experience in actively applying MBSE to space systems, the value we've seen to the company is in the reuse of the base model from program to program.

(**Duffy et. al. 2021**) [214] researched how to determine the return on investment of value for companies utilizing MBSE approaches for systems engineering. Their research examined many parameters such as company size, engineer experience, and operational complexity for conducting systems engineering tasks. By comparing these parameters against task complexity (in dollars to complete), (**Duffy et. al. 2021**) [214] were able to show that MBSE tools can have a significant return on investment (up to 1260%). This study is significant for its definition of the cost benefit of utilizing an MBSE approach for space engineering. This study provides some data points beyond the typical subjective benefits that MBSE promotes, which are efficiency and reduced cost/schedule for programs when shifting from document-based to model-bases engineering. However, the research also recognizes the lack of standardized task performance metrics and engineer experience criteria used in the model calculations, and it emphasizes that there is an investment period when shifting to an MBSE methodology during which companies will not see immediate ROI results. Similar to the **Duffy et. al.** study, we have also observed that there is an upfront cost to buy the tools, train people, and built the architecture for the first program which only future programs can benefit from.

Our approach to an MBSE cyber-secure architecting process provides two key values to enable a return on investment:

1) Promotes team collaboration to minimize requirement defects, and.
2) Promotes model reuse and productization of the architecture.

5.8.3.1 Promotes Team Collaboration to Minimize Requirement Defects

Traditional Systems Engineering approaches inherently promote isolation between the systems engineers and various satellite subsystem engineers. Some programs will have the Systems Engineers manage requirements, and there may or may not be seamless collaboration with the design engineers, leading to disconnects and downstream error corrections. Similar to the separation of spacecraft engineers and cybersecurity engineer as explained in Research Question 2, this isolation is prevalent within the designer's realm.

While MBSE is a new kind of systems engineering, it is not a replacement for systems engineering. Its purpose, rather, is to enable good and efficient systems engineering. An MBSE approach promotes an integrated development process with multiple stakeholders to ensure there is a reduction in team ambiguity in the understanding of the requirements because MBSE explicitly links the requirements directly to the system design and architecture. [215]. Our approach to MBSE model content generation requires constant collaboration and teamwork to rationalize about what the architecture must accomplish to meet the mission, and how each component fits into the bigger picture. Our proposed approach encourages communication between the members of the whole design team. It is not important who leads the design decomposition process; whether SV designer, Security engineer, or someone else, this person or team must ensure alignment and agreement across the entire team or teams as they define and decompose the architecture.

A challenge we've experienced when applying MBSE approaches to space programs is using the MBSE model as an SSOT and determining how the model is populated with the architecture. An important first step is to recognize that MBSE is not a one-size-fits-all solution. Instead, the process needs to be tailored to the specific company needs and values. MBSE is still largely novel for spacecraft architecting due to previously well-established document-based tools. A synergistic approach needs to be identified so that the MBSE tool isn't thought of as replacing external analysis tool; rather, it serves as an SSOT for the inputs and outputs of each disparate analytical tool as a Digital Engineering environment.

The research papers we reviewed in **Section 5.8.1** looked at the different phases of MBSE adoption and found out that the majority of challenges occur during the initial adoption phase during which a company does not have much previous MBSE experience. This shows the importance of tailored MBSE adoption roadmaps. When programs are initially adopting MBSE, a key challenge we have observed includes a model driven by industry experts who expect MBSE novices to suddenly and expertly use the model. This results in a detailed SSOT model that only one or two people understand while the rest of the people on the program are unable to use or even decipher the model, rendering the model not beneficial to the overall success of the program. The efficiency of utilizing MBSE therefore depends on the experience level and background of the system engineers who are expected to use the tool, and how the company has implemented the MBSE process [216].

In our experience, by offering a base model that the majority of the team can comprehend and which promotes constant collaboration, our MBSE approach will help improve requirement quality which will minimize program cost overruns due to an increase in overall understanding of the requirements earlier in the program life cycle across the program engineers. One

additional benefit of our MBSE approach is that requirement defects are integrated directly with the architecture process. At the system level, the requirements are directly linked to the system use cases, and at the subsystem level the requirements are linked to their applicable hardware components. This improves upon the document-based approach where requirements are housed in documents separate from the architecture. Under our MBSE approach, as the program progresses through its iterative development approach, changes to requirements can be more easily evaluated for architecture impact.

5.8.3.2 Promotes Model Reuse and Productization of the Architecture

To understand the benefit behind MBSE model reuse, it's important to recognize the required initial investment when adopting an MBSE approach. Companies must be willing to provide the necessary training for engineers to utilize the new MBSE tools; willing to recognize how imperative it is to establish governance for how they as a company want to operate and drive consistency into the MBSE modeling approach; and finally willing to acknowledge the need to model in such a way that the vast majority of new and experience engineers see the value of the model and can understand where information is located and how to retrieve the information. MBSE approaches reduce the unrealized costs associated with various stakeholders inadvertently recreating system interfaces and artifacts, if they actually treat the MBSE model as a single source of truth, which in turn becomes a cost savings [217].

Using MBSE methods allows companies to benefit from the reuse of the base model and process from program to program because after the base architecture has been created, it can be leveraged and tailored to suit later individual programs [218]. Reusing the model shifts the narrative from "creating from scratch" to "performing gap analysis against existing architectures and artifacts." If a company wants to productize the spacecraft architecture, having an SSOT

model enables faster design iteration. The existing model can be used to rapidly modify the design parameters from the top down or bottom up and, since all the model elements are linked, it's easy to evaluate impact to the design and the overall ripple through the architecture [219]. In additionally, another key benefit is that, because the model is an SSOT, the architecture and artifacts are in one location regardless of staff overturn, rather than being spread out on various desktops, network drives, cloud share drives, e-mail, and of course lost forever to the churn of staff departure and new hiring.

We propose a productized MBSE cyber-secure architecture method which is driven by capabilities and defining common product components. There are two key situations that drive the depth of a gap analysis:

- The company is leveraging an established architecture but it is changing the payload for a new mission.
- The company is building a completely new satellite and wants to leverage an existing architecture as a starting point.

In our opinion, the approach to model reuse is the same; however, the depth of a gap analysis may change. The gap analysis starts with reviewing and modifying the established CONOPS & DRM diagrams to determine which capabilities are relevant versus which need to be updated. If a company is leveraging an existing satellite architecture, the bulk of the satellite capabilities are going to be common across programs. For example, the capability to receive a command or execute a tasking link will trace to the same (or similar) requirements and same types of components no matter which actual program is the focus of the work. Per our method of modeling, these satellite capabilities are mapped throughout the architecture from requirements

and subsystems to component behavior, enabling full updates to capabilities throughout all affected model elements. The other benefit of the MBSE approach is to start creating a common components database. For example, if there are two kinds of Reaction Wheels that could be used on a given satellite, these components can be modeled in such a way that the MBSE block which represents the Reaction Wheel contains all the relevant information, making it easy to replace one block for the other and update the linkages. As a result, the reuse of conveying a mission CONOPS from the top level down to the component is retained and the external analysis tools that utilized the SSOT information can easily retrieve updated model parameters.

5.8.3.3 Example of Model Reuse

To demonstrate the reusability of the MBSE model from **Section 5.7**, we will model a new mission and demonstrate a few key diagrams to show the value of this approach for companies utilizing MBSE. For this example of model reuse, we must make a few assumptions in order to define the context of the reusability:

- The mission demonstrated in **Section 5.7** has been successfully launched and the MBSE model is sufficiently populated to define the architecture.
- The new satellite program is also a Class D satellite.
- The company is building this satellite as a common satellite product line. This means that the architecture of the satellite (minus payload) is fairly consistent between missions. If the satellite requires component selection modifications, these modifications will be evaluated through gap analysis in the MBSE model.

In **Section 5.7** we demonstrated a satellite that images the earth weather. For this mission, we will demonstrate a science satellite that measures solar radiation. For the new mission, we

will pull that earth weather satellite model as the baseline and leverage the existing model elements to create the solar radiation mission. **Figure 41** shows that we have created a new DRM use case diagram to show the primary mission of the Solar Weather satellite.

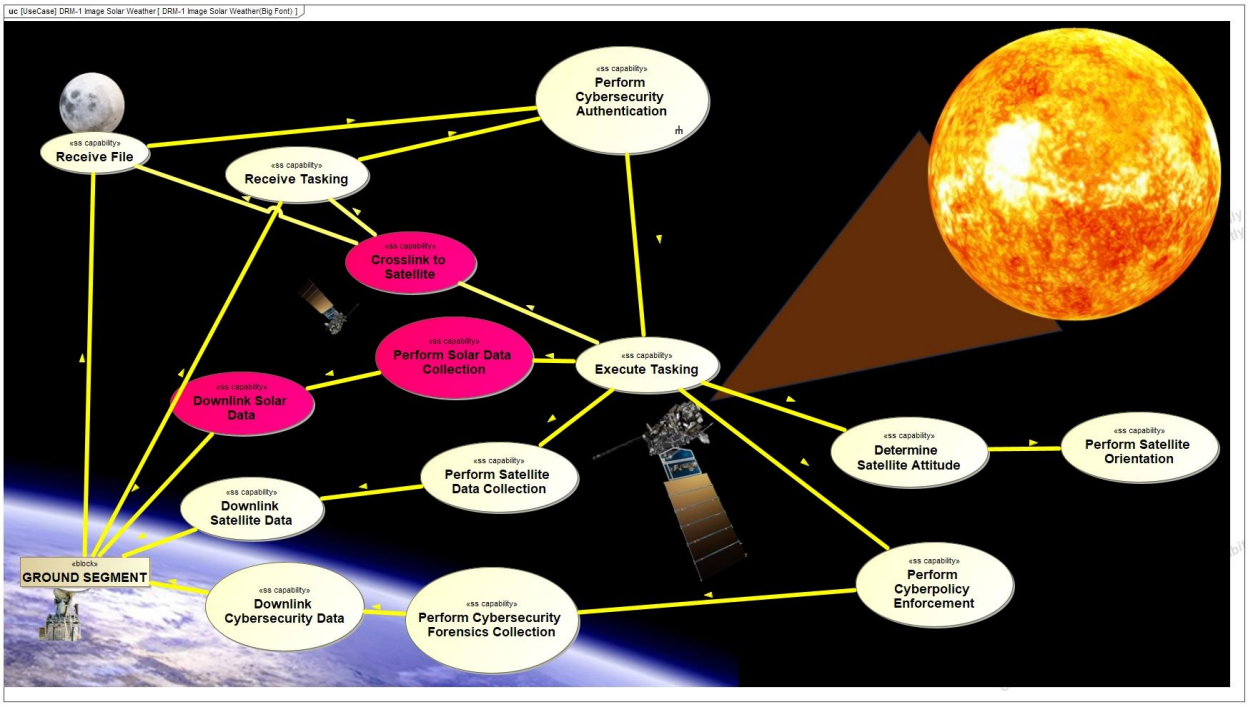

Figure 41: Image Solar Weather Use Case Diagram Representing the new Satellite Program

This Use Case Diagram depicts the satellite procedure when the Ground Operator sends a command to the satellite to image solar radiation. Upon receiving the command, the satellite will conduct operations to verify the authenticity of the command through a cybersecurity authentication gate, then orient its position, take a series of radiation measurements, and finally downlink the data to back to the Ground Segment on Earth. This Use Case diagram is similar to **Figure 11** because the mission is leveraging the same architecture. The main difference is the mission-specific capabilities to collect and downlink solar data rather than weather data. However, to further demonstrate how the prior architecture can be leveraged, in this example we

added the new capability of cross-communication with the weather satellite from the prior mission. This allows more flexibility of the mission depending on the availability of the ground stations. We will now walk through the same decomposition from **Section 5.7** to show how we'd modify the model elements.

Adding these additional capabilities for communication requires us to modify the Use Case description for the "Perform Cybersecurity Authentication" capability as shown in **Figure 42.**

#	Name	Documentation
12	Perform Cybersecurity Authentication	DEFINITION: This capability will serve as adjudicating uplinked tasking or files to the satellite to ensure it's authentic and correctly formatted for how the satellite is intended to operate. The sattelite will validate that all commanding and tasking authentic as define in a on board database of allowable commanding, allowable ground stations, and allowable file signatures. INITIATING ACTOR & EVENT: Sattelite either receives a tasking or file from the ground. PRE-CONDITIONS: Sattelite is operational and in a state that it can receive tasking and/or files from the ground. SCENARIOS: 1. Tasking uplinked from ground segment 2. File uplinked from ground segment 3. Tasking uplinked from crosslink 4. File uplinked from crosslink

Figure 42: Modified Perform Cybersecurity Authentication Use Case

We've added two additional scenarios: Scenario 3) "Tasking uplinked from crosslink" and Scenario 4) "File uplinked from crosslink." These scenarios were added to provide cybersecurity authentication to crosslinked satellites.

For the purpose of this example, none of the cybersecurity space segment (see **Figure 13)** or subsystem requirements (see **Figure 19)** need to be changed because these requirements were written to cover overall cybersecurity behavior and were specific to the external interface on the

spacecraft. Next, we updated the Course of Events (see **Figure 12)** for the Perform Cybersecurity Authentication capability to include the two new scenarios as, shown **Figure 43**.

#	Name	Documentation
1	Perform Cybersecurity Authentication	SCENARIOS: 1. Tasking uplinked from ground segment 2. File uplinked from ground segment
2	S1 - Tasking uplinked from ground segment	SCENARIO DESCRIPTION: The satellite receives a tasking list from the ground segment and the satellite processes the tasking list for cybersecurity authentication prior to accepting the tasking. COURSE OF EVENTS: 1. Satellite receives a tasking list from the ground segment. 2. The Satellite sends the tasking list from the communication subsystem to the cybersecurity subsystem. 3. The cybersecurity subsystem process the tasking list to ensure all commands are valid and authentic. 4. The cybersecurity subsystem sends the tasking list to flight software for further task execution.
3	S2 - File uplinked from ground segment	SCENARIO DESCRIPTION: The satellite receives a file (either software updates or configuration files) from the ground segment and the satellite processes the files for cybersecurity authentication prior to accepting the file. COURSE OF EVENTS: 1. Satellite receives a file from the ground segment. 2. The Satellite sends the file from the communication subsystem to the cybersecurity subsystem. 3. The cybersecurity subsystem process the file to ensure all commands are valid and authentic. 4. The cybersecurity subsystem sends the file to flight software for further task execution.
4	S3 - Tasking uplinked from crosslink	SCENARIO DESCRIPTION: The satellite receives a tasking list from a crosslinked satellite and the satellite processes the tasking list for cybersecurity authentication prior to accepting the tasking. COURSE OF EVENTS: 1. Satellite receives a tasking list from a crosslinked satellite. 2. The Satellite sends the tasking list from the communication subsystem to the cybersecurity subsystem. 3. The cybersecurity subsystem process the tasking list to ensure all commands are valid and authentic. 4. The cybersecurity subsystem sends the tasking list to flight software for further task execution.
5	S4 - File uplinked from crosslink	SCENARIO DESCRIPTION: The satellite receives a file (either software updates or configuration files) from a crosslinked satellite and the satellite processes the files for cybersecurity authentication prior to accepting the file. COURSE OF EVENTS: 1. Satellite receives a file from a crosslinked satellite. 2. The Satellite sends the file from the communication subsystem to the cybersecurity subsystem. 3. The cybersecurity subsystem process the file to ensure all commands are valid and authentic. 4. The cybersecurity subsystem sends the file to flight software for further task execution.

Figure 43: Updated Course of Events to Perform Cybersecurity Authentication

For the new scenarios we will model an Activity diagram for the Tasking uplinked from cross link, as shown in **Figure 44**.

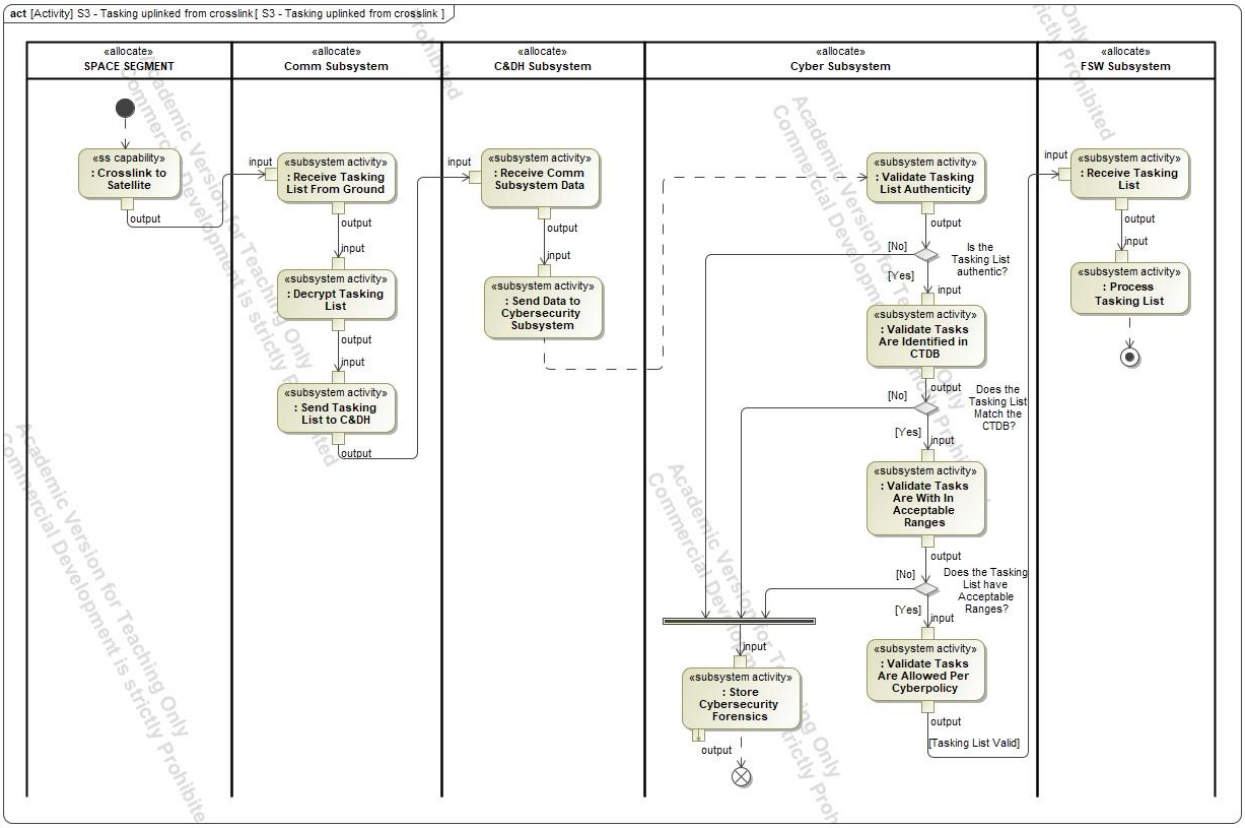

Figure 44: Activity Diagram Depicting Tasking Uplinked From Ground Segment

The key difference between the new mission as depicted in **Figure 44** when compared to **Figure 18** from the old mission is the initial node. In Figure 18, the tasking comes from the Ground Segment, while in Figure 44 the tasking originates at another satellite within the Space Segment. Otherwise, the scenario flow on the two missions is identical, demonstrating recyclability of the model on the new mission with minimal changes.

The next step is to modify the Logical/Physical Modeling to include the new payload. As a result of the decomposition of the new "Crosslink to Satellite" capability, we must update the communication subsystem to add an additional switch and low noise amplifier to allow for additional antennas so that we can talk with the ground segment and crosslinked satellite without having to reorient the satellite for each kind of communication. Otherwise, after analyzing the

performance needs of this new mission, we determined that the remaining components didn't require change and could be duplicated from the weather satellite model. The physical components from **Figure 22** were updated to reflect these changes, as shown in **Figure 45**.

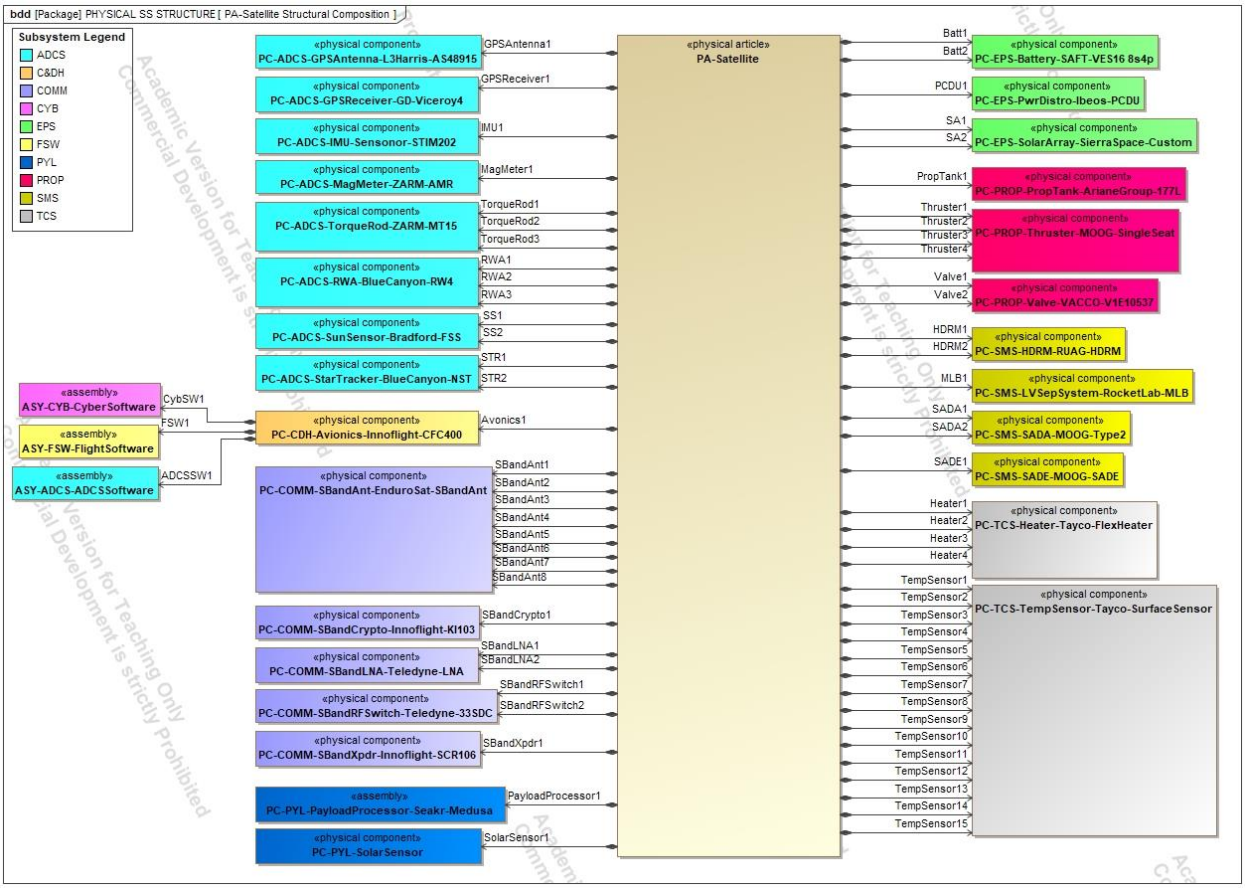

Figure 45: Solar Monitoring Satellite Physical Components

The next step is to update the Communication Subsystem Physical Architecture shown in **Figure 46**, to reflect the additional switch and antennas.

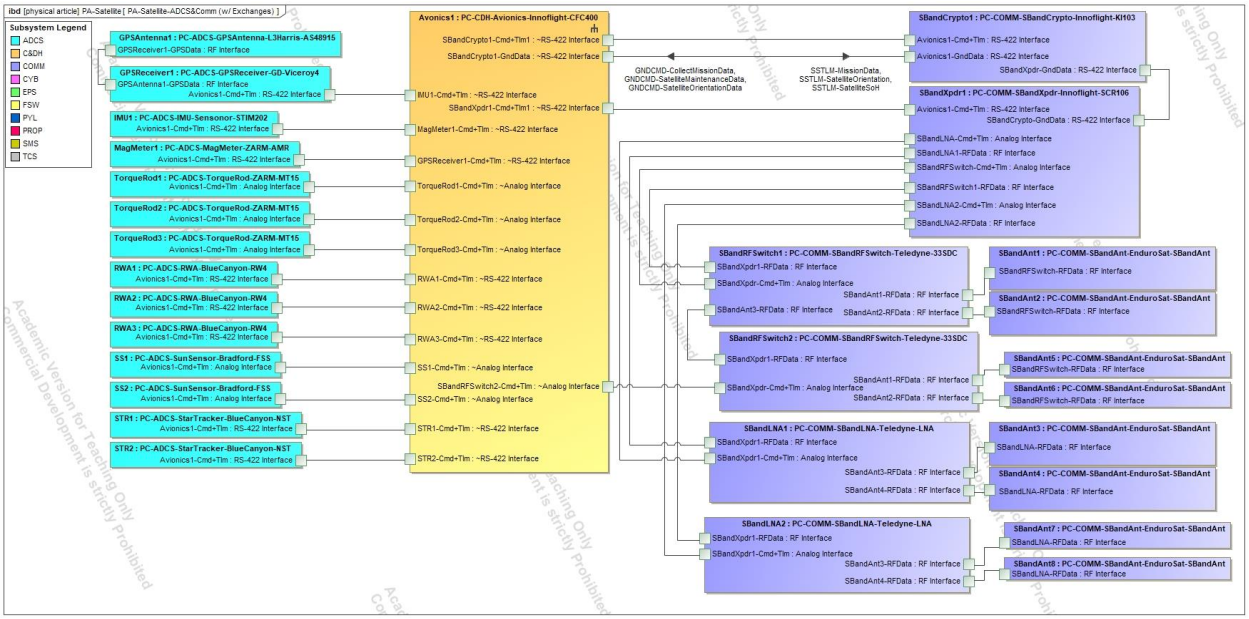

Figure 46: Solar Monitoring Physical Architecture (Comm and ADCS)

Since we didn't add an additional S-Band radio or encryption device we don't need to modify the Cybersecurity Software Location diagram from **Figure 26.** For the purposes of this example mission, we did not change the Flight Software modes or the Cybersecurity Subsystem states, meaning that we don't need to make any changes to the satellite mode in the Behavioral modeling shown in **Figure 29**, or to the Cybersecurity Subsystem States shown in **Figure 32**. Since this is a follow-on mission, the Cybersecurity Threats shown in **Figure 34** should be reevaluated to determine if the cybersecurity subsystem needs to be updated for new threats, but the Cybersecurity Subsystem Activity Diagram shown in **Figure 36** doesn't require updating since the basic operation is going to be leverage for this mission.

5.8.3.4 Measurement of Model Reuse

The measurement of the value proposition of our model reuse example is dependent first on the company utilizing the MBSE methodology as defined in **Section 5.7**, and also on how they've designed and marketed their satellite product line.

For example, the example company of our research project leveraged a common satellite product line to build the satellites for the Earth Weather Observatory and Solar Weather Observatory. In order to maximize their return on investment and reduce the repetition of engineering for customers, they decide to offer a well-defined payload Size, Weight, and Power allocation that customers need to stay within. This allows the company to minimize changes to the satellite architecture from program to program, and provide a best value service.

This example company has been leveraging MBSE for their satellite architecture. Truly measuring the value of model reuse from program to program is difficult without having metrics from a well-established program that has spent years, and many personnel-hours, to develop the model throughout an entire program lifecycle. Leveraging what we have built in our MBSE model, we will use approximation to size our model similarly to a program that has completed its lifecycle.

Based on the cybersecurity subsystem modeling we conducted, we would expect 100% model reuse between programs at this company. The company would leverage the same flight software and cybersecurity software from program to program. Since the cybersecurity threat landscape is forever evolving, the company would be expected to conduct continuous improvement to this portion of the model to adapt to new and emerging threats. Even so, the overall architecture approach for how the cybersecurity operates wouldn't change, just the specific threats to look for. From an MBSE perspective this could be either modifying existing SysML blocks that represent the threats, or adding more SysML blocks for new threats. This would also drive additional cybersecurity penetration testing development and additional test cases to verify the satellite can protect against the new threats. For the purposes of this example, we are going to assume the cybersecurity subsystems between the Earth Weather Observatory

and the Solar Weather Observatory is identical. So for a mission like the Solar Weather Observatory, we'd expect 100% reusability for these model elements.

We have 1660 elements in our basic MSBE model. This encompasses all the blocks, diagrams, and linkages within the model. For the purpose of this modeling effort we focused on the cybersecurity subsystem and only modeled it to a pre-PDR level. Of the 1660 elements, roughly 304 of those are specific to the Cybersecurity Subsystem. Based on our experience we estimate that cybersecurity subsystem is only modeled to about 20% of the depth required for a complete Class D program since we lack specific hardware ICDs and communication protocols. Extrapolating from that, we would expect about 1520 elements specific to the Cybersecurity Subsystem in a completed model. Assuming each of the subsystems are of similar complexity on average, there are 10 subsystems in total including the cybersecurity subsystem. So, this would put us at approximately 15,000 model elements for the Space Segment.

Comparing the MBSE model of the Earth Weather Observatory to that of the Solar Weather Observatory, we added new space segment capabilities and a new payload. For the purpose of this value proposition, we will assume similar complexity between the payload subsystems. The main changes to the model therefore were the new capabilities and components added. Comparing the two MBSE models, we determined roughly 194 blocks were either added or modified for the new capabilities for the payload. Applying the estimated 20% completion rate and factoring up this would equate to about 970 model elements plus the payload subsystem for 1520 elements which would comes to a final total of about 2490 new / modified model elements.

This demonstrates that when taking the overall architecture in our example architecture and leveraging the Earth Weather Observatory to design the Solar Weather Observatory, as shown in **Figure 47**,we project roughly we can reuse ~83% of the MBSE model, where 83% of the model elements are unchanged.

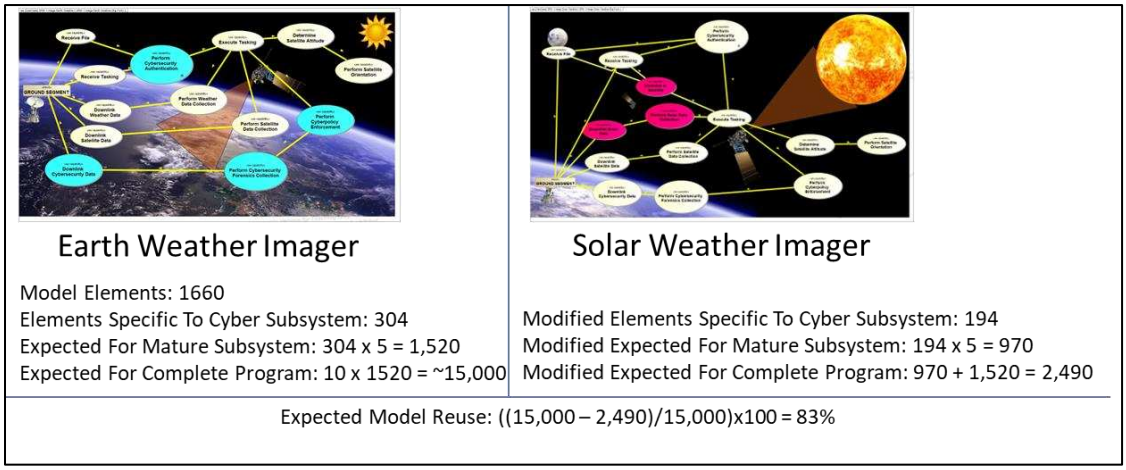

Figure 47: Model Reuse

If a company wants to create different satellite product lines, then they would be expected to have a lot less ability to reuse their models. However, based on our approach to MBSE modeling, the SS Capabilities are still transferable from program to program. For example, the behavior for how satellite communicates to the ground will be similar from program to program, even if the specific components and interfaces could be different.

This concludes the Model Reuse Demonstration, whereby we were able to leverage and modify an existing architecture on a similar program. All of the model elements and requirements were already linked, allowing us to rapidly create a mission-specific architecture. The only changes required were those unique to this new mission (i.e. payload, communication subsystem, and impacted FSW functions). As our example company continues to evolve their satellite product line to meet new customer needs and payloads, they will have an SSOT MBSE

model that they can continually model, recycle, and improve as they design and build their satellites.

5.9 Research Question 3 Conclusion

Upon completion of this research, we have answered Research Question 3: Evaluate how an MBSE cyber-secure satellite architecting process preserves the benefits of utilizing MBSE.

In this research question, we have expanded our rudimentary baseline cyber-secure satellite architecture from Research Questions 1 and 2 into a more robust cyber-secure MBSE architecture model which could be leveraged by a company as they embark on a Class C/D satellite program. We demonstrated our architectural design process for a cyber-secure satellite program, and discussed the reusability potential of the MBSE model as well as the additional benefits of implementing MBSE beyond the cybersecurity benefits which industry could realize, such as an overall reduction in costs. This research provides evidence to support the assertion that MBSE is a toolkit that has real value at both an organizational and an individual level, and will benefit companies as they design cyber-secure modern satellites.

Chapter 6.

Conclusions

This research has defined a series of tasks to address the primary research challenges associated with defining the cybersecurity vulnerabilities in commercial LEO Space Critical Infrastructure. This research conducted a scholarly review of literature on cybersecurity threats and mitigation techniques in space and satellite systems to determine the vulnerabilities of a satellite system. The results of this research include an improved and generalizable understanding of cybersecurity vulnerabilities in LEO Space Critical Infrastructure, and a determination in how and to what degree MBSE can aid in realizing and securing these cybersecurity vulnerabilities. Furthermore, we demonstrated our MBSE architectural design process for a cyber-secure satellite program, and explained the benefits and reusability of such an architecture, proving its desirability over the traditional documents-based systems engineering process.

6.1 Research Contributions

The research presented within this dissertation provides the following contributions to the fields of satellite cyber-secure architectures:

- A scholarly review of the literature on cybersecurity threats and mitigation techniques in space and satellite systems, to identify the cybersecurity threats, common architectural components, and common internal and external interfaces for satellite systems.

- A mapping/evaluation of a set of cybersecurity mitigation architectures to the requirements of the space and satellite systems application. This was used to derive a novel and scalable set of security requirements for the LEO application.
- A model-based systems engineering example case for a cyber-security enabled satellite system, demonstrating the utility of our proposed cyber-secure satellite architecting process.
- An evaluation of the costs and benefits of a SysML MBSE-enabled architecting process as applied to an industrial satellite system architecting process, quantifying metrics to collaboration and reuse.

When taken together, these research contributions describe an overall enhancement to the process of incorporating cybersecurity into the systems engineering satellite design process through the implementation of an integrated cyber-secure architecting process and MBSE to help integrate cybersecurity into satellite design and MBSE on small-scale projects in the space industry, while demonstrating the efficiency and reusability of an MBSE model.

These research contributions represent novel contributions to the state of defining the cybersecurity vulnerabilities for Space Systems, and show how MBSE can aid in the continued evolution of the development and operation of Space Systems while also benefitting the companies responsible for completing these satellite projects.

6.2 Future Work

We have determined three areas of future work that could be pursued beyond the scope of this dissertation: 1) Applying the methodology to non-product line satellite programs, 2)

Measuring the ROI by tracking actual program hours, and 3) Measuring success in industry with multiple experienced MBSE practitioners.

The cyber-secure architecture process demonstrated in this dissertation was specific to a company that has a common satellite product line. Through the common product line approach, companies are able to maximize their investment into a MBSE approach, as stated in Section 5. However, if a company is not utilizing a common product line approach, we don't expect the ROI to be as significant. For future work researchers could apply the approach we have defined in this dissertation to multiple different programs which are not based on a common product line. This would allow the researchers to quantify the reusability of the MBSE model versus what needs to change between non-common product line programs within a given company. We do still believe a significant amount of benefits can be seen even by non-product line programs if the MBSE governance as defined in this dissertation is implemented. Even if programs are substantially different from each other, if the company can leverage this methodology, they won't need to spend time and money on the creation of their own methods of modeling. Additionally, there are various model elements (i.e. logical models, model structure methodology, type of capabilities) that are still transferable from program to program even among non-common product line programs. We expect that programs utilizing different architectures could still leverage company-defined MBSE models, with an increase in effort on each individual program to either modify existing model elements or add elements to address the uniqueness of each program.

Measurement of the actual ROI of this MBSE approach could be accomplished by measuring the actual labor hours to perform the engineering effort to design, build, and test a satellite first using traditional Systems Engineering methods and second while implementing

MBSE. However, this measurement would be difficult to scientifically perform, because no two satellite programs are exactly identical so it is difficult to strictly define a "control" effort under the traditional systems approach, without the experience being carried over into the MBSE trial. Attempting to measure the variable of systems engineering design methodology without changing any other confounding variables (i.e. the involved engineers and their experience levels, program differences or variety, the company accomplishing the work to be measured, etc.) is nearly impossible; however, an attempt to measure it could be made by finding a company that has utilized the traditional SE methods and already measured their labor hours for SE artifacts to design, built, and test a satellite. Then have that same company with the same team members follow the MBSE approach defined in this dissertation and measure the labor hours to do the same level of work on a comparable program. Lastly, we would need to measure MBSE hours across multiple programs of similar size which the original MBSE model is utilized as a baseline. The challenge to this effort would include several factors: the programs are likely to be multiyear programs which while comparable would likely not be exactly identical in perceived effort; it is highly unlikely that each program will have the exact same execution team which could introduce different amounts of personnel experience leading to efficiency challenges (either new team members are less efficient, or old team members who were part of the baseline "control" program have gained that design experience leading to increased efficiency on subsequent programs they participate in such as the new MBSE program); and the source of funding for this research would need to be determined. This cost & schedule complexity is potentially one of the driving reasons this level of research has not yet been performed.

When adopting the MBSE approach defined in this dissertation, the overall MBSE approach may need to be tailored to the specific companies' skill set. Ideally companies will adopt the process as defined in this dissertation, since this approach was defined for people new to MBSE modeling. However, there are many skilled MBSE engineers within the space design community. They may choose to follow their own approaches which they are familiar with or have even developed themselves. Without having MBSE modeling governance that ensures consistency from program to program, as laid out in this dissertation, it becomes increasingly challenging to compare program to program to determine an ROI. However, if these skilled MBSE engineers want to follow their own approach to modeling, they would be able to help define a ROI from a consistent approach as long as they are consistent in their MBSE processes from program to program.

References

[1] G. Satell, "The Industrial Era Ended, and So Will the Digital Era," Harvard Business Review, 18 July 2018. [Online]. Available: https://hbr.org/2018/07/the-industrial-era-ended-and-so-will-the-digital-era.

[2] A. Craig and B. Valeriano, "Reacting to Cyber Threats: Protection and Security in the Digital Age," *Global Security and Intelligence Studies,* vol. 1, no. 2, pp. 21 - 41, 2016.

[3] D. Craigen, N. Diakun-Thibault and R. Purse, "Defining Cybersecurity," *Technology Innovation Management Review,* vol. 4, no. 10, pp. 13 - 21, 2014.

[4] E. Mossburg, J. Gelinne and H. Calzada, "Beneath the surface of a cyberattack: A deeper look at business impacts," Deloitte, 14 June 2016. [Online]. Available: https://www2.deloitte.com/content/dam/Deloitte/us/Documents/risk/us-risk-beneath-the-surface-of-a-cyber-attack.pdf.

[5] K. Chadd, "The history of cybersecurity," Avast, 24 November 2020. [Online]. Available: https://blog.avast.com/history-of-cybersecurity-avast.

[6] C. Bai, P. Dallasega, G. Orzes and J. Sarkis, "Industry 4.0 technologies assessment: A sustainability perspective," *International Journal of Production Economics,* vol. 229, no. 107776, 2020.

[7] Touro University, "The 10 Biggest Ransomware Attacks of 2021," 12 November 2021. [Online]. Available: https://illinois.touro.edu/news/the-10-biggest-ransomware-attacks-of-2021.php.

[8] K. Collier, "Barely able to keep up': America's cyberwarriors are spread thin by attacks," NBC News, 8 July 2021. [Online]. Available: https://www.nbcnews.com/tech/security/ransomware-attacks-leave-cybersecurity-experts-barely-able-keep-rcna1337.

[9] M. Smith and J. Monken, "The Colonial Pipeline Hack Shows We Need a Better Federal Cybersecurity Ecosystem," Modern Ware Institute, 01 June 2021. [Online]. Available: https://mwi.westpoint.edu/the-colonial-pipeline-hack-shows-we-need-a-better-federal-cybersecurity-ecosystem/.

[10] F. Batista, M. Hirtzer and M. Dorning, "All of JBS's U.S. Beef Plants Were Forced Shut by Cyberattack," Bloomberg, 31 May 2021. [Online]. Available: https://www.bloomberg.com/news/articles/2021-05-31/meat-is-latest-cyber-victim-as-hackers-hit-top-supplier-jbs.

[11] J. Marks, "Space could be the next frontier for cyber threats," The Washington Post, 8 November 2021. [Online]. Available: https://www.washingtonpost.com/politics/2021/11/08/space-could-be-next-frontier-cyber%20threats/.

[12] M. King and S. Goguichvili, "Cybersecurity Threats in Space: A Roadmap for Future Policy," Wilson Center, 8 October 2020. [Online]. Available: https://www.wilsoncenter.org/blog-post/cybersecurity-threats-space-roadmap-future-policy.

[13] L. Shadbolt, "Technical Study Satellite Cyberattacks and Security," HDI Global Specialty SE, July 2021. [Online]. Available: https://www.hdi.global/globalassets/_local/international/downloads/specialty/hdis209_satellite-cyberattack_whitepaper_v8_05july21-1.pdf.

[14] B. Weeden and V. Samson, "Global Counterspace Capabilities: An Open Source Assessment," Secure World Foundation, 2018.

[15] B. Bailey, R. J. Speelman, P. A. Doshi, N. C. Cohen and W. A. Wheeler, "Defending Spacecraft in the Cyber Domain," The Aerospace Corporation Center for Space Policy and Strategy, 2019.

[16] G. Falco, "The Vacuum of Space Cyber Security," in *AIAA Space and Astronautics Forum*, Orlando, 2018.

[17] M. Manulis, C. P. Bridges, R. Harrison, V. Sekar and A. Davis, "Cyber Security in New Space," *International Journal of Information Security,* vol. 20, no. June 2021, pp. 287-311, 2020.

[18] G. Falco, "Job One for Space Force: Space Asset Cybersecurity," Harvard Kennedy School Belfer Center for Science and International Affairs, Cambridge, MA, 2018.

[19] B. Nussbaum and G. Berg, "Cybersecurity Implications of Commercial off the Shelf (COTS) Equipment in Space Infrastructure," in *Space Infrastructures: From Risk to Resilience Governance*, Amsterdam, IOS Press, 2020, pp. 91-99.

[20] C. Poole, R. Bettinger and M. Keith, "Shifting Satellite Control Paradigms Operational Cybersecurity in the Age of Megaconstellations," *AIR & SPACE POWER JOURNAL,* 2021.

[21] G. Falco, "Our satellites are a prime target for a cyberattack. And things could get worse.," *Washington Post,* 7 May 2019.

[22] D. Winder, "In space, no one can hear cyber security professionals scream," *The Register,* 2 September 2021.

[23] A. Kerzhner, K. Tan and E. Fosse, "Analyzing cyber security threats on cyber-physical systems using Model-Based Systems Engineering.," *AIAA SPACE 2015 Conference and Exposition,* p. 4575, 2015.

[24] INCOSE, "System Engineering Vision 2035," 2021. [Online]. Available: https://www.incose.org/publications/se-vision-2035.

[25] J. M. Borky and T. H. Bradley, Effective model-based systems engineering, Springer, 2018.

[26] United States Government Accountability Office, "Weapon Systems Cybersecurity: Guidance would help DOD Programs Better Communicate Requirements to Contractors," US Government Accountability Office, Washington DC, 2021.

[27] The White House, "Space Policy Directive-5—Cybersecurity Principles for Space Systems," 4 September 2020. [Online]. Available: https://trumpwhitehouse.archives.gov/presidential-actions/memorandum-space-policy-directive-5-cybersecurity-principles-space-systems/.

[28] G. Murray, M. Johnstone and C. Valli, "The convergence of IT and OT in critical infrastructure," *Australian Information Security Management Conference,* vol. 15, p. 149, 2017.

[29] Canadian Centre For Cyber Security, "An Introduction To The Cyber Threat Environment," 2020. [Online]. Available: https://cyber.gc.ca/sites/default/files/publications/Intro-ncta-2020_e.pdf.

[30] SentinelOne, "Threat Actor Basics: The 5 Main Threat Types," 9 September 2020. [Online]. Available: https://www.sentinelone.com/blog/threat-actor-basics-understanding-5-main-threat-types/.

[31] Wilson Center, "Cybersecurity on the Final Frontier: Protecting Our Critical Space Assets from Cyber Threats," 14 July 2021. [Online]. Available: https://www.wilsoncenter.org/event/cybersecurity-final-frontier-protecting-our-critical-space-assets-cyber-threats.

[32] D. Bodeau, J. Fabius-Greene and R. Graubart, "How Do You Assess Your Organization's Cyber Threat Level?," The MITRE Corporation, August 2010. [Online]. Available: https://www.mitre.org/sites/default/files/pdf/10_2914.pdf.

[33] J. Pavur and I. Martinovic, "SOK: Building a Launchpad for Impactful Satellite Cyber-Security Research," *arXiv:2010.10872,* 2010.

[34] S. Visner and S. Kordella, "Cyber Best Practices for Small Satellites," *ASCEND 2020,* p. 4013, 2020.

[35] Otorio, "IT vs OT cyber security: The Operational Technology Guide," 01 September 2020. [Online]. Available: https://www.otorio.com/blog/it-security-vs-ot-security-the-operational-technology-cybersecurity-guide-for-industry-professionals/.

[36] NASA, "NASA's Cybersecurity Readiness," 18 May 2021. [Online]. Available: https://oig.nasa.gov/docs/IG-21-019.pd.

[37] Consultative Committee for Space Data Systems, "Security Threats Against Space Missions," NASA CCSDS, Washington DC, 2022.

[38] Committee on National Security Systems, "Security Categorization and Control Selection for National Security Systems," CNSSI, 2014.

[39] M. Scholl and T. Suloway, "NISTIR 8270, Intro to Cybersecurity for Commercial Satellite Operations," NIST, July 2023. [Online]. Available: https://csrc.nist.gov/pubs/ir/8270/final.

[40] D. Livingstone and P. Lewis, "Space, the Final Frontier for Cybersecurity?," Chatham House. The Royal Institute of International Affairs., 2016.

[41] H. Kramer, "CASSIOPE (Cascade SmallSat and Ionospheric Polar Explorer)," eoPortal, 28 May 2012. [Online]. Available: https://www.eoportal.org/satellite-missions/cassiope.

[42] H. Kramer, "GIOVE-B (Galileo In-Orbit Validation Element-B)," eoPortal, 29 May 2012. [Online]. Available: https://www.eoportal.org/satellite-missions/giove-b.

[43] H. Kramer, "LADEE (Lunar Atmosphere and Dust Environment Explorer)," eoPortal, 31 May 2012. [Online]. Available: https://www.eoportal.org/satellite-missions/ladee#spacecraft.

[44] H. Kramer, "MIOSat (Missione Ottica Su MicroSatellite)," eoPortal, 13 June 2012. [Online]. Available: https://www.eoportal.org/satellite-missions/miosat.

[45] H. Kramer, "QBITO CubeSat," eoPortal, 12 June 2012. [Online]. Available: https://www.eoportal.org/satellite-missions/qbito.

[46] H. Kramer, "Sich-2 (Optical Observation Mission-2)," eoPortal, 14 June 2012. [Online]. Available: https://www.eoportal.org/satellite-missions/sich-2.

[47] Data Device Corporation, "SCS750 - Single Board Computers for Space," 2017. [Online]. Available: https://www.artisantg.com/info/DDC_SCS750_Datasheet_202042115039.pdf.

[48] Southwest Research Institute, "Single Board Computers," N.D.. [Online]. Available: https://www.swri.org/industry/space-engineering/single-board-computers.

[49] CAES, "DS4350272-X00," N.D.. [Online]. Available: https://caes.com/sites/default/files/documents/Datasheet-DS4350272-X00.pdf.

[50] Aitech, "SP0-S | Radiation Tolerant 3U CompactPCI SBC," N.D.. [Online]. Available: https://aitechsystems.com/product/sp0-s-rad-tolerant-3u-compactpci-sbc/.

[51] BAE Systems, "RAD5545 SpaceVPX single-board computer," 2018. [Online]. Available: https://satsearch.s3.eu-central-1.amazonaws.com/datasheets/satsearch_datasheet_iadixe_bae_systems_rad5545.pdf.

[52] Innoflight, "CHAMPS Flight Computer (MPSoC CFC-400)," 26 January 2019. [Online]. Available: https://satcatalog.s3.amazonaws.com/components/434/SatCatalog_-_Innoflight_-_CFC-400_-_Datasheet.pdf.

[53] Seakr, "Medusa SBC," 2020. [Online]. Available: https://satcatalog.s3.amazonaws.com/components/446/SatCatalog_-_SEAKR_Engineering_-_Medusa_SBC_-_Datasheet.pdf.

[54] Ibeos, "B28-135," N.D.. [Online]. Available: https://satsearch.co/products/ibeos-b28-135-satellite-battery.

[55] SAFT, "VES16 8s4p battery," N.D.. [Online]. Available: https://www.saftbatteries.com/products-solutions/products/saft-solution-leo-and-small-geo-applications.

[56] SparkWing, "SmallSat Solar Array," N.D.. [Online]. Available: https://sparkwing.space/satellite-solar-panels.

[57] EnduroSat, "6U Deployable Solar Array," N.D.. [Online]. Available: https://endurosat.com/cubesat-store/cubesat-solar-panels/6u-deployable-solar-array/.

[58] Blue Canyon, "3U Solar Array," N.D.. [Online]. Available: https://bluecanyontech.com/components.

[59] L3Harris Technologies, "MSX-765 Transceiver," N.D.. [Online]. Available: https://www.satcatalog.com/component/msx-765-transceiver/.

[60] IQ Spacecom, "SLink-PHY Transceiver," N.D.. [Online]. Available: https://www.iq-spacecom.com/products/slink-phy.

[61] Tethers Unlimited, "SWIFT-SLX Transceiver," N.D.. [Online]. Available: https://www.tethers.com/wp-content/uploads/2021/03/SWIFT-SLX-2.pdf.

[62] Honeywell, "STC-MS03," N.D.. [Online]. Available: https://aerospace.honeywell.com/content/dam/aerobt/en/documents/learn/products/sensors/brochures/N61-1603-000-000_S-BandTT_and_C_Transceiver_br.pdf.

[63] Innoflight, "SCR-106," N.D.. [Online]. Available: https://www.innoflight.com/product-overview/scrs/scr-106/.

[64] Blue Canyon, "SDR," N.D.. [Online]. Available: https://storage.googleapis.com/blue-canyon-tech-news/1/2021/03/BCT_DataSheet_Components_SoftwareDefinedRadios.pdf.

[65] Radiall, "Low Power Coaxial DP3T Switch," N.D.. [Online]. Available: https://www.radiall.com/products/space-qualified-components/space-coaxial-switches.html.

[66] Teledyne, "33SDC," N.D.. [Online]. Available: https://www.teledynedefenseelectronics.com/relays/Datasheets/HCR-33.indd1.pdf.

[67] Renaissance Electronics Corporation, "SW-316," N.D.. [Online]. Available: https://www.rec-usa.com/wp-content/uploads/2021/03/SW-316-datasheet.pdf.

[68] Innoflight, "KI-103," N.D.. [Online]. Available: https://www.innoflight.com/product-overview/ecus/ki-103/.

[69] L3Harris Technologies, "MCU-110C," N.D.. [Online]. Available: https://www.l3harris.com/all-capabilities/secure-satellite-communications-encryption-unit-mcu-110c.

[70] EnduroSat, "S-Band Antenna Commercial," N.D.. [Online]. Available: https://www.endurosat.com/cubesat-store/cubesat-antennas/s-band-antenna-commercial/.

[71] SpaceQuest Ltd., "AC-2000," N.D.. [Online]. Available: https://www.spacequest.com/components/4.

[72] L3Harris Technologies, "AS-48917," N.D.. [Online]. Available: https://www.l3harris.com/all-capabilities/48917-gps-band-omnidirectional-antenna.

[73] Innovative Solutions In Space B.V. (ISIS), "S-band Patch Antenna," N.D.. [Online]. Available: https://www.isispace.nl/product/s-band-patch-antenna/.

[74] ANYWAVES, "S-band Antenna," N.D.. [Online]. Available: https://anywaves.eu/products/s-band-ttc-antenna/.

[75] Blue Canyon, "Full Extension NST," N.D.. [Online]. Available: https://storage.googleapis.com/blue-canyon-tech-news/1/2021/03/BCT_DataSheet_Components_StarTrackers.pdf.

[76] Terma, "T1 (45 Deg Baffle)," N.D.. [Online]. Available: https://www.terma.com/media/pokirm23/t1-star-tracker-aug-2021.pdf.

[77] Space Micro, "μSTAR-200M," N.D.. [Online]. Available: https://www.satcatalog.com/component/star-200m/.

[78] Sodern, "Hydra-TC," N.D.. [Online]. Available: https://www.sodern.com/website/docs_wsw/RUB_215/tile_508/Datasheet_HYDRA_TC_2017.pdf.

[79] L3Harris Technologies, "CIRUS-EX," N.D.. [Online]. Available: https://www.l3harris.com/all-capabilities/compact-inertial-reference-units-space-cirus-cirus-ex.

[80] Sensonor AS, "STIM202," N.D.. [Online]. Available: https://www.sensonor.com/media/1074/datasheet-stim202-ts1439-r6.pdf.

[81] Northrop Grumman, "LN-200S," N.D.. [Online]. Available: https://www.northropgrumman.com/wp-content/uploads/LN-200S-Inertial-Measurement-Unit-IMU-datasheet.pdf.

[82] Honeywell, "MIMU," N.D.. [Online]. Available: https://aerospace.honeywell.com/us/en/learn/products/space/miniature-inertial-measurement-unit-mimu.

[83] Blue Canyon, "RWP100," N.D.. [Online]. Available: https://storage.googleapis.com/blue-canyon-tech-news/1/2020/06/BCT_DataSheet_Components_ReactionWheels_06_2020.pdf.

[84] L3Harris Technologies, "RWA-15," N.D.. [Online]. Available: https://www.satcatalog.com/component/rwa-15/.

[85] Microsat Systems Canada, "MicroWheel 1000," N.D.. [Online]. Available: http://www.mscinc.ca/products/mw-1000.html.

[86] Hyperion Technologies, "RW210-6.0," N.D.. [Online]. Available: https://hyperiontechnologies.nl/wp-content/uploads/2018/07/HT-RW210_V1.02_Flyer.pdf.

[87] Millennium Space Systems, "RWA-1000," N.D.. [Online]. Available: https://www.millennium-space.com/brochures/RWA1000Brochure.pdf.

[88] Bradford Space, "W45," N.D.. [Online]. Available: https://static1.squarespace.com/static/603ed12be884730013401d7a/t/6054f630baf06f76bbabe02a/1616180789682/be_datasheet_rwu_2019dec.pdf.

[89] ZARM Technik AG, "AMR Magnetometer," N.D.. [Online]. Available: https://www.satcatalog.com/component/amr-magnetometer/.

[90] Magson GmbH, "MACM Fluxgate," N.D.. [Online]. Available: https://www.satcatalog.com/component/macm-fluxgate/.

[91] NewSpace Systems, "NMRM-Bn25o485," N.D.. [Online]. Available: https://www.newspacesystems.com/wp-content/uploads/2020/10/NewSpace-Magnetometer_2020_10a.pdf.

[92] ZARM Technik AG, "MT70-2," N.D.. [Online]. Available: https://www.satcatalog.com/component/mt70-2/.

[93] Sinclair Interplanetary, "TQ-15," N.D.. [Online]. Available: https://www.satcatalog.com/component/tq-15/.

[94] NewSpace Systems, "NCTR-M012," N.D.. [Online]. Available: https://www.newspacesystems.com/wp-content/uploads/2020/10/NewSpace-Magnetorquer-Rod_2020-10a-1.pdf.

[95] Chang Guang Satellite Technology, "Magnetic Torquer," N.D.. [Online]. Available: https://satsearch.co/products/cgsatellite-magnetic-torquer.

[96] Adcole Maryland Aerospace, "Coarse Sun Sensor," N.D.. [Online]. Available: https://adcolespace.com/product/coarse-sun-sensor/.

[97] Solar MEMS Technologies, "SSOC-A60," N.D.. [Online]. Available: https://www.cubesatshop.com/wp-content/uploads/2016/06/SSOCA60-Brochure-1.pdf.

[98] Hyperion Technologies, "SS200," N.D.. [Online]. Available: https://hyperiontechnologies.nl/products/ss200-sun-sensor/.

[99] Bradford Space, "Fine Sun Sensor," N.D.. [Online]. Available: https://static1.squarespace.com/static/603ed12be884730013401d7a/t/6054f58b93435f7449136e83/1616180620554/be_datasheet_fss_2017jan.pdf.

[100] RUAG, "LEORIX GNSS Receiver," N.D.. [Online]. Available: https://www.ruag.com/system/files/media_document/2020-04/LEORIX%20V1.0.pdf.

[101] Surrey Satellite Technology, "SGR-Axio," N.D.. [Online]. Available: https://www.sstl.co.uk/getmedia/28f193a8-95b2-41ab-9688-a1974bffcbb8/SGR-Axio-Datasheet-2018-V2.pdf.

[102] MOOG, "NavSBR," N.D.. [Online]. Available: https://www.moog.com/content/dam/moog/literature/Space_Defense/spaceliterature/avionics/moog-navigation-single-board-receiver-navsbr-datasheet.pdf.

[103] General Dynamics, "Viceroy-4 GPS Spaceborne Receiver," N.D.. [Online]. Available: https://gdmissionsystems.com/products/communications/spaceborne-communications/spaceborne-gps-receivers/viceroy-gps-receiver.

[104] Airbus, "LION 1100Neo GNSS Receiver," N.D.. [Online]. Available: https://www.satcatalog.com/component/lion-1100neo-gnss-receiver/.

[105] MINCO, "4009/003," N.D.. [Online]. Available: https://escies.org/download/webDocumentFile?id=60862.

[106] Tayco Engineering, "Flexible Heater," N.D.. [Online]. Available: https://www.taycoeng.com/proa.htm.

[107] All Flex, "Thermofoil Heaters," N.D.. [Online]. Available: https://allflexheaters.com/.

[108] Tayco Engineering, "Surface Temperature Sensor," N.D.. [Online]. Available: https://www.taycoeng.com/prod.htm.

[109] Variohm, "ESA/ESCC Space Qualified NTC Thermistors with Leads," N.D.. [Online]. Available: https://www.variohm.com/products/temperature-sensors/space-qualified-hi-rel-thermistors/esa-escc-space-qualified-ntc-thermistors-with-leads.

[110] Renesas, "ISL71590SEH," N.D.. [Online]. Available: https://www.renesas.com/us/en/products/space-harsh-environment/rad-hard-hermetic/rad-hard-analog/rad-hard-temperature-sensors/isl71590seh-radiation-hardened-2-terminal-temperature-transducer.

[111] MOOG, "Thrust Single Seat," N.D.. [Online]. Available: https://www.moog.com/content/dam/moog/literature/Space_Defense/spaceliterature/propulsion/moog-bipropellant-thruster-valves-datasheet.pdf.

[112] Bradford Space , "1N HPGP," N.D.. [Online]. Available: https://www.ecaps.space/products-1n.php.

[113] Aerojet, "MR-103G," N.D.. [Online]. Available: https://www.rocket.com/sites/default/files/documents/In-Space%20Data%20Sheets%204.8.20.pdf.

[114] ArianeGroup , "Low Pressure Latch Valve," N.D.. [Online]. Available: https://www.space-propulsion.com/spacecraft-propulsion/valves/latch-valve.html.

[115] VACCO, "V1E10537-01," N.D.. [Online]. Available: https://www.vacco.com/index.php/space/latch-valves1.

[116] Stellar Technology, "ST1300," N.D.. [Online]. Available: https://www.stellartech.com/products/techsheets-press/st1300.pdf.

[117] Taber Industries, "2211," N.D.. [Online]. Available: https://www.tabertransducer.com/content/documents/Taber%202211%20Data%20Sheet%20%2810-2021%29.pdf.

[118] ArianeGroup, "177 L Hydrazine Tank - OST 31-1," N.D.. [Online]. Available: https://www.space-propulsion.com/brochures/propellant-tanks/104-177lt-n2h4-tank-ost-31-0.pdf.

[119] MT Aerospace AG , "E3000-590," N.D.. [Online]. Available: https://www.satcatalog.com/component/e3000-590/.

[120] MOOG, "Type 2 Solar Array Drive Assembly," N.D.. [Online]. Available: https://www.moog.com/content/dam/moog/literature/Space_Defense/spaceliterature/spacecraft_mechanisms/moog-type-2-solar-array-drive-assembly-datasheet.pdf.

[121] SNC, "C14 Bi-Axis Gimbal," N.D.. [Online]. Available: https://www.sncorp.com/media/3189/space-systems-product-catalog-2020.pdf.

[122] RUAG, "SEPTA® 24," N.D.. [Online]. Available: https://www.ruag.com/system/files/2016-12/Structures-Brochure_1.pdf.

[123] SNC, "Hold Down Release Mechanism (HDRM)400 W Articulated Array," N.D.. [Online]. Available: https://www.sncorp.com/media/3189/space-systems-product-catalog-2020.pdf.

[124] RUAG, "Hold Down and Release Mechanism," N.D.. [Online]. Available: https://www.ruag.com/en/products-services/space/spacecraft/satellite-mechanisms/deployment-and-separation.

[125] T. J. K. R. T. &. Y. M. Harrison, "Space Threat Assessment 2020," 30 March 2020. [Online]. Available: https://www.csis.org/analysis/space-threat-assessment-2020.

[126] D. Snyder, J. D. Powers, E. Bodine-Baron and B. Fox, "Improving the Cybersecurity of U.S. Air Force Military Systems Throughout Their Life Cycles," RAND Corporation, 27 October 2015. [Online]. Available: https://www.rand.org/content/dam/rand/pubs/research_reports/RR1000/RR1007/RAND_RR1007.pdf.

[127] S. F. Lokman, A. T. Othman and M. H. Abu-Bakar, "Intrusion detection system for automotive Controller Area Network (CAN) bus system: a review," *EURASIP Journal on Wireless Communications and Networking 2019,* vol. 1, pp. 1-17, 2019.

[128] C. Plummer, P. Roos and L. Stagnaro, "CAN Bus as a Spacecraft Onboard Bus," *Proceedings of DASIA 2003 (ESA SP-532),* 2003.

[129] L. Pan, X. Zheng, H. Chen, T. Luan, H. Bootwala and L. Batten, "Cyber security attacks to modern vehicular systems," *Journal of Information Security and Applications,* vol. 36, pp. 90-100, 2017.

[130] S. Abbott-McCune and L. A. Shay, "Intrusion prevention system of automotive network CAN bus," *IEEE International Carnahan Conference on Security Technology (ICCST),* vol. 2016, pp. 1-8, 2016.

[131] M. J. Varghese, A. Anwar, F. Jiang and R. Doss, "Novel CAN Bus Fuzzing Framework for Finding Vulnerabilities in Automotive Systems," *2024 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S),* vol. 2024, pp. 56-58, 2024.

[132] M. Lombardi, P. F. and D. Santaniello, "Two-Step Algorithm to Detect Cyber-Attack Over the Can-Bus: A Preliminary Case Study in Connected Vehicles," *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg,* vol. 8, 2021.

[133] G. Falco, "Cybersecurity Principles for Space Systems," *Journal of Aerospace Information Systems,* vol. 16, no. 2, pp. 61-70, 2019.

[134] C. Vasquez and E. Groll, "Satellite Hack on Eve of Ukraine War was a Coordinated, Multi-pronged Assault," *Cyberscoop,* 10 August 2023.

[135] M. Guenot, "Hackers Figured our 3 Separate Ways to Break into US Air Force Satellites, and won up to $50K for doing it," *Business Insider,* 17 August 2023.

[136] T. McKay, "Hackers Compete to Break into the Space Force's Moonlighter Satellite," *IT Brew,* 12 September 2023.

[137] J. Willbold, M. Schloegel, M. Vogele, M. Gerhardt, T. Holz and A. Abbasi, "Space Odyssey: An Experimental Software Security Analysis of Satellites," in *44th IEEE Symposium on Security and Privacy*, San Francisco, 2023.

[138] F. Calvelli, "Memorandum: Space Acquisition Tenets," Department of the Air Force Office of the Assistant Secretary, Washington DC, 2022.

[139] L. Franceschi-Bicchierai, *This $1000 Device Let Hackers Hijack Satellite Communications,* https://www.vice.com/en/article/xywjpa/this-1000-device-lets-hackers-

hijack-satellite-communications, 2015, pp. https://www.vice.com/en/article/xywjpa/this-1000-device-lets-hackers-hijack-satellite-communications.

[140] United States Government Accountability Office, " Weapon Systems Cybersecurity: DOD Just Beginning to Grapple with Scale of Vulnerabilities," US Government Accountability Office, Washington DC, 2018.

[141] G. Johnson-Roth and D. Pinkley, "TOR-2011(8591)-21: Mission Assurance Guidelines for A-D Mission Risk Class," The Aerospace Corporation, 2011.

[142] B. Braun and L. Jasper, "How Satellites are Moving Beyond the Class System: Class Agnostic Development and Operations Approaches for Constraints-Driven Missions," in *Small Sat Conference*, Virtual, 2021.

[143] National Institute of Standards and Technology, "Security and Privacy Controls for Information Systems and Organizations, SP 800-53 Rev. 5," NIST, 2020.

[144] National Institute of Standards and Technology, "Standards for Security Categorization of Federal Information and Information Systems," NIST, 2004.

[145] Office of the DoD Chief Information Officer , "DOD Instruction 8510.01 - Risk Management Framework for DOD Systems," US Department of Defense, 2022.

[146] S. Mills and T. Denman, "The Cybersecurity and Acquisition Life-Cycle Integration Tool," Defense Acquisition University, 2017.

[147] Office of Inspector General, "NASA's Cybersecurity Readiness, Report No. IG-21-019," National Aeronautics and Space Administration, 2021.

[148] NASA Office of the Inspector General, "Cybersecurity Management and Oversight at the Jet Propulsion Laboratory," National Aeronautics and Space Administration, Washington, DC, 2019.

[149] National Aeronautics and Space Administration, "NPR 2810.1F NASA Information Security Policy," NASA Policy Directive, 2022.

[150] National Aeronautics and Space Administration, "NASA-STD-1006 Space System Protection Standard," 15 07 2022. [Online]. Available: https://standards.nasa.gov/standard/NASA/NASA-STD-1006.

[151] National Aeronautics and Space Administration, "NPR 7150.2D NASA Software Engineering Requirements," 8 March 2022. [Online]. Available: https://nodis3.gsfc.nasa.gov/displayDir.cfm%3Ft%3DNPR%26c%3D7150%26s%3D2D.

[152] National Aeronautics and Space Administration, "NASA-HDBK-1005 NASA Space Mission Architecture Framework (SMAF) Handbook For Uncrewed Space Missions," 11 03 2021. [Online]. Available: https://standards.nasa.gov/standard/NASA/NASA-HDBK-1005.

[153] National Aeronautics and Space Administration, "NASA-STD-8739.8B Software Assurance and Software Safety Standard," 08 09 2022. [Online]. Available: https://standards.nasa.gov/standard/NASA/NASA-STD-87398.

[154] SAM.GOV, "80GSFC22R0009 -Geostationary Extended Observations Spacecraft Solicitation," 26 1 2023. [Online]. Available: https://sam.gov/opp/591373d792254d10832076340c56baf8/view.

[155] National Aeronautics and Space Administration, "Geostationary Operational Environmental Satellites - R Series," [Online]. Available: https://www.goes-r.gov/.

[156] National Security Agency Cybersecurity Directorate, "UEFI Secure Boot Customization," NSA, 2023.

[157] National Security Agency, "Announcing the Commercial National Security Algorithm Suite 2.0," NSA Cybersecurity Advisory, 2022.

[158] Consultative Committee for Space Data Systems, "Space Data Link Security Protocol," NASA CCSDS, Washington DC, 2022.

[159] Consultative Committee for Space Data Systems, "CCSDS Cryptographic Algorithms," NASA CCSDS, Washington DC, 2019.

[160] KSAT, "Our history," [Online]. Available: https://www.ksat.no/about-us/.

[161] INCOSE, Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities Version 4.0, Hoboken, NJ: John Wiley and Sons, Inc, 2015.

[162] Department of Defense Office of Prepubliscation and Security Review, Department of Defense Cybersecurity Test and Evaluation Guidebook, Department of Defense, 2020.

[163] Defense Acquisition University, "Cybersecurity Test and Evaluation Guidebook, Version 2, Change 1," Department of Defense, 6 Februray 2020. [Online]. Available:

https://www.dau.edu/cop/test/documents/cybersecurity-test-and-evaluation-guidebook-version-2-change-1.

[164] Defense Information Systems Agency (DISA) and National Security Agency (NSA) Zero Trust Engineering Team, "Zero Trust Reference Architecture Version 2.0," Department of Defense, July 2022. [Online]. Available: https://dodcio.defense.gov/Portals/0/Documents/Library/(U)ZT_RA_v2.0(U)_Sep22.pdf.

[165] K. Scarfone and P. Mell, "NIST SP 800-94 Guide to Intrusion Detection and Prevention Systems (IDPS)," National Institute of Standards and Technology, February 2007. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/94/final.

[166] E. Kraft and D. Chesebrough, "Aerospace, Defense Industry Must Join Digital Revolution," *National Defense,* vol. 102, no. 775, pp. 16-17, 2018.

[167] E. R. Carroll and R. J. Malins, "SAND2016-2607 Systematic literature review: How is model-based systems engineering justified?," SANDIA REPORT, 2016.

[168] U.S. Government Accountability Office, "Our Core Values," [Online]. Available: https://www.gao.gov/about/what-gao-does/our-core-values.

[169] United States Government Accountability Office, "GAO-19-336SP Weapon Systems Annual Assessment: Limited Use of Knowledge-Based Practices Continues to Undercut DOD's Investments," 07 May 2019. [Online]. Available: https://www.gao.gov/products/gao-19-336sp.

[170] J. B. Holladay, J. Knizhnik, K. J. Weiland, A. Stein, Schwindt and T. S. a. P., "MBSE Infusion and Modernization Initiative (MIAMI): “Hot” Benefits for Real NASA Applications," *2019 IEEE Aerospace Conference,* pp. 1-14, 2019.

[171] G. F. Dubos, D. P. Coren, A. Kerzhner, S. H. Chung and J.-F. Castet, "Modeling of the flight system design in the early formulation of the Europa Project," in *IEEE Aerospace Conference*, 2016.

[172] K. Henderson, "MBSE adoption experiences in organizations: Lessons learned," *Systems Engineering,* vol. 27, no. 1, p. 214–239, 2024.

[173] D. R. Call and D. R. Herber, "Applicability of the diffusion of innovation theory to accelerate model-based systems engineering adoption," *INCOSE Systems Engineering,* vol. 25, no. 6, pp. 574-583, 2022.

[174] S. Friedenthal, A. Moore and R. Steiner, A Practical Guide to SysML: The Systems Modeling Language, Morgan Kaufmann, 2014.

[175] B. P. Douglass, Agile Model-Based Systems Engineering Cookbook - Second Edition, Packt Publishing, 2022.

[176] C. Madariaga, A. Bashir and C. Swickline, "Applying MBSE in Space Based Systems Development," *33rd Annual INCOSE International Symposium 15–20 July 2023 — Honolulu, HI,* vol. 33, no. 1, pp. 584-600, 2023.

[177] J. R. Knizhnik, K. J. Weiland, T. A. Grondin, K. M. Jones-McDowall and J. B. Holladay., "Realized Benefits from the Model-Based Systems Engineering Infusion and Modernization Initiative," *63rd Space Sciences and Technology Conference,* 2019.

[178] M. L. Rozek, K. M. Donahue, M. D. Ingham and J. D. Kaderka, "A Tool for Model-Based Generation of Scenario-Driven Electric Power Load Profiles," *AIAA SPACE 2015 Conference and Exposition,* p. 4463, 2015.

[179] R. Ferguson, J. Marshall and L. Assadzadeh., "Automated Power Analysis of Onboard Spacecraft Electronics with Model Based Systems Engineering," *2019 IEEE Aerospace Conference,* pp. 1-8, 2019.

[180] P. M. Fischer, D. Lüdtke, C. Lange, F.-C. Roshani, F. Dannemann and A. Gerndt, "Implementing model-based system engineering for the whole lifecycle of a spacecraft," *CEAS Space journal,* vol. 9, no. 3, pp. 351-365, 2017.

[181] M. Hause, "How to Fail at MBSE," 2013. [Online]. Available: https://www.incose.org/docs/default-source/texas-gulf-coast/m_hause_how_to_fail_at_mbse.pdf.

[182] J. Maurandy, E. Gill, A. Helm and R. Stalford, "Cost-Benefit Analysis of SysML Modelling for the Atomic Clock Ensemble in Space (ACES) Simulator," *INCOSE Rome, Italy, July 9–12, 2012,* vol. 22, no. 1, pp. 1726-1745, 2014.

[183] C. F. Claver, G. Dubois-Felsmann, F. Delgadol, P. Hascall, S. Marshall, M. Nordby, T. Schalk, G. Schumacher and J. Sebag, "Using SysML for MBSE analysis of the LSST system," *Modeling, Systems Engineering, and Project Management for Astronomy IV,* vol. 7738, pp. 518-527, 2010.

[184] A. I. Anyanhun, A. Anzagira and W. W. .. Edmonson, "Intersatellite communication: An MBSE operational concept for a multiorbit disaggregated system," *IEEE Journal on Miniaturization for Air and Space Systems,* vol. 1, no. 1, pp. 56-65, 2020.

[185] J. Gregory, L. Berthoud, T. Tryfonas and A. Prezzavento, "Early validation of the data handling unit of a spacecraft using MBSE.," *2019 IEEE Aerospace Conference,* pp. 1-15, 2019.

[186] M. Russell, "Using MBSE to enhance system design decision making," *Procedia Computer Science,* vol. 8, pp. 188-193, 2012.

[187] S. Subarna, A. K. Jawale, A. S. Vidap, S. D. Sadachar, S. Fliginger and S. Myla, "Using a model based systems engineering approach for aerospace system requirements management," *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC),* pp. 1-8, 2020.

[188] W. Vaneman, R. Carlson and C. Wolfgeher, "NPS-SE-20-016 Defining a model-based systems engineering approach for milestone technical reviews," Acquisition Research Program, 2019.

[189] A. M. Madni, C. C. Madni and S. D. Lucero, "Leveraging digital twin technology in model-based systems engineering.," *MDPI Systems,* vol. 7, no. 1, 2019.

[190] J. Walker and J. M. Borky, "Test Planning, Documentation, and Impact Analysis with SysML," *ITEA Journal of Test & Evaluation,* vol. 41, no. 4, pp. 258-266, 2020.

[191] N. Shevchenko, "Evaluating Threat-Modeling Methods for Cyber-Physical Systems," SEI Blog, 4 February 2019. [Online]. Available: https://insights.sei.cmu.edu/blog/evaluating-threat-modeling-methods-for-cyber-physical-systems/.

[192] Microsoft, "Threats - Microsoft Threat Modeling Tool - Azure.," 25 August 2022. [Online]. Available: https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats.

[193] VerSprite, "Process for attack simulation and threat analysis - VerSprite," 23 May 2024. [Online]. Available: https://versprite.com/cybersecurity-listings/offsec/pasta-threat-modeling/.

[194] B. Schneier, "Attack Trees," *Dr. Dobb's Journal,* vol. 24, no. 12, pp. 21-29, 1999.

[195] MITRE, "MITRE ATT&CK," n.d.. [Online]. Available: https://attack.mitre.org/.

[196] Aerospace Corporation, "SPARTA | The Aerospace Corporation," n.d.. [Online]. Available: https://aerospace.org/sparta.

[197] Z. Li, D. Qin and F. Yang, "Exploration of a Capability-Focused Aerospace System of Systems Architecture Alternative with Bilayer Design Space, Based on RST-SOM Algorithmic Methods," *Scientific World Journal,* 2014.

[198] M. Chami and J. M. Bruel, "A Survey on MBSE Adoption Challenges," in *NCOSE EMEA Sector Systems Engineering Conference (INCOSE EMEASEC 2018)*, Berlin, Germany, 2018.

[199] J. A. Estefan, "Survey of model-based systems engineering (MBSE) methodologies.," *Incose MBSE Focus Group,* vol. 25, no. 8, pp. 1-12, 2007.

[200] N. Shevchenko, "Modeling Capabilities with Model-Based Systems Engineering (MBSE)," SEI Blog, 28 November 2022. [Online]. Available: https://insights.sei.cmu.edu/blog/modeling-capabilities-with-model-based-systems-engineering-mbse/.

[201] J. Hummell, "Functional Architecture using SySML," MBSE Solutions, 15 June 2018. [Online]. Available: https://mbse.solutions/functional-architecture-using-sysml/.

[202] National Aeronautics and Space Administration, "Mission Phase Definitions," [Online]. Available: https://science.nasa.gov/earth-science/programs/flight-programs/mission-phase-definitions/.

[203] C. S. Wasson, "System Analysis, Design, and Development: Concepts, Principles, and Practices," John Wiley & Sons, Inc, 2005, pp. 189 - 205.

[204] C. S. Wasson, "System Phases, Modes, and States Solutions to Controversial Issues," in *INCOSE International Symposium*, Denver, 2014.

[205] T. Weilkiens, "Adoption of MBSE in an Organization," *Handbook of Model-Based Systems Engineering,* pp. 1-19, 2022.

[206] T. Bayer, "Is MBSE helping? Measuring value on Europa Clipper," *2018 IEEE Aerospace Conference ,* pp. 1 - 13, 2018.

[207] S. Sanders, "Does a Model Based Systems Engineering Approach Provide Real Program Savings? Lesson Learnt," OMG Sysml, 25 Oct 2011. [Online]. Available: htttp://www.omgsysml.org/Does_a_MBSE_Approach_Provide_Savings Lessons_Learnt-Saunders 200111.pdf.

[208] H. Gans, "Use Of Model Based Systems Engineering (MBSE) To Improve Program Metrics," Harris, 2018. [Online]. Available: https://ndiastorage.blob.core.usgovcloudapi.net/ndia/2018/systems/Thurs_21539_Gans_C.pdf.

[209] J. Krasner, "How Product Development Organizations can Achieve Long-Term Cost Savings Using Model-Based Systems Engineering (MBSE)," EmbeddedMarket Forecasters, October 2015. [Online]. Available: https://www.ptc.com/~/media/2BAF3B088FB44C8591F67F46BB8B6C93.ashx.

[210] S. Y. Kim, D. Wagner and A. Jimenez, "Challenges in applying model-based systems engineering: human-centered design perspective," in *INCOSE HSI2019*, Biarritz, Franc, 2019.

[211] K. Campo, T. Teper, C. Eaton, A. Shipman, G. Bhatia and B. Mesmer, "Model-based systems engineering: evaluating perceived value, metrics, and evidence through literature," *Systems Engineering,* vol. 26, no. 1, pp. 104-129, 2023.

[212] P. Whitehead, T. Light, A. Luna and J. Mignano, "A Framework for Assessing the Costs and Benefits of Digital Engineering.," 13 March 2024. [Online]. Available: https://www.rand.org/pubs/research_reports/RRA2418-1.html.

[213] T. McDermott, N. Hutchison, M. V. A. E. Clifford, A. Salado and K. Henderson, "Benchmarking the Benefits and Current Maturity of Digital Engineering/Model-Based SE," System Engneering Research Center (SERC), Stevens Institute of Technology, Hoboken, NJ, 2020.

[214] J. B. Duffy, J. Feng, R. Combs and J. P. Richardson, "Return on Investment in Model-Based Systems Engineering Software Tools," *INCOSE International Symposium,* vol. 31, no. 1, pp. 791-805, 2021.

[215] P. Simpkins, "Use of Requirements in Model-Based Systems Engineering for a Legacy Design," KIHOMAC, 2012. [Online]. Available: https://ndiastorage.blob.core.usgovcloudapi.net/ndia/2012/system/track514888.pdf.

[216] B. Cavell and K. Lam, "Model-Based Systems Engineering Cost Study. In 2023 NASA Cost and Schedule Symposium (OTR 2023-00350)," The Aerospace Corporation, 2023. [Online]. Available: https://www.nasa.gov/wp-content/uploads/2023/06/05-mbse-cost-study-otr-2023-00350.pdf.

[217] E. Villhauer, "Why Model-Based Systems Engineering Reduces Costs," Nomagic, 04 January 2016. [Online]. Available: https://blog.nomagic.com/why-model-based-systems-engineering-reduces-costs/.

[218] D. Cook and W. Schindel, "Utilizing MBSE Patterns To Accelerate System Verificaiton," *Insight (International Council on Systems Engineering),* vol. 20, no. 1, pp. 32-41, 2017.

[219] L. Wheatcraft, "Applicability of MBSE to Different Types of Projects," RequirementsExperts, 28 February 2014. [Online]. Available: https://reqexperts.com/2014/02/28/applicability-of-mbse-to-different-types-of-projects/.

[220] A. Gibney, Director, *Zero Days.* [Film]. United States: Participant Media; Showtime Documentary Films; Global Produce; Jigsaw Productions, 2016.

[221] N. Perlroth, "In Cyberattack on Saudi Firm, US Sees Iran Firing Back," The New York Times, 23 October 2012. [Online]. Available: https://www.nytimes.com/2012/10/24/business/global/cyberattack-on-saudi-oil-firm-disquiets-us.html.

[222] N. Perlroth and Q. Hardy, "Bank Hacking Was the Work of Iranians, Officials Say," The New York Times, 8 January 2013. [Online]. Available: https://www.nytimes.com/2013/01/09/technology/online-banking-attacks-were-work-of-iran-us-officials-say.html.

Appendix A

Case Study – Stuxnet and the Natanz Uranium Refining Centrifuges [220]

Perhaps one of the best examples of terrestrial cyberattack encompassing all of the aspects which are applicable to cybersecurity in space occurred at the Natanz Nuclear Power Plant in Iran in 2009. The US National Security Agency (NSA) in conjunction with Israel designed a piece of malware now known as the Stuxnet virus which was aimed at Supervisory Control and Data Acquisition (SCADA) devices called Programmable Logic Controllers (PLCs) which were manufactured by Siemens. These Siemens PLCs were known to control the centrifuges responsible for uranium enrichment at the Natanz plant. Unbeknownst to the operators, this computer worm infiltrated the PLCs and changed their logic, causing them to increase their rate of rotation and tear themselves apart, thus setting back Iranian production of nuclear-grade uranium and Iranian nuclear weapons development. What marks this example as especially prescient to the security of satellite systems are the following characteristics:

- The Iranian plant was in a remote and difficult-to-access location. As a nuclear facility, it was also heavily locked down with highly restricted access and patrolled perimeters. These characteristics are all reasonably analogous to the conditions of space-based satellites and the generalized defense tactics taken by satellite owners (i.e., the satellite is remote and therefore “impossible” to access).
- The PLCs in use at Natanz were not connected to the internet, nor were they located on a system that was accessible remotely via intermittent internet-to-network connection. The

Stuxnet worm was instead introduced to non-connected networks via USB device, with the hope that eventually it would proliferate and spread via infected computer networks to a different USB device that would end up plugged into a computer at Natanz which was connected to the PLCs. Once again, this situation is analogous to current satellite defense which does not predict a non-connected satellite to be vulnerable except through its ground station communications.

- The targeted Siemens PLCs were a COTS product which Iran chose to use to save time and costs in developing their own PLCs. This made it easy for the NSA to study the PLCs at their own facilities, discover the zero-day vulnerabilities which their final virus product exploited, and even practice to perfect their worm and see the results first-hand. This COTS product incorporation strategy is already being used in the satellite design to bring down costs – but Stuxnet illustrates that the risks of using COTs products must be defined and mitigated ahead of an attack.
- Finally, up until the irreversible damage to the centrifuges occurred, operators at Natanz had no way of knowing that their system had been targeted long ago and was under ongoing attack for months while the worm slowly worked its way through the Middle East, hopping from computer network to network on USB drives until it finally reached its target. Even after the damage occurred, the attack wasn't discovered until the worm was found by someone else on an unrelated computer system (the result of being released "into the wild" on a USB stick). The operators at Natanz instead assumed operator error and replaced many sets of centrifuges (and fired many operators) for more than a year. Similarly, a satellite operator is unlikely to know that the satellite is under a similar

cyberattack until after the attack has already succeeded, and even then the satellite operator might assume that something else went wrong instead of a cyberattack.

The Stuxnet case proves that any operational endpoint is reachable and attackable and that irreversible damage or total destruction can occur before the target is even aware that an attack has been launched. If a remote and disconnected nuclear facility can be infiltrated in a cyberattack, then it is not a question of "if" but rather "when" a cyberattack on a satellite will be successful, even despite the hostile environment of space and the exposure to extreme temperature variations, radiation, and vacuum conditions which an attacker must overcome to reach the satellite.

In retaliation to Stuxnet, the Iranians set up their own Iranian Cyber Army and launched their own cyberattacks two years later, attacking the world's most valuable (at the time) oil company, Saudi Aramco, and wiping out every line of code on 30,000 computer devices, replacing the data with the picture of a burning American flag [221]. An additional months-long attack on American banking systems has also been attributed to Iranian State attackers and resulted in sporadic Distributed Denial of Service (DDoS) attacks which collapsed large banking sites such as Bank of America and Wells Fargo by hijacking underprotected cloud computing and data centers and using them to flood the banking websites with data until they became overwhelmed and shut down, denying actual users the ability to log in and manage their money [222]. While officially the US, Israeli, and Iranian governments deny any involvement in all cyberattacks, the message of such sophisticated cyberattacks remains: "If you can do it to us, we can do it to you."

Appendix B

MBSE Model

The MBSE model that was used to generate the findings for this dissertation is located on GitHub at the following location:

https://github.com/CSUDENGSP2025/CSUDENG-SP2025

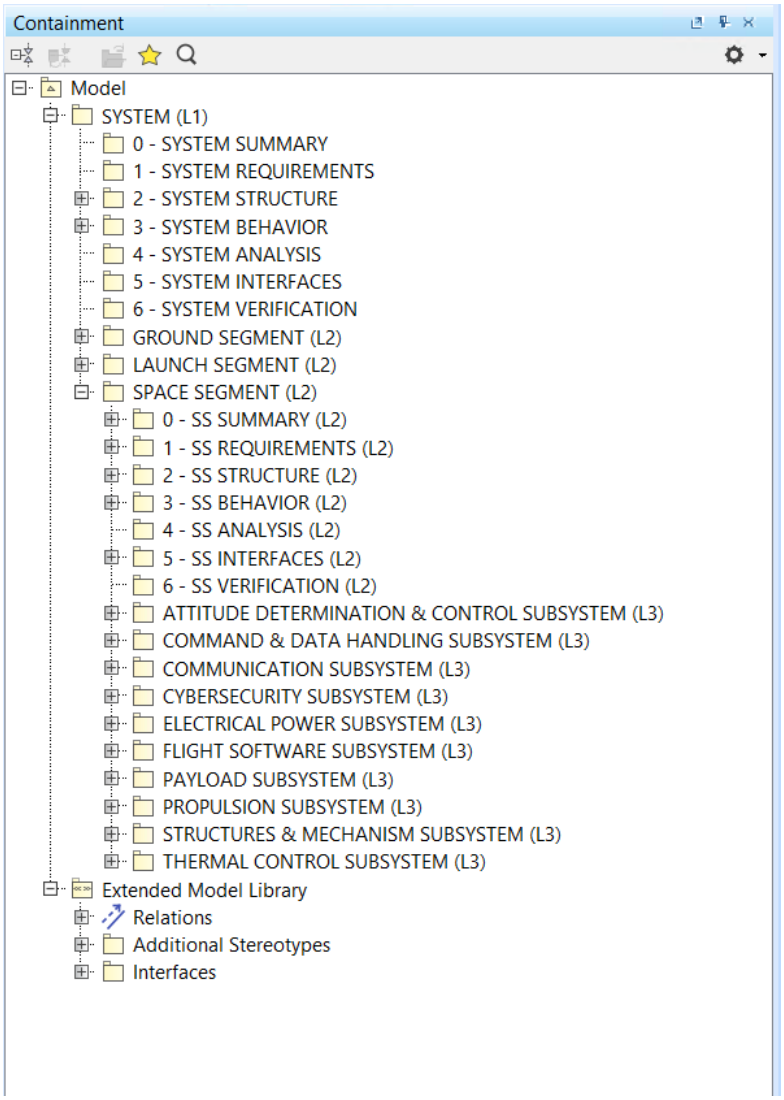

Figure 48: MBSE Model Containment Tree