This is the peer reviewed author accepted manuscript (post print) version of a published work that appeared in final form in:

Falco, Gregory, Boschetti, Nicolo, Viswanathan, Arun, Bailey, Brandon, Slay, Jill, Bennett, Shane, Yahia, Olfa Ben 2024 'Minimum requirements for space system cybersecurity-ensuring cyber access to space', Proceedings - 2024 IEEE 10th International Conference on Space Mission Challenges for Information Technology, SMC-IT 2024, pp. 78-88

This un-copyedited output may not exactly replicate the final published authoritative version for which the publisher owns copyright. It is not the copy of record. This output may be used for non-commercial purposes.

The final definitive published version (version of record) is available at:
https://doi.org/10.1109/SMC-IT61443.2024.00016

Persistent link to the UniSA Research Outputs Repository:
https://researchoutputs.unisa.edu.au/11541.2/41635

General Rights

Copyright and moral rights for the publications made accessible in the UniSA Research Outputs Repository are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

- Users may download and print one copy for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the persistent link identifying the publication in the UniSA Research Outputs Repository

If you believe that this document breaches copyright, please contact us and provide details. We will remove access to the work immediately and investigate your claim.

Minimum Requirements for Space System Cybersecurity - Ensuring Cyber Access to Space

Gregory Falco
Cornell University
Sibley School of Mech. & Aerospace Engineering
Ithaca, USA
gfalco@cornell.edu

Nicolo` Boschetti
Cornell University
Ithaca, USA
nb624@cornell.edu

Arun Viswanathan
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, USA
arun.a.viswanathan@jpl.nasa.gov

Brandon Bailey
Aerospace Corporation
West Virginia, USA
brandon.bailey@aero.org

Carsten Maple
University of Warwick
Coventry, UK
CM@warwick.ac.uk

Gunes Karabulut Kurt
Polytechnique Montre´al
Montre´al, Canada
gunes.kurt@polymtl.ca

Johannes Willbold
Ruhr University Bochum
Bochum, Germany
johannes.willbold@rub.de

Jill Slay
University of South Australia
Adelaide, Australia
Jill.Slay@unisa.edu.au

Edward Birrane
JHU Applied Physics Laboratory
Laurel, USA
Edward.Birrane@jhuapl.edu

David Logsdon
Information Technology Industry Council
Washington D.C., USA
dlogsdon@itic.org

Shane Bennett
University of South Australia
Adelaide, Australia
Shane.Bennett@missioncyber.com.au

William Ferguson
ethicallyHackingSpace
Luxembourg
william.o.ferguson@ethicallyhacking.space

James Curbo
JHU Applied Physics Laboratory
Laurel, USA
James.Curbo@jhuapl.edu

Johan Sigholm
Swedish Defence University
Stockholm, Sweden
johan.sigholm@fhs.se

Cameron Mehlman
Cornell University
Ithaca, USA
cpm222@cornell.edu

Rajiv Thummala
Cornell University
Ithaca, NY
rkt34@cornell.edu

Matteo Calabrese
Cornell University
Ithaca, USA
mc2884@cornell.edu

Anh Tuan Le
University of Warwick
Coventry, UK
a.le.1@warwick.ac.uk

Kymie Tan
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, USA
kymie.tan@jpl.nasa.gov

Erin Miller
Space ISAC
Colorado Springs, USA
erin@spaceisac.org

Gregory Epiphaniou
University of Warwick
Coventry, UK
Gregory.Epiphaniou@warwick.ac.uk

Ugur Ilker Atmaca
University of Warwick
Coventry, UK
Ugur-Ilker.Atmaca@warwick.ac.uk

Wayne C. Henry
Air Force Institute of Technology
Wright-Patterson AFB, USA
wayne.henry@us.af.mil

Gürkan Gü r
Zurich University of Applied Sciences
Winterthur, Switzerland
gurkan.gur@zhaw.ch

Olfa Ben Yahia
Polytechnique Montre´al
Montre´al, Canada
olfa.ben-yahia@polymtl.ca

***Abstract*—Space systems are continuously under cyber attack. Minimum cybersecurity design requirements are necessary to preserve our access to space. This paper proposes a scalable, extensible method for developing minimum cyber design principles and subsequent requirements for a space system based on any given mission priority. To test our methodology, we selected the fundamental mission priority of preserving access to space by preventing the permanent loss of control of a satellite. We then**

generate the minimum number of secure-by-design principles that can result in the permanent loss of control of a satellite and translate these into example minimum requirement 'shall' statements. Our proposed minimum requirements methodology and example can serve as a starting point for policymakers aiming to establish security requirements for the sector. Further, our methodology for establishing minimum requirements will be engaged for prioritizing the efforts of the emergent IEEE International Technical Standard for Space Cybersecurity (Working Group P3349).

***Index Terms*—Space Cybersecurity, Cybersecurity Requirements, Secure-by-Design, Space Security Standards**

I. Introduction

The indispensable nature of space access today extends far beyond the confines of traditional defense and civil science operations, permeating into the commercial markets of global communication, navigation, and financial services. The ubiquity of societal reliance on space underscores the criticality of securing this domain against adversarial threats.

The dynamic threat environment, characterized by the development and potential deployment of anti-satellite capabilities, underscores the urgency for robust security frameworks. The cyberattack on Viasat, occurring in the context of the Russia-Ukraine conflict, serves as a case study in the multifaceted nature of threats to space systems [1]. Such events catalyze a reevaluation of existing space cybersecurity paradigms and society's collective need to ensure cyber access to space.

Despite the fundamental role of space systems, there remains a notable absence of security requirements specifically aimed at safeguarding space access against such threats. This deficiency can partly be attributed to the lack of technical standards for the secure design and operation of space systems. In response, the IEEE International Technical Standard for Space System Cybersecurity, through Working Group P3349, is developing secure-by-design technical specifications for space systems [2]. This is directly in line with the United States of America's Office of the National Cyber Director (ONCD) technical report issued in February 2024 entitled *Back to the Building Blocks: A Path Toward Secure and Measurable Software* that describes the need for bottom-up security requirements for software, with an emphasis on space systems [3].

The ultimate degree of implementation of the IEEE Standard for Space System Cybersecurity will inherently result in variability in the security posture of individual missions. This raises a fundamental question: What are the minimum cybersecurity requirements to ensure space system security?

The question of minimum requirements is not merely technical but also strategic, reflecting a balance between security and the imperative for innovation and mission flexibility.

This paper aims to address this question by clarifying the concept of minimum requirements, outlines a methodology for determining secure-by-design principles that will inform these requirements, illustrates an example of the methodology applied to a specific mission objective, transforms the illustrated minimum required design principles into 'shall' statements, and concludes by discussing the implications of establishing minimum requirements. Authored by leaders of the IEEE International Technical Standard for Space System Cybersecurity, this paper seeks to inform the ongoing work of the P3349 Working Group and assist policymakers in establishing a baseline for future regulations and policies.

II. Prior Art

A. Existing Guidance for Space Systems

Despite the existence of best practices, guidelines, and protocols, there remains a notable gap in a granular, bottom-up approach to end-to-end standards and hard requirements tailored to the secure development and operation of space systems. Existing standards largely address top-down risks, which does not account for bottom-up security design specifications for newly developed systems. Ultimately, the space cybersecurity community needs both. This section aims to provide a succinct overview of the existing guidance in the domain.

1) National Aeronautics and Space Administration (NASA): NASA has developed comprehensive standards and guidelines to secure its missions and payloads. The Space System Protection Standard (NASA-STD-1006) ensures resilience against threats, focusing on command authority and data integrity [4]. The Science Mission Directorate's Rideshare Users Guide (2021 SMD SPA RUG with DNH) aids in spacecraft design and launch integration, ensuring compatibility and non-interference with other missions [5]. The Space Security Best Practices Guide (BPG) translates NIST security controls into NASA's context for widespread applicability and executive-level cybersecurity management [6]. Additionally, the Security of Information Systems directive (NPR 2810.1F) defines the information security program's requirements, aligning with federal standards and NIST publications for comprehensive information protection throughout its lifecycle [7].

2) European Cooperation for Space Standardization (ECSS): The ECSS has introduced the Software Product Assurance (ECSS-Q-ST-80C) standard, emphasizing the importance of assuring the quality and reliability of space software products. This includes both commercial and modified off-the-shelf software. It advocates for a risk-based approach and adherence to systems engineering practices to enhance software reliability and performance in space missions. Further, ECSS has been working towards establishing security engineering guidance that complements its risk-based approach.

3) National Institute of Standards and Technology (NIST): NIST offers comprehensive frameworks and guidance to bolster the cybersecurity and systems security engineering of satellite networks and operations. NIST SP 800-160 outlines a framework for integrating systems security engineering within the systems engineering process, catering to various stages and complexities of systems' lifecycles [8]. NIST IR 8441 provides practical guidance for stakeholders in satellite network design, acquisition, and operation, focusing on aligning cybersecurity practices with organizational risk tolerance and enhancing the protection, detection, response, and recovery of

hybrid satellite network services [9]. Additionally, NIST IR 8401 assists organizations in applying the NIST Cybersecurity Framework to satellite ground segments, emphasizing command, control, and payload systems without dictating specific requirements [10]. Meanwhile, NIST IR 8270 introduces basic concepts of cybersecurity risk management for the commercial satellite industry, aiming to foster discussions and provide references for further understanding of effective cybersecurity risk management practices in space [11].

4) Consultative Committee for Space Data Systems (CCSDS): CCSDS has developed a series of documents focusing on the application of security measures to its protocols, enhancing the security of data transmission in space missions. CCSDS 350.0-G details the application of security measures, specifically through the Space Data Link Security (SDLS) Protocol, to ensure authentication, confidentiality, and integrity across space data link protocols such as TM, TC, and Advanced Orbiting Systems (AOS) [12]. CCSDS 355.0-B introduces the Space Data Link Security Protocol, offering a structured approach to apply data authentication and confidentiality at the Data Link Layer [13]. CCSDS 356.0-B extends the application of the SDLS Protocol to missions using the Internet Protocol (IP), providing guidance for network layer security [14]. Lastly, CCSDS 357.0-B expands on authentication mechanisms by defining "credential profiles" based on X.509 certificates and password authentication, aimed at missions requiring robust end-to-end data security measures [15]. These documents collectively contribute to the secure and reliable transmission of data in space exploration and satellite communication.

5) Japanese Ministry of Economy, Trade and Industry (METI): METI's Cybersecurity Guidelines for Commercial Space Systems encourage voluntary cybersecurity measures in the commercial space sector by outlining security risks and basic security measures, alongside references for further guidance [16].

6) German Federal Office for Information Security (Bundesamt für Sicherheit in der Informationstechnik or BSI): BSI's IT-Grundschutz Profile for Space Infrastructures offers a template for space-related security concepts, emphasizing company-specific IT-Grundschutz profiles based on minimum security requirements, adaptable to unique mission needs [17].

B. Emerging IEEE International Technical Standard for Space System Cybersecurity

The aforementioned documents serve to advocate for and facilitate the implementation of cybersecurity practices; however, they often adopt a piecemeal, top-down, risk-based approach and may lack the technical granularity that would inform requirements and aid in the secure design of a space system. In response to these limitations, the IEEE Standard for Space Cybersecurity introduces a forward-thinking framework. This standard proposes a security-centric, component-based specification designed to underpin the development of future space missions from their inception. It emphasizes the incorporation of security principles that are robust against a broad spectrum of threats, including those posed by existing, hypothetical, and future adversaries. By prioritizing a foundational, approach to security, this standard seeks to ensure that space missions are designed with inherent resilience against cyber threats, aligning with the evolving space cybersecurity landscape. The IEEE Standard for Space Systems Cybersecurity does not seek to replace other standards; instead, where possible, we aim to point to and reference existing work by bodies such as BSI, EECS, NIST, and CCSDS when their standard's granularity suits the required bottom-up secure-by-design approach.

Our *secure-by-component* approach moves away from categorizing missions into classes or recommending a uniform reference architecture or a set of reference architectures [18]. Instead, we adopt a system-of-systems perspective when considering secure-by-design for space missions, given that each segment of a space system can be decomposed into subsystems and further broken into components and subcomponents. Secure-by-component centers on addressing the complexity and diversity of space systems by prioritizing the security of their fundamental building blocks. These foundational components, referred to as *secure blocks*, can be flexibly combined to create secure architectures tailored to the specific requirements of each unique space mission.

The standard focuses on applying this strategy to several common low-level components found within the user, space, ground, integration layer, and link segments of space missions. The ultimate output of this technical standard will be the creation of an extensive catalog of 'shall' statement requirements for each subcomponent that can be assembled to formulate secure blocks for space missions. It is unlikely that the complete set or catalog of 'shall' statement requirements for a particular subcomponent will be used together when assembling a secure block. Doing so would be redundant given multiple secure-by-design approaches and their associated requirements in the catalog address the same weaknesses. However, in the context of a mission, engaging certain secure-by-design approaches could be better suited than others given mission objectives and constraints. Therefore, realistically, a systems engineer would combine relevant shall statements to compose secure blocks to suit the mission. A systems engineer would build one of two versions of secure blocks that can be assembled with the subcomponent requirements catalog: (a) a *maximum-security design*, crafted to mitigate all the identified threats, with respect to the high-level objectives of a mission; (b) a *minimum-security design*, aimed at achieving a narrowly defined security objective. By providing these two distinct levels of security designs, system architects can choose a design that best fits the unique objectives of their space mission, ensuring an appropriate level of security for mission success.

However, a key challenge we face is rigorously defining how to establish which requirements compose minimum and maximum secure blocks. This paper focuses on detailing a rigorous methodology for establishing the minimum security requirements for a space system.

III. Methodology

A. Scope Definition and Initial Approach

Our methodology development process began with a critical scoping exercise aimed at defining the essence of "minimum." We first sought to develop attack trees for each facet of the space ecosystem (space, ground, link, and user segments and the integration layer) to outline potential attack vectors. This process, while insightful, revealed a significant limitation due to the subjective nature of attack tree permutations. These permutations spanned a broad spectrum of security concerns, including confidentiality, integrity, and availability challenges, without yielding a clear path to actionable insights. The realization of the subjective and arbitrary nature of attack trees prompted a strategic pivot in our analytical approach.

In response to the limitations identified, our methodology evolved to incorporate fault tree analysis. This analytical shift allowed for a more objective and focused examination of the system, particularly through the lens of a mission security objective. For our example, we chose to focus on satellite availability as the security priority. The selection of permanent loss of satellite control, defined as an indefinite loss of availability, as the primary failure mode to be addressed, anchored our analysis.

B. Component Analysis, CWE Identification and Secure by Design Principle Mapping

With the mission security priority established, the methodology advanced to a detailed component analysis within each system segment. This phase involved identifying components at risk of becoming single points of failure, potentially leading to the catastrophic loss of satellite control. Following this identification, our efforts focused on mapping common weakness enumerations (CWEs) to attack techniques that the IEEE P3349 Working Group identified were relevant to each vulnerable component. CWEs represent standardized types of software vulnerabilities targeted by attack techniques.

Having mapped out the weaknesses, our methodology advanced to the enumeration of prevention measures for each initiating event (the CWEs). This was achieved through the adoption of secure-by-design principles as outlined in NIST SP 800-160 v2r1. These principles are designed to fortify the software against identified CWEs, thereby mitigating the risk of component failure across the system's segments.

C. Matrix Row Reduction for Solving Minimum Requirements

The final phase of our methodology concentrated on distilling the enumerated secure-by-design approaches into minimum requirements. This was accomplished by assessing the capacity of these approaches to address multiple CWEs concurrently. Through a process of matrix row reduction, we aimed to identify the most efficient combination of secure-by-design approaches that could comprehensively mitigate the risk of cyber threats leading to a permanent loss of satellite control. This streamlined approach not only ensures that the mission priority is achieved but also delineates the most concise set of approaches necessary to safeguard against critical failures. The approaches are ultimately translated to 'shall' statements relevant to the single-point-of-failure components identified. Such requirements will enable the establishment of secure blocks for our secure-by-component strategy.

D. Approach Scalability for Diverse Mission Objectives

The *secure-by-component* approach discussed in Section II-B takes a mission profile agnostic view of security, enabling developers to assemble secure blocks (containing either minimum or maximum requirements) for a secure mission. Our proposed minimum requirements methodology enables a developer to infuse requirements specific to mission objectives and priorities into each secure block. While our illustration includes the rudimentary objective of preserving availability, the methodology is adaptable to accommodate alternative mission-specific security objectives. For example, a national security mission may have the objective of confidential communications. As such, rather than anchoring the fault tree with the failure of 'permanent loss of control of the satellite', the root could be 'breach of confidentiality'. By selecting a failure mode relevant to a mission objective and following the outlined sequence of operations, the minimum cyber requirements for each secure block become inextricably linked to the overall mission objectives. Thus, secure blocks with only minimum requirements focus on the security necessary for a particular mission objective.

We can apply a similar method to establish the *maximum cyber requirements* for a secure block. This entails analyzing all the critical failure modes corresponding to a mission's objectives to determine the most extensive set of requirements. In essence, the maximum cyber requirements for a secure block are a union of the requirements derived from an analysis of all the mission objectives. It is important to note that these maximum requirements are tailored to each mission and may not be universally applicable across all missions.

IV. Fault Tree

We utilized a fault tree to represent the relationship between the subsystems of each segment, the CWEs (the initiating events) and the secure-by-design approaches (prevention measures for the initiating events). Fault trees help to effectively identify the relationship between failure events and system elements for complex systems, rendering them as *gates.* In this study, we identified relationships described by OR and AND gates. "OR" gates have been used to depict scenarios when any single fault or initiating event of a given set can lead to a further failure in the system. "AND" gates have been used to identify scenarios in which all the conditions of a given set must happen to lead to further failure. Each secure-by-design Approach strengthens one or more CWEs, and this relationship is described in a bottom-up association.

A. Segments, Subsystems, and Functions

As shown in Figure 1, five different segments (space, ground, link, user, and the integration layer) are identified as elements of a space mission that can lead to a permanent loss

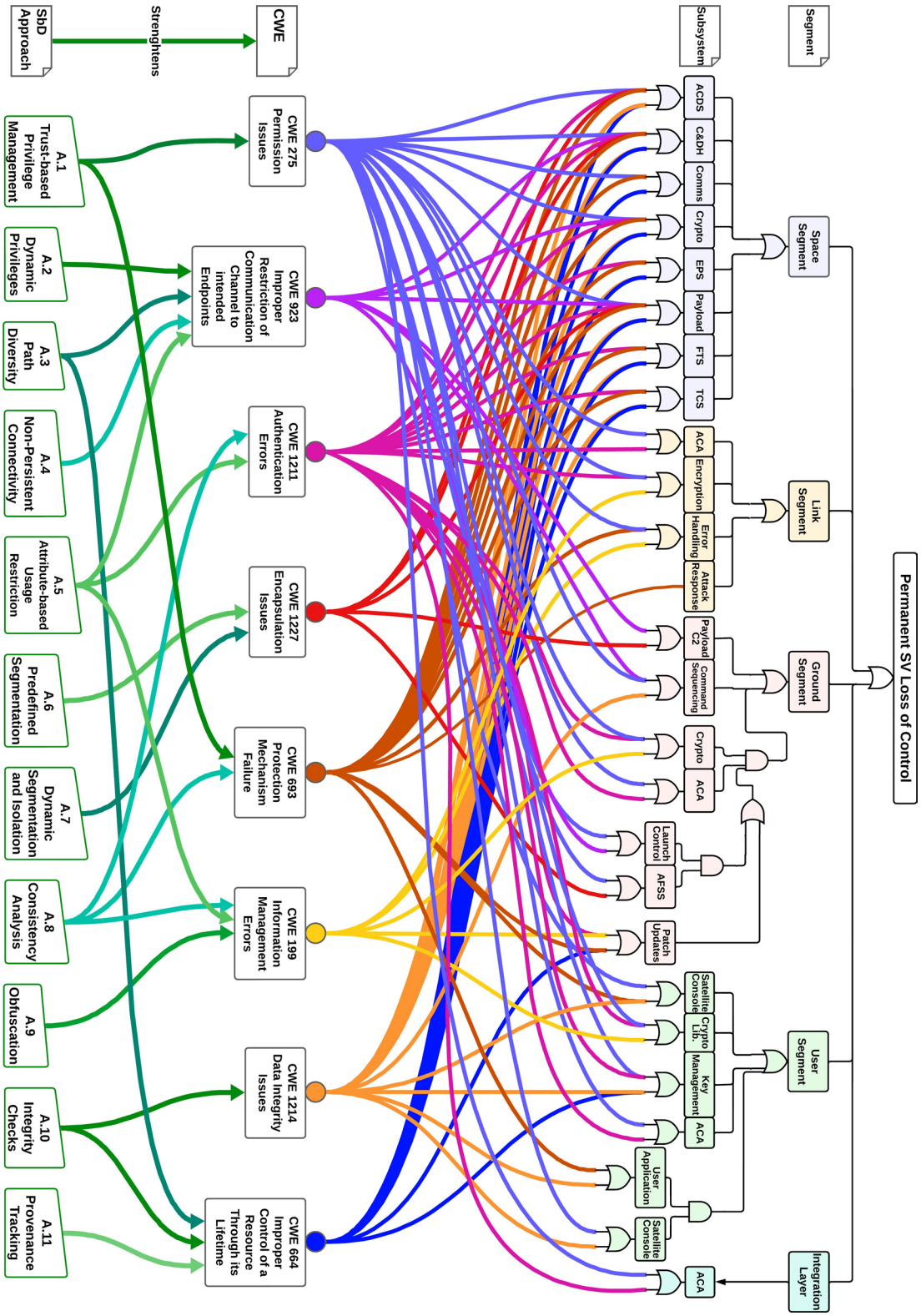


Fig. 1. Fault tree analysis of a space system for the scenario involving the permanent loss of satellite control.

of the Space Vehicle (SV). The different segments share the identified CWEs that can be targeted to achieve a permanent loss of the SV. Each segment is comprised of either subsystems or functions, each synergistically contributing to the segment's overall performance. Within the context of this paper, subsystems are defined as comprising both hardware and embedded software, while functions are exclusively software-based.

Space Segment consists of the space infrastructure that is not deployed on Earth and does not belong to the electromagnetic spectrum. In this analysis, the Space Segment coincides with the SV, which justifies the high impact many CWEs have on this segment.

Link Segment comprises the communication links between the Space Segment and the Ground and User Segments. In addition to the communication methods, radio frequency (RF) or free-space optical (FSO), this segment also consists of fundamental functions such as Authentication, Credentialing, Access Management / Authorization (ACA), Error Handling, and several others related to the assurance of quality and confidentiality of the information shared between the space segment and the other nodes of the space mission. A disruption in the link segment can lead to a permanent loss of the SV, as illustrated in the fault tree.

Ground Segment includes the infrastructure hosted on Earth that enables the communications and commanding of the Space Segment, the computing of the mission data, and the launch infrastructure. Functions and subsystems like Command Sequencing, Patch Updates, and ACA, if compromised, can lead to the permanent loss of the SV during the mission, while functions such as Launch Control and Automated Flight Safety System (AFSS) can lead to catastrophic faults during the launch operations,

User Segment comprises all interfaces and infrastructure accessible to users through which the services of a particular space system are made available. The User Segment is not interpreted only as a passive node of the system but can also issue commands and requests to other segments. Commercial services such as Ground Station as a Service (GSaaS) allow the user to directly control satellites and payloads. The active nature of this segment, embodied by functions like Satellite Console, can lead to severe faults in the Space Segment.

Integration Layer consists of a cohesive set of functions connecting and coordinating the segments of a space mission. It ensures interoperability among the segments by defining APIs and services. If compromised, it can lead to a permanent loss of the SV since the communication with other segments or external services would be interrupted.

B. Common Weakness Enumerations (CWEs)

This paper identifies eight CWEs variably shared by the four segments and the integration layer, using the following approach. First, we identified relevant threat techniques for each of the components identified for each segment in Figure 1. We then grouped the threat techniques based on the specific weaknesses that were being exploited. Finally, we mapped those weaknesses to corresponding CWEs from the **Software Development (699)** and **Research Concepts (1000)** views. We note that selected CWEs are not individual weaknesses, but rather refer to a group of weaknesses, referred to as *Pillars*, *Categories*, and *Classes* in CWE parlance.

CWE 275 Permission Issues (Category) This category groups classes of weaknesses related to improper assignment or handling of permissions. Permissions are linked to a resource, delineating which individuals or entities are permitted to access the resource and the actions they can undertake with that access, such as reading or modifying it.

CWE 923 Improper Restriction of Communication Channel to Intended Endpoints (Class): This class is a cluster of CWEs related to establishing a communication channel between a segment or component and an endpoint for privileged or protected operations without properly ensuring that it communicates with the correct endpoint. This weakness can be caused by unprotected primary and secondary channels (CWE 419 and CWE 420) or improper authentication mechanisms between system nodes (CWE 297 and CWE 940).

CWE 1211 Authentication Errors (Category): This category bundles weaknesses associated with the authentication components of the segments. If not addressed during the design or implementation of the space mission, the inability to verify a user's or agent's identity could compromise several functions and subsystems. Among the weaknesses of this category, the ones related to authentication bypassing (CWE 289, CWE 290, CWE 294) and lack of proper authentication mechanisms (i.e., CWE 295 and CWE 306) are hazardous for ACA and encryption functions.

CWE 1227 Encapsulation Issues (Category) The weaknesses of this category relate to data management and segmentation in a system. In particular, CWE 1100 *Insufficient Isolation of System-Dependent Functions* and CWE 1105 *Insufficient Encapsulation of Machine-Dependent Functionality* are significant in the space segment where attackers able to perform lateral movement across various subsystems could achieve the irreversible disruption of mission-critical elements.

CWE 693 Protection Mechanism Failure (Pillar) This pillar encompasses weaknesses related to three scenarios – a "missing," an "insufficient," and an "ignored" protection mechanism. CWE 311 *Missing Encryption of Sensitive Data* well describes the lack of protection mechanisms. CWE 345 *Insufficient Verification of Data Authenticity* and CWE 654 *Reliance on a Single Factor in a Security Decision* can instead represent the second and third scenarios.

CWE 199 Information Management Errors (Category) The improper handling of sensitive information in a system is asserted in the weaknesses grouped in this category. Examples such as CWE 201 *Insertion of Sensitive Information Into Sent Data* and CWE 359 *Exposure of Private Personal Information to an Unauthorized Actor* represent this category's emphasis on the system's lack of data and information protection.

CWE 1214 Data Integrity Issues (Category) Weaknesses in this category pertain to the data integrity aspects of a system. They concern the system's ability to maintain the integrity of various data types, including messages, resource files,

deployment files, and configuration files. This is exemplified by weaknesses such as CWE 322 *Key Exchange without Entity Authentication*, and CWE 347 *Improper Verification of Cryptographic Signature*. The CWE 829 *Inclusion of Functionality from Untrusted Control Sphere* demonstrates the importance of data integrity protection in systems like a space mission, interacting with heterogeneous or external services.

CWE 664 Improper Control of a Resource Through its Lifetime (Pillar) This pillar clusters weaknesses related to the inadequate or erroneous management of resources across their entire lifecycle, encompassing creation, usage, and release stages. In a space mission, these weaknesses also apply to concerns and threats derived from the supply chain.

C. Secure-by-Design Approaches

Given the rising impact of cyber attacks on critical infrastructure, several agencies are developing cyber resiliency and secure-by-design frameworks. In *Special Publication 800-160, Volume 2*, NIST has presented a cyber-resiliency framework consisting of techniques and implementation approaches. In November 2021, the UK National Cyber Security Centre (NCSC) released the *Device Security Guidance*, which provides cybersecurity principles aimed at connected systems' design and manufacturing phases.

In this work, we selected a subset of the principles outlined by NIST, since their more abstract nature makes them better suited for the heterogeneity of a space mission.

A.1 Trust-based Privilege Management: This approach defines, assigns, and maintains privileges based on the least privilege criteria in a system. It focuses on the separation of roles and the implementation of efficient and secure authorization methods. This approach is mapped to **CWE 275** and **CWE 693**, both related to permission and protection issues.

A.2 Dynamic Privileges: This approach indicates the necessity for varying privilege levels based on contextual or transient factors. Privileges based on the context and the system's current status help strengthen **CWE 923** by solving issues related to the low protection of secondary or transient channels and users of the system.

A.3 Path Diversity: This approach suggests equipping a system with multiple and independent paths to enable communications and C2 functions. This redundancy can strengthen **CWE 923** and **CWE 664**. In the first case, it provides secure solutions to corrupted secondary or external channels. In the second case, it provides alternative communication paths in case of faults generated by corrupted or flawed hardware and software components.

A.4 Non-Persistent Connectivity: Suited to address the **CWE 923**, this approach suggests managing connections in the system based on demand, terminating them when no longer needed. This can limit the consequences of corrupted channels among the space mission segments.

A.5 Attribute-based Usage Restriction: This approach highlights the importance of defining, assigning, maintaining, and applying user usage restrictions based on mission criticality. Restrictions rooted in criteria such as data sensitivity can strengthen **CWE 923**, **CWE 1211**, and **CWE 199**. In particular, this approach can reduce the risk of authentication bypass and the consequences of exposing sensitive data.

A.6 Predefined Segmentation: This approach suggests restricting and dividing functions and resources into segments and enclaves based on their criticality or trustworthiness. In this case, segmentation is defined *a priori* in the system's design phase. This principle is suited to address **CWE 1227**, given its relation to encapsulation issues in services and data.

A.7 Dynamic Segmentation and Isolation: Similarly, this approach suggests dividing functions and data into enclaves and protected segments. The difference lies in the dynamic nature of the segmentation, being informed by transient events. This approach concurs in strengthening the **CWE 1227**, addressing threats arising after the deployment of the system.

A.8 Consistency Analysis: This approach suggests adopting tools to check the consistency of ACA, information and process flows, and system policies. **CWE 1211**, **CWE 693**, and **CWE 199** share the dependency on robust authentication and data management and protection mechanisms. This approach can be instrumental in minimizing the possibility of attacks exploiting inconsistency in the security measures deployed in the various segments. If applied holistically to the entire system, it also provides consistency in the integration layer, ensuring compatibility between segments.

A.9 Obfuscation: As demonstrated by the substantial impact of **CWE 199** on the various segments and especially Ground, protecting information and credentials is crucial for the system's security. This approach suggests the consistent use of cryptographic and protection systems to reduce the risk of exposing sensitive data outside the allowed spheres.

A.10 Integrity Checks: This approach strengthens weaknesses related to the system's corruption or manipulation of data and components. It suggests applying and validating checks for the integrity of components to identify corrupted elements. It applies primarily to **CWE 1214** and **CWE 664**. In the first case, the data integrity is preserved in case of modifications. In the second one, events and faults related to the supply chain can be promptly identified and isolated.

A.11 Provenance Tracking: This approach indicates enforcing a system-wide identification and tracking of the provenance of data, software, and hardware elements. Verifying the source of components and functions is instrumental in strengthening supply chain weaknesses, like **CWE 664**.

V. Minimum Requirements Analysis

A. Space

For the defined priority of preventing permanent loss of control for the satellite, the security of the endpoint itself, the space segment, is crucial. As depicted in Figure 1, nearly every component of the SV, if caused to fail, will result in permanent loss of the vehicle. This is because the SV is a system of systems where the mission may end if a single subsystem fails. For example, the *C&DH* subsystem is responsible for the bus and commanding the various other vehicle subsystems. A compromise of the *C&DH* could lead

to the inability of critical signals to be sent to or reach the propulsion system or the attitude determination and control system for a necessary station-keeping maneuver. Similarly, such a compromise could cause uncontrollable spinning, which could permanently disable the functionality of the SV. For example, to show the importance of weaknesses mapping in the *C&DH* subsystem and related secure-by-design principles selection, it is beneficial to highlight **CWE 275** and **CWE 1227**. Weaknesses related to permission issues can allow the attacker to gain privileges instrumental in modifying on-board values that could lead to loss of control from the operator. Finally, encapsulation issues can be exploited to perform lateral movement in the SV and compromise other subsystems. For this reason, few secure-by-design approaches alone, **A.1**, **A.6**, and **A.7**, can reduce the impact of the cited CWEs with indirect benefits for the security of the entire space segment.

B. Link

The Link segment connects the SV to the rest of the space mission ecosystem. The corruption or disruption of this segment could permanently hinder the ability to communicate with the space segment. In this study, we identified several link functions that, if flawed by the above-mentioned CWEs, could lead to a permanent loss of availability of the SV. The **Error Handling (ECC/encoding/decoding)** function is a helpful example to demonstrate the link segment's crucial role in the scenario analyzed by this paper. This function comprises components like Forward Error Correction (FEC) Codes and error detection algorithms. Given their role in the segment, these components are susceptible to **CWE 275**, **CWE 693**, and **CWE 199**. Attacks exploiting permission issues and authentication errors could bypass the protection mechanisms of the segment and perform data analysis exploitation to inform further steps of the attack. In case of failures in the protection mechanisms and the information management, the attacker could alter the signal and deliberately inject errors in the signal, leading to disruptions in the communication link, possibly unrecoverable. For this reason, the **A.1**, **A.5**, **A.8**, and **A.9** secure-by-design approaches can be mapped back to the Error Handling function. A.1 and A.5 can reduce the risk of error handling corruption or exploitation to escalate the attack in the SV.

A.8 provides cross-validation tools to check the consistency of ACA, information and process flows, and system policies, with a multi-layer perspective. Namely, it strengthens the weaknesses of Authentication Errors, Protection Mechanism Failures and Information Management Errors, ensuring that the system's security capabilities are not used incompletely or inconsistently. Finally, adopting A.9 limits the adversarial capabilities of performing data analysis in sub-functions like FEC codes.

C. Ground

The Ground segment is vital for commanding and controlling every aspect of the space vehicle. Various ground components are essential for ensuring access and control of the spacecraft. Compromising any of these components could result in the permanent loss of control of the space vehicle. In Figure 1, we illustrate seven components that, if compromised, could lead to the permanent loss of control of the space vehicle. For instance, let's consider the *Patch Updates* component, which is responsible for sending software updates to the spacecraft. Any compromise of the patch updater allows an attacker to compromise the integrity of the spacecraft, potentially leading to loss of control. An attacker may achieve this by either inserting malicious firmware (CWE 199); compromising any weak protection mechanisms such as improper or weak encryption or improper or weak authentication mechanisms (CWE 693); or compromising the patches during their handling at various stages of the update pipeline (CWE 664). Following Figure 1, mitigating these weaknesses requires incorporating at a minimum the secure-by-design approaches such as performing proper privilege management (A.1), ensuring alternate communication channels for patch updates (A.3), proper usage restrictions (A.5), integrity checking of patches (A.10), and proper provenance tracking (A.11).

D. User

A critical aspect often overlooked is the potential for satellite manipulation from the user segment, rather than the traditional ground segment. Contrary to common perception, where the user segment is often associated with merely receiving data (as in GNSS receivers), sophisticated cyber threats now enable unauthorized control and command executions to exploit uplink communication from this segment.

We analyze an example using the *satellite console* component. The satellite console is a vital interface between operators and orbiting satellites, playing a fundamental role in the success of space missions. It enables critical functions such as maneuvering the spacecraft and managing onboard instruments – essentially controlling the satellite. Due to its direct influence over the spacecraft's behavior, the satellite console demands robust cybersecurity measures to prevent unauthorized access that could lead to potentially disastrous consequences, including the permanent loss of satellite control.

Potential vulnerabilities include **CWE-275** (Permission Issues), where excessive access rights could allow unauthorized individuals or processes to manipulate satellite functions. **CWE-693** (Protection Mechanism Failure) points to inadequate security controls, enabling unauthenticated commands or unauthorized actions from the user devices. Additionally, **CWE-1214** (Data Integrity Issues) highlights corrupted data risks, which could lead to erroneous command execution.

To address these vulnerabilities, secure-by-design principles play a critical role. **A.1** mandates that only authorized personnel have the necessary access rights to perform their duties, significantly limiting the potential exploitation of excessive permissions **(CWE-275)**. **A.8** involves continuously scrutinizing system behavior for anomalies. This proactive approach helps detect potential breaches or issues that could indicate a failure in protection mechanisms **(CWE-693)**. Finally, Integrity Checks **A.10** are essential to uphold the accu-

racy and authenticity of commands and data. Validating the integrity of information before processing directly addresses concerns surrounding data corruption **(CWE-1214)**, ensuring that commands originate from the intended source and have not been compromised. By implementing these secure-by-design principles, space systems can create a layered defense for the satellite console, enhancing its cyber resilience.

E. Integration

The Authentication, Credentialing, Access Management/Authorization (ACA) is a helpful example to demonstrate the integration layer's crucial role in the scenario analyzed by this paper. This function comprises components like Authentication Method and Credentials Storage. Given their role in the segment, these components are susceptible to **CWE 275** and **CWE 1211**. Attacks exploiting Permission Issues and Authentication Errors could bypass the protection mechanisms of the layer and compromise the APIs integrating the different segments to reach the SV to perform further steps of the attack. If an attacker can gain privileges in the Integration Layer or, later, in the Space Segment, they could be allowed to modify data and policies in the system, leading to a loss of control by the authorized user. For this reason, the **A.1**, **A.5**, and **A.8** secure-by-design approaches can be mapped back to the ACA function. A.1 and A.5 can reduce the risk of ACA being corrupted or exploited to escalate the attack in the SV. Finally, A.8 is beneficial to add redundancy to the ACA function, ensuring that information, credentials, or permissions shared by the Integration Layer are correct.

VI. Minimum Requirements Results

We solve for the most efficient combination of secure-by-design approaches by defining the design approaches and CWEs as a system of matrices. This system is defined in Equation 1, where $\boldsymbol{a}$ is a vector containing all of the possible design approaches $\{a_1, \ldots, a_{11}\}$, $\boldsymbol{c}$ is a vector of all of the possible CWEs as shown in Equation 2, and P is a binary matrix that translates which CWEs are covered by each design approach a_1 through a_{11}. The matrix P is computed using the diagram in Figure 1.

$$\boldsymbol{a} = P\boldsymbol{c} \tag{1}$$

$$\begin{bmatrix} a_1 \\ a_3 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \\ CWE_{275} \\ CWE_{923} \\ CWE_{1211} \\ CWE_{1227} \\ CWE_{693} \\ CWE_{199} \\ CWE_{1214} \\ CWE_{644} \\ \\ \\ \end{bmatrix} \tag{2}$$

To obtain the minimum number of design requirements needed to defend against all CWEs we can perform a row reduction of the system. This gives us Equation 3, where $\boldsymbol{a}$ is a vector filled with the resulting linear combinations of design approaches due to the linear operations performed in the row reduction, and P' is a matrix with its first row full of 1's, and every other value being 0 as seen in Equation 4. The first row of this system of equations provides a combination of design approaches that successfully cover all 8 of the CWEs.

$$\boldsymbol{a} = P'\boldsymbol{c} \tag{3}$$

$$P' = \begin{bmatrix} 1 & \ldots & 1 \\ 0 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 0 \end{bmatrix} \tag{4}$$

The result is $a_1 = a_1 + a_5 + a_7 + a_{10}$, which means that all CWE threats listed in $\boldsymbol{c}$ can be strengthened by only implementing these four design approaches. **Thus, the minimum secure-by-design approaches that can be applied to the relevant single-point-of-failure components to prevent the permanent loss of control of a satellite include: trust-based privilege management, attribute-based usage restriction, dynamic segmentation and isolation, and integrity checks**. We can verify that this is the lowest number of design approaches possible to cover all potential attacks using the following logic:

1) There are 3 unique rows within P that contain a singular 1, or cover a CWE that no other non-unique row covers.
2) The combination of these 3 rows does not cover all 8 CWEs.
3) Therefore, it is impossible to cover all 8 CWEs with any less than 4 design approaches.

The identification of the design principles informs the derivation of engineering requirements or 'shall' statements. The requirements derived from our minimum approaches can be established at various levels of abstraction at either the segment, subsystem, component, or sub-component level. Below is an example of how the four previously identified minimum design approaches can be transformed into minimum technical requirements for the SV at the subsystem level for the Attitude Determination and Control System (ADCS).

A.1 Trust-based Privilege Management

- The ADCS components shall implement least privilege.
- The ADCS privileges shall be designed as location-based and/or time-based.
- Actions on and from the ADCS shall require dual authorization.

A.5 Attribute-Based Usage Restriction

- The ADCS shall employ role-based access control.
- The ADCS shall employ attribute-based access control.
- The use of maintenance tools shall be restricted.

A.7 Dynamic Segmentation and Isolation

- The ADCS components shall be dynamically isolated.
- The ADCS shall be equipped with virtualized sandboxes or detonation chambers for eventual untrusted commands from other subsystems.
- The ADCS shall not accept commands or data from other subsystems without proper validation.

A.10 Integrity Checks

- The ADCS shall be equipped with automated tools for data quality checking.
- The ADCS shall be equipped with non-modifiable executables.
- The ADCS components shall perform attestation at each stage of the startup and ensure an overall trusted boot.
- The ADCS shall use information input validation.
- The ADCS shall perform integrity checks on connections and interactions with systems.
- The ADCS, upon detection of a potential integrity violation, shall provide the capability to audit the event and send data to C&DH for downlink to ground.

The above process would repeat for all the identified components of the subsystems in each segment.

VII. Discussion

The space community is at an inflection point. New space systems from the past decade have now outnumbered legacy systems in orbit. We can either continue developing and launching space systems that lack basic security or we can collectively acknowledge the existential security threat to our access to space. Such an acknowledgment would include taking the step of mandating that future space systems are designed with minimum security requirements.

A. Minimum Requirements Challenges

By engaging our proposed minimum requirements methodology for the provided example, and subsequently distilling the four security approaches into necessary requirements, we demonstrate how a space system developer can eliminate the capability of a threat actor to permanently deny control of your satellite. From a space developer or prime integrator standpoint, yes, this would require additional resources to enact these software measures and may require additional labor hours. From an acquisition specialist and procurement office standpoint, yes, it may require additional 'shall' statements and therefore necessitate a budgetary line item to accommodate this. However, we argue that the requirements stipulated will preserve our access to space and protect space investments. Additionally, adopting minimum requirements for space system cybersecurity can present implementation and policy challenges in the space sector. We acknowledge the sector's reliance on legacy systems, which would lack the capability to adopt these design requirements. Therefore, the secure-by-component minimum requirements we propose for new systems must be complemented by security controls described in Prior Art and threat information sharing through organizations such as the Space Information Sharing and Analysis Center (ISAC).

We are convinced that the cautions concerning implementing minimum security requirements fail to outweigh the benefits. The proposed requirements should be introduced gradually to the industry to entice collaboration across all of the key groups required to address the challenge and drive adoption.

B. Future Work

While applicable to national security and civil space assets, we expect our minimum requirements methodology to particularly support the commercial space sector given their financial sensitivities and need for highly focused minimum requirements. Future work includes investigating the unique features of the commercial space sector to evaluate their minimum requirements and the associated cost-benefit.

Next steps for the IEEE P3349 Working Group include engaging the minimum requirements methodology and example from this paper to begin writing requirements for the space, ground, link, user, and integration layer's secure blocks. Each segment will derive mission requirements from the down-selected secure-by-design approaches for their respective components and subcomponents identified on the fault tree. This will yield mission objective-relevant minimum requirements for secure blocks across the segments to preserve access to space and prevent the permanent loss of satellite control.

VIII. Conclusion

This paper demonstrates a method for establishing a dynamic set of minimum requirements based on mission objectives. The example mission objective identified for this paper included preserving access to space by preventing the permanent loss of control of a satellite. Using our method, we assessed that to avoid permanent loss of control, a space system designer must incorporate the design principles of trust-based privilege management, attribute-based usage restriction, dynamic segmentation and isolation, and integrity checks into the single-point-of-failure components identified across the space, ground, link, user and integration segments.

As the IEEE International Technical Standard for Space System Cybersecurity continues to develop, the minimum requirements method will aid the P3349 Working Group in identifying minimum requirements for each secure block. We encourage others to use this methodology to establish minimum requirements unique to their mission objectives. Critically, the minimum secure-by-component design requirements should be used in tandem with top-down, risk-based security controls for legacy systems.

Acknowledgement

A portion of this research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004). We would like to acknowledge the members of IEEE's Standard for Space System Cybersecurity Working Group P3349 for discussions leading to this method.

REFERENCES

[1] N. Boschetti, N. G. Gordon, and G. Falco, "Space cybersecurity lessons learned from the viasat cyberattack," in *ASCEND 2022*, 2022, p. 4380.

[2] G. Falco, W. Henry, M. Aliberti, B. Bailey, M. Bailly, S. Bonnart, N. Boschetti, M. Bottarelli, A. Byerly, J. Brule *et al.*, "An international technical standard for commercial space system cybersecurity-a call to action," in *ASCEND 2022*, 2022, p. 4302.

[3] *Back to the Building Blocks: A Path Toward Secure and Measurable Software*, February ed., The White House, 2024. [Online]. Available: https://www.whitehouse.gov/wp-content/uploads/2024/02/Final-ONCD-Technical-Report.pdf

[4] *Space System Protection Standard*, NASA-STD-1006A ed., National Aeronautics and Space Administration (NASA), 2022. [Online]. Available: https://standards.nasa.gov/standard/NASA/NASA-STD-1006

[5] *Science Mission Directorate (SMD) Launch Vehicle Secondary Payload Adapter Rideshare Users Guide with Do No Harm*, Rev. 3 ed., National Aeronautics and Space Administration (NASA), 2023. [Online]. Available: https://www.nasa.gov/wp-content/uploads/2023/09/smd-spa-rug-with-dnh-generic-july2023.pdf

[6] *Space Security:Best Practices Guide (BPG)*, Rev. B ed., National Aeronautics and Space Administration (NASA), 2024. [Online]. Available: https://swehb.nasa.gov/display/SWEHBVD/7.22+-+Space+Security%3A+Best+Practices+Guide

[7] *NPR 2810.1F: Security of Information and Information Systems*, National Aeronautics and Space Administration (NASA), 2022. [Online]. Available: https://nodis3.gsfc.nasa.gov/displayDir.cfm?Internal_ID=N_PR_2810_001F_&page_name=main&search_term=npr%202810%2E1f

[8] *Special Publication 800-160: Engineering Trustworthy Secure Systems*, Rev. 1 ed., National Institute of Standards and Technology (NIST), 2022. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/160/v1/r1/final

[9] *IR 8441: Cybersecurity Framework Profile for Hybrid Satellite Networks (HSN)*, National Institute of Standards and Technology (NIST), 2023. [Online]. Available: https://csrc.nist.gov/pubs/ir/8441/final

[10] *IR 8401: Satellite Ground Segment: Applying the Cybersecurity Framework to Satellite Command and Control*, National Institute of Standards and Technology (NIST), 2022. [Online]. Available: https://csrc.nist.gov/pubs/ir/8401/final

[11] M. Scholl and T. Suloway, "Introduction to cybersecurity for commercial satellite operations," *Draft NISTIR*, vol. 8270, 2021.

[12] *350.0-G: The Application of Security to CCSDS Protocols*, Issue 3 ed., The Consultative Committee for Space Data Systems (CCSDS), 2019. [Online]. Available: https://public.ccsds.org/Pubs/350x0g3.pdf

[13] *355.0-B: Space Data Link Security Protocol*, Issue 2 ed., The Consultative Committee for Space Data Systems (CCSDS), 2022. [Online]. Available: https://public.ccsds.org/Pubs/355x0b2.pdf

[14] *356.0-B: Network Layer Security Adaptation Profile*, Issue 1 ed., The Consultative Committee for Space Data Systems (CCSDS), 2018. [Online]. Available: https://public.ccsds.org/Pubs/356xb1.pdf

[15] *357.0-B: CCSDS Authentication Credentials*, Issue 1 ed., The Consultative Committee for Space Data Systems (CCSDS), 2019. [Online]. Available: https://public.ccsds.org/Pubs/356xb1.pdf

[16] *Cybersecurity Guidelines for Commercial Space Systems*, Ver. 1.1 ed., Space Industry Office, Manufacturing Industries Bureau, Ministry of Economy, Trade and Industry (METI), 2023. [Online]. Available: https://www.meti.go.jp/shingikai/mono_ info_ service/sangyo_cyber/wg_seido/wg_uchu_sangyo/pdf/20230331_1e.pdf

[17] *IT-Grundschutz Profile for Space Infrastructures*, Federal Office for Information Security, 2022. [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Grundschutz/profiles/Profile_Space-Infrastructures.pdf?__blob=publicationFile&v=2

[18] A. Viswanathan, B. Bailey, K. Tan, and G. Falco, "Secure-by-component: A system-of-systems design paradigm for securing space missions," in *Submitted to the IEEE Conference on Security for Space Systems*. IEEE, 2024.