Space System Survivability Against AI-Driven Cyberattacks

Applying Structural Security Design Principles for Mission Resilience

Dr. Ron Ross
CHIEF SCIENTIST, KAIPERSHIELD

Dr. Kymie Tan
CHIEF TECHNOLOGY OFFICER, KAIPERSHIELD

Abstract

Space systems have become indispensable to national security, economic activity, and civil society, making them prime targets for sophisticated adversaries. A new generation of AI-enabled offensive capabilities, referred to as "Mythos-class" capabilities, can autonomously analyze software, discover vulnerabilities, and generate functional exploit code at machine speed, representing a qualitative escalation in the scale, speed, and economics of cyber offense. For mission-critical space systems that cannot be rapidly patched or reconfigured once deployed, traditional cybersecurity approaches centered on vulnerability discovery and remediation remain necessary but are no longer sufficient. This article argues that the emergence of AI-enabled cyber threats makes a disciplined systems security engineering approach essential for protecting mission-critical systems by treating resilience as a fundamental architectural design objective. It demonstrates how the security design principles codified in NIST SP 800-160 provide the structural foundation for resilient space systems. Domain separation, together with eight complementary security design principles, constrains adversarial movement, limits the propagation of cyber effects and preserves mission continuity through structural design rather than reactive vulnerability management. The article describes how the design principles apply across the three interconnected segments of a space system (ground, link, and space) and presents findings from the NASA SunRISE pilot project, which applied these principles to a real satellite mission to generate empirical evidence of their effectiveness. The results demonstrate that trustworthy, resilient space systems are achieved not through the deployment of additional security products, but through the disciplined application of systems security engineering to the architecture of the system itself.

Introduction

Space systems, including satellites, ground stations, launch infrastructure, and the communication links that connect them, have become indispensable to national security, economic activity, and civil society. They underpin GPS navigation, intelligence collection, weather forecasting, missile warning, and global communications. That indispensability makes them high-value targets for sophisticated nation-state adversaries intent on subversion.

A new generation of AI-driven attack tools is reshaping the threat landscape, not by changing the nature of exploitation, but by fundamentally changing its economics. Large language models (LLMs) and AI agents are now capable of autonomous code analysis and exploit development including scanning software, identifying exploitable vulnerabilities, and generating functional exploit code with minimal human intervention and at machine speed. But exploit generation is only one dimension of the threat. AI enables the full offensive campaign: autonomous reconnaissance that maps attack surfaces at scale, campaign planning that prioritizes high-value targets and sequences attack chains, adaptive malware that modifies its behavior in response to defensive measures, and agentic orchestration that coordinates multi-stage intrusions across system boundaries without human direction.

These capabilities are exemplified by models such as Anthropic's Claude Mythos [6], an LLM developed for cybersecurity applications including vulnerability analysis. Throughout this article, the term "Mythos-class" refers not to a specific tool or threat actor, but to this broader class of AI-enabled offensive capabilities characterized by autonomous, machine-speed vulnerability discovery, weaponization, and campaign execution. The attack techniques themselves (e.g., initial access, lateral movement, privilege escalation, persistence) remain what they have always been. What *has* changed is that AI compresses the time, cost, and expertise required to execute them at scale. For military and commercial space systems, whose software stacks are complex, long-lived, and difficult to modify in orbit, this shift in the economics of exploitation represents a qualitative and quantitative escalation in risk, fundamentally changing the scale, speed, and economics of system exploitation.

That speed, scale, and economic shift exposes the fundamental limitation of the defensive strategies most organizations rely on today. Approaches centered on finding and fixing vulnerabilities assume that defenders can discover and remediate flaws faster than adversaries can discover and weaponize them. Against Mythos-class capabilities operating at machine speed across complex software stacks, that assumption no longer holds. Vulnerability management remains necessary, but it is no longer sufficient. When adversaries can enumerate an entire attack surface autonomously and generate functional exploits in hours, defenders cannot rely on patching to mitigate risk and protect mission-critical systems. Structural engineering (i.e., designing systems so that the consequences of any individual compromise are inherently bounded), becomes the primary mechanism for preserving mission resilience.

The root cause of most successful cyberattacks is not a failure of security controls, but a failure of system architecture and design. Systems security engineering views security, like safety and reliability, as an *emergent property* of a well-designed and well-architected system. The security design principles described in NIST SP 800-160 [1] provide the structural foundation and architectural discipline required to protect space systems against this emerging class of AI-driven threats. This article examines how those principles apply to space systems and describes an active experiment that is generating the assurance evidence needed to support structural security and risk-based decisions about the capability, protection, and resilience of mission-critical systems.

The Space System Attack Surface

A space system is not a single entity. It is a complex system of hardware, software, and firmware components with an integrated architecture spanning three interconnected segments, each with distinct functions, trust relationships, and attack surfaces. Mission threads (i.e., the end-to-end sequences of system functions that deliver mission capability), traverse all three segments, creating architectural dependencies and control authority relationships that must be explicitly understood and protected. An adversary who maps these dependencies gains a blueprint for disrupting mission outcomes, not merely individual components.

Ground Segment. This includes mission operations centers, command and control (C2) systems, data processing facilities, and ground-based networks. The ground segment provides the operational interface for mission execution, consequently it has become the primary entry point for adversaries intent on subversion. A successful compromise of the ground segment can provide access to the uplink, enabling unauthorized commands to be transmitted to the spacecraft.

Link Segment. This includes the radio frequency (RF) and optical communication paths connecting the ground infrastructure to the spacecraft. Threats include jamming (denial of service), spoofing (false

signal injection), and eavesdropping. Adversaries capable of compromising the uplink can potentially inject commands into the link segment (uplink) and gain control of the spacecraft.

Space Segment. This includes the spacecraft's bus, payload, onboard computers, and flight software. Once on orbit, mitigating vulnerabilities is constrained by bandwidth, operational schedules, and mission risk, allowing vulnerabilities to persist far longer than they would in terrestrial systems.

A Mythos-class cyberattack tool applied to the ground segment software stack can autonomously enumerate attack paths, chain multiple vulnerabilities, and traverse toward the command uplink, potentially causing severe or catastrophic impacts across all three segments. The cyberattack lifecycle closely parallels the model for terrestrial systems: initial access, execution and establishment, discovery, lateral movement, privilege escalation, persistence and defense evasion, and mission execution.

Domain Separation: The Foundation of Space System Defense

NIST SP 800-160 defines the security design principle of *domain separation*, or isolation, as the physical or logical separation of domains with distinctly different protection needs. Domain separation delivers two critical protective effects: *susceptibility reduction* (shielding a domain from external influence) and *containment* (preventing a compromised domain from propagating damage outward).

Applied to space systems, domain separation means treating the ground segment, link segment, and space segment as distinct security domains with controlled, mediated interfaces between them. Within the ground segment, the mission operations network, the command generation system, the command uplink path, and the administrative/IT systems must be separated domains, not a flat network where a compromised workstation can reach the uplink subsystem.

Without this structure, an adversary who gains initial access to a ground segment IT system can traverse laterally to command generation, escalate privileges, and ultimately inject unauthorized commands to the spacecraft potentially affecting mission resilience and survivability. Domain separation makes that traversal structurally impossible across domain boundaries, not merely detectable after the fact.

Complementary Security Design Principles

Domain separation is most effective when combined with complementary security design principles from NIST SP 800-160. Eight of the principles are especially relevant to space systems:

Least Privilege. *Each system element is allocated privileges that are necessary to accomplish its specified functions but no more.*

The principle of least privilege directly limits the scope of damage achievable from within any single domain. A compromised process or account can only access what it legitimately needs, preventing credential-based escalation within a domain. Least privilege and domain separation are synergistic: domain separation sets the coarse-grained boundary between security domains; least privilege enforces fine-grained access control within each domain. For space systems, each software component and operator account is granted only the access required for its defined function. This means that ground operators responsible for telemetry monitoring have no need for command uplink privileges. Enforcing this separation of domains limits what an adversary can achieve even after compromising an account.

Mediated Access. *All access to and operations on system elements are mediated.*

The principle of mediated access addresses *access* to the system element (only authorized entities gain access) and *use* of the system element (only authorized operations are permitted). Both are relevant to defending against a Mythos-class attack. Mediated access creates mandatory chokepoints at every domain boundary and within domains, forcing all interactions through a policy-based access decision that cyberattack tools cannot bypass without detection. For space systems, all access to critical system elements, particularly the command uplink path, is enforced through a policy-based reference monitor that validates every request. An AI-driven attack tool operating through a compromised element cannot bypass a mediated boundary without generating a detectable policy violation.

Least Functionality. *Each system element has the capability to accomplish its required functions but no more.*

The principle of least functionality reduces the attack surface available to adversaries within each domain. Mythos-class tools thrive on the richness of the average enterprise environment: SMB file sharing, PowerShell remoting, DCOM, and legacy administrative interfaces all provide execution and traversal vectors. Disabling unneeded services eliminates entire categories of exploitation. Least functionality applied rigorously within each domain including disabling or blocking services not required for the domain's mission, eliminates these attack surfaces. For space systems, each domain presents only the services its mission requires. For example, ground data system software is configured to provide only the services required for mission operations. Unnecessary services, protocols, ports, and interfaces are disabled. This directly reduces the attack surface available to autonomous vulnerability-scanning tools.

Anomaly Detection. *Any salient anomaly in the system or its environment is detected in a timely manner that enables effective response action.*

The principle of anomaly detection requires monitoring system behaviors and outcomes to determine deviations from design intent. Anomaly detection provides the sensor layer that detects Mythos-class tool activity at domain boundaries and within domains, enabling response before mission execution. Domain separation changes the value proposition of anomaly detection fundamentally. In a flat architecture, the adversary's lateral movement traffic looks like normal east-west traffic. There is no structural difference between legitimate and malicious communications. In a domain-separated architecture, domain boundaries define what is normal. Any unauthorized attempt to cross a boundary is structurally anomalous. This transforms anomaly detection from a needle-in-a-haystack problem into a targeted, high-confidence signal, critical for space systems where operational windows are time-constrained and response time is limited.

Defense in Depth. *Loss is prevented or minimized by employing multiple coordinated mechanisms.*

The principle of defense in depth establishes three pillars: (1) multiple lines of defense (i.e., barriers) should be placed along loss scenario sequences; (2) loss control should not rely on a single defensive element; and (3) successive barriers should be diverse in nature (i.e., technical, operational, and organizational). For space systems, defense in depth means that no single failure, whether of a firewall, an access control, or a ground station operator, results in mission compromise. Adversaries must overcome independent, diverse barriers to progress from initial access in an IT network to the command uplink that controls the spacecraft. When combined with domain separation, defense in depth ensures that each domain boundary represents an additional, independently enforced line of defense.

Least Persistence. *System elements and other resources are available, accessible, and able to fulfill their design intent only for the time for which they are needed.*

The principle of least persistence defines three conditions that must be satisfied for a system element to be usable: (1) presence (installed and configured), (2) accessibility (able to be invoked), and (3) ability to function (powered and enabled). Least persistence attacks all three: elements are not installed until needed, are deactivated when not in use, and have access mediated even when present. For space systems, this principle is especially relevant to ground segment software. Command uplink subsystems, cryptographic key management systems, and privileged administrative interfaces should be active only during authorized operational windows. Mythos-class attack tools depend on persistent footholds, including registry entries, scheduled tasks, and service installations, to maintain access across reboots and sessions. Least persistence systematically denies those persistence mechanisms.

Least Sharing. *System resources are shared among system elements only when necessary and among as few elements as possible.*

The principle of least sharing eliminates the shared mechanisms and pathways that Mythos-class tools exploit for lateral movement and information access. Every shared mechanism represents a potential information path between system elements and must be carefully designed. Shared infrastructure, including shared storage, shared naming services, shared authentication infrastructure, and shared administrative channels, provides the information paths that lateral movement exploits. In a space system ground architecture, shared network segments between mission operations and administrative IT are a common source of lateral movement vulnerability. Least sharing reinforces domain separation by ensuring that the mechanisms shared between domains are explicitly designed, minimized, and controlled, not the accidental consequence of convenience-driven architectural choices.

Reduced Complexity. *The system design is as simple as practicable.*

Complexity can be found in the system structure, interfaces, dependencies, data and control flows, and the interaction of the system with its operational environment. A complex design increases uncertainty, leads to errors, and hinders confidence in the system's security posture. Reduced complexity contributes to more informed trade decisions and decreases uncertainty in the design and its realization as a system. For space systems, unnecessary complexity in the ground segment software architecture and the interfaces between ground, link, and space segments creates attack surface that is difficult to analyze, verify, and protect. Mythos-class cyberattack tools thrive on complexity: the richer and more interdependent the software environment, the more unintended paths exist for exploitation. Reduced complexity directly limits the attack surface available to autonomous vulnerability discovery.

The SunRISE Pilot Project: Principles in Practice

The principles described in this article are not hypothetical. NIST collaborated with NASA and the Jet Propulsion Laboratory (JPL) in an experiment designed to address system security and mission resilience from a systems engineering perspective, grounded in the NIST SP 800-160 security design principles and applied directly to a real space mission.

The subject of the initial experiment was SunRISE (Sun Radio Interferometer Space Experiment), a NASA mission consisting of an array of six toaster-size CubeSats that were designed to study solar radio bursts and their role in driving space weather [2]. The mission engineers who developed the SunRISE system recreated the SDLC process in a simulation mode using a cyber analysis and visualization tool known as

CAVE. [1] The NASA-JPL team applied a targeted subset of the security design principles and resiliency techniques from NIST SP 800-160, Volumes 1 and 2 [3] to guide and inform their engineering decisions throughout the simulation. The design principles and resiliency techniques were applied in the context of current threat intelligence and threat data, not as a static, retrospective checklist, but as an active engineering input to design decisions.

The rigor of the SunRISE experiment and the measurability of its results, distinguish it from prior advocacy for systems security engineering. The experiment used a controlled methodology: an exact digital replica of the SunRISE Ground Data System (GDS) was created, producing two identical twins subjected to the same designed attacks. Twin A proceeded through the traditional cybersecurity process with a vulnerability-centric, compliance-based approach; Twin B had the NIST SP 800-160 security design principles integrated from the design phase forward. Red (attacker) and blue (defender) teams were separated and did not communicate, ensuring that the attacks were developed independently of the defenses. The preliminary results were unambiguous. Mean time to detect a data tampering attack was reduced from weeks in Twin A to minutes in Twin B. Detection of mission data destruction was reduced from weeks to seconds. A malware-based data tampering incursion executed against both twins was not detected by the traditional cybersecurity approach on Twin A but was detected by the systems security engineering approach on Twin B.

Beyond detection performance, the SunRISE experiment demonstrated that early lifecycle integration of security design principles produces compounding, low-cost benefits that traditional, siloed cybersecurity approaches cannot replicate after the fact. Under the *Reduced Complexity* principle, mission engineers redesigned existing telemetry sensors to serve dual purposes, detecting operational faults and potential adversarial activity, at no added complexity or cost. Under *Mediated Access*, mission-engineers designed targeted anti-virus scan placement that was simultaneously more effective and more economical than what the traditional, controls-only approach produced. Adding equivalent system monitoring and logging capability, capturing CPU usage, memory patterns, and communication profiles for adversarial detection, required approximately two to three lines of code during the design phase. The same capability, added at the end of the system lifecycle to an already-integrated system, was assessed as prohibitively costly. These results provide empirical grounding for the central argument of this article: that structural security applied as an engineering discipline from the system lifecycle, is both more effective and more economical than reactive vulnerability management applied at the end of it.

Applying these principles is not without cost. The SunRISE pilot exposed a fundamental organizational challenge: cybersecurity engineers framed adversarial threats as the primary mission risk, while mission engineers regarded them as one of several co-equal hazards alongside structural failures and human error. This disconnect served to effectively block the successful integration of security into mission systems. Timing is the dominant cost variable. Security capabilities integrated at the design phase cost as little as two to three lines of code; the equivalent capability added after a compliance-based assessment was deemed to be prohibitively expensive. For legacy space systems and on-orbit assets, where architectural redesign is constrained by bandwidth, operational schedules, and mission risk, full

[1] The cyber analysis and visualization (CAVE) tool used by the mission engineers in the SunRISE experiment, was designed by NASA's Jet Propulsion Laboratory. The tool has been licensed to KAIperShield, a Caltech/NASA JPL spinout and Caltech-seeded company developing an AI-powered platform for systems security engineering and mission resilience. The flagship KAIperShield platform, AULIS, combines AI, graph analytics, digital twins, and advanced systems modeling to operationalize the systems security engineering concepts in NIST SP 800-160.

application of these principles may require significant re-engineering investment. These tradeoffs are real, but they argue for earlier adoption, not against the engineering approach itself.
The results from the SunRISE pilot project demonstrated the value of a disciplined, engineering-focused approach to securing and protecting mission-critical systems and high value assets. These findings carry significant implications for how the broader space systems community and other critical infrastructure sectors approach the security of mission-critical systems:

- **Systems security engineering must become the default.** Trustworthy secure systems engineering, using state-of-the-practice security design principles, concepts, and techniques alongside all-source threat intelligence, must become the standard methodology for how mission-critical systems and high value assets are designed, developed, protected, deployed, and sustained over time.

- **Component-level security matters.** Individual system components for mission-critical systems and high value assets should also employ security design principles, concepts, and techniques. This includes the use of formal methods and memory-safe programming languages at the component level, not only at the system architecture level.

- **Evidence-based assurance is the measure of trustworthiness.** Evidence gathered from the systems engineering process, not compliance attestations, should determine the trustworthiness of mission-critical systems and how resilient those systems are with respect to their ability to withstand cyberattacks from nation-state adversaries intent on subversion, manipulation, damage, and destruction.

- **Laws, policies, regulations, and methodologies must evolve.** The current regulatory, cybersecurity, and acquisition frameworks do not consistently require or incentivize trustworthy secure systems engineering. The initial SunRISE experiment and its follow-on for SPHEREx [4] are building the empirical case for that change.

Conclusion

AI-driven attack tools capable of autonomous exploit development represent a qualitative escalation in the threat to space systems. The long operational lifetimes of spacecraft, the constraints on in-orbit patching, and the cascading consequences of a compromised command uplink make structural security, not just operational security, essential.

Domain separation, as defined in NIST SP 800-160, provides the architectural foundation for a defense that cannot be bypassed through credential theft or lateral movement alone. When combined with least privilege, mediated access, least functionality, anomaly detection, defense in depth, least persistence, least sharing, and reduced complexity, it creates a mutually reinforcing structural defense that imposes cost on the adversary at every phase of the cyberattack lifecycle, from initial access through mission execution.

The NASA SunRISE pilot project translated these principles from theory into empirical evidence. By applying security design principles to a real mission in simulation mode and comparing the results against the as-built system, the project generated the data needed to make the case for trustworthy secure systems engineering as the default approach for all mission-critical space programs.

The case for structural security does not end at the boundary of the space systems community. CISA identifies sixteen critical infrastructure sectors whose disruption or destruction would have a debilitating

effect on national security, economic stability, or public health and safety. These sectors include Chemical, Commercial Facilities, Communications, Critical Manufacturing, Dams, Defense Industrial Base, Emergency Services, Energy, Financial Services, Food and Agriculture, Government Facilities, Healthcare and Public Health, Information Technology, Nuclear Reactors Materials and Waste, Transportation Systems, and Water and Wastewater Systems [5].

In every one of these sectors, there are mission-critical systems and high-value assets for which the loss of confidentiality, integrity, or availability could produce consequences as severe as those that motivate structural security for NASA's space systems. The same security design principles and engineering discipline that make it structurally difficult for an adversary to traverse from a ground station IT network to a spacecraft command uplink also make it structurally difficult to traverse from a business network to an industrial control system managing a power grid or dam, from an enterprise IT environment to the operational technology controlling a water treatment facility, or from a hospital network to the medical devices sustaining patient life.

The SunRISE pilot project demonstrated these principles in the specific context of a satellite mission. Its broader lesson is universal: wherever a system's failure could produce catastrophic and potentially irreversible consequences, the structural security engineering discipline demonstrated at SunRISE is the appropriate and necessary path to mission resilience.

Trustworthy secure systems are not built by deploying more security products. They are built by applying rigorous systems security engineering discipline to the architecture of the system itself, ensuring that the structure of the system makes damaging attacks difficult by design, not merely detectable after the fact. Moreover, organizations must be able to take advantage of the same cutting-edge AI technology to operationalize systems security engineering and do so at scale, particularly as AI-boosted attacks change the threat landscape.

The failure mode is not inadequate technology. It is inadequate engineering.

References

[1] Ross, R., Winstead, M., & McEvilley, M. (2022). NIST Special Publication 800-160, Volume 1, Revision 1: *Engineering Trustworthy Secure Systems*. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-160v1r1

[2] NASA Jet Propulsion Laboratory. SunRISE: Sun Radio Interferometer Space Experiment. https://www.jpl.nasa.gov/missions/sun-radio-interferometer-space-experiment-sunrise

[3] Ross, R. et al. (2021). NIST Special Publication 800-160, Volume 2, Revision 1: Developing Cyber-Resilient Systems. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-160v2r1

[4] NASA Jet Propulsion Laboratory. SPHEREx: Spectro-Photometer for the History of the Universe, Epoch of Reionization and Ices Explorer. https://www.jpl.nasa.gov/missions/spherex

[5] Cybersecurity and Infrastructure Security Agency (CISA). Critical Infrastructure Sectors. https://www.cisa.gov/topics/critical-infrastructure-security-and-resilience/critical-infrastructure-sectors

[6] Anthropic. (2026). Introducing Claude Fable 5 and Claude Mythos 5. https://platform.claude.com/docs/en/about-claude/models/introducing-claude-fable-5-and-claude-mythos-5