NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0011	Vulnerability Scanning	Vulnerability scanning systematically identifies known security weaknesses in commercial off-the-shelf (COTS) and open-source software (OSS) components, including vulnerable dependencies and outdated software versions, across spacecraft and ground system environments. Custom-developed code requires separate analysis approaches, such as static and dynamic code analysis, that fall outside the scope of this countermeasure; however, COTS, OSS, and third-party dependencies incorporated into custom-developed software remain within the scope of vulnerability scanning and SCA. Scanning programs should incorporate software composition analysis (SCA) to detect vulnerabilities introduced through third-party libraries and dependency chains, which represent a significant and frequently underestimated attack surface in modern space systems. Scanning tools and processes should conform to recognized interoperability standards that support enumeration of platforms, software flaws, and configuration weaknesses; standardized formatting of checklists and test procedures; and consistent measurement of vulnerability impact. Adherence to these standards enables automation of key vulnerability management workflow steps, facilitates tool interoperability across the mission ecosystem, and produces outputs that are comparable across programs and organizations. Vulnerability scanning shall be performed against the approved as-built or deployed software baseline using methods that do not jeopardize mission operations. Active scanning of live spacecraft, safety-critical ground systems, or operational technology should occur only after the scanning method has been authorized and evaluated for operational impact. Where live scanning presents unacceptable risk, scanning should be performed against representative software images, firmware packages, or test environments that match the approved operational baseline.	Functional	Intersects With	Vulnerability Scanning	VPM-12	Mechanisms exist to detect vulnerabilities and configuration errors by routine vulnerability scanning of systems and applications.	5		VPM-06
CM0041	User Training	Human factors represent one of the most consistently exploited attack vectors against space mission systems; adversaries routinely use spear phishing, social engineering, and other user-interaction techniques to gain initial access to mission networks, development environments, and operational infrastructure. User training reduces this attack surface by building personnel awareness of adversary techniques and the behaviors required to recognize and resist them. Training must be role-differentiated: general security awareness training is required for all personnel with access to mission systems, while personnel with assigned security roles and responsibilities require additional role-based training commensurate with the specific duties and access rights they hold. Training must be completed before a user is granted access to mission systems or authorized to perform assigned security duties; updated when system changes introduce new threats, procedures, or access controls that alter the security context in which personnel operate; and refreshed at least annually, or more frequently if the mission's security policy specifies a shorter interval. Training program effectiveness should be measurable, with completion records maintained and knowledge validation incorporated to confirm that training objectives are being achieved rather than simply delivered.	Functional	Intersects With	Role-Based Security, Compliance & Resilience Training	SAT-05	Mechanisms exist to provide role-based security, compliance and resilience-related training: (1) Before authorizing access to the system or performing assigned duties; (2) When required by system changes; and (3) Annually thereafter.	5		SAT-03
CM0041	User Training	Human factors represent one of the most consistently exploited attack vectors against space mission systems; adversaries routinely use spear phishing, social engineering, and other user-interaction techniques to gain initial access to mission networks, development environments, and operational infrastructure. User training reduces this attack surface by building personnel awareness of adversary techniques and the behaviors required to recognize and resist them. Training must be role-differentiated: general security awareness training is required for all personnel with access to mission systems, while personnel with assigned security roles and responsibilities require additional role-based training commensurate with the specific duties and access rights they hold. Training must be completed before a user is granted access to mission systems or authorized to perform assigned security duties; updated when system changes introduce new threats, procedures, or access controls that alter the security context in which personnel operate; and refreshed at least annually, or more frequently if the mission's security policy specifies a shorter interval. Training program effectiveness should be measurable, with completion records maintained and knowledge validation incorporated to confirm that training objectives are being achieved rather than simply delivered.	Functional	Intersects With	Sensitive / Regulated Data Storage, Handling & Processing	SAT-05.1	Mechanisms exist to ensure that every user accessing a system processing, storing or transmitting sensitive and/or regulated data is formally trained in data handling requirements.	5		SAT-03.3
CM0041	User Training	Human factors represent one of the most consistently exploited attack vectors against space mission systems; adversaries routinely use spear phishing, social engineering, and other user-interaction techniques to gain initial access to mission networks, development environments, and operational infrastructure. User training reduces this attack surface by building personnel awareness of adversary techniques and the behaviors required to recognize and resist them. Training must be role-differentiated: general security awareness training is required for all personnel with access to mission systems, while personnel with assigned security roles and responsibilities require additional role-based training commensurate with the specific duties and access rights they hold. Training must be completed before a user is granted access to mission systems or authorized to perform assigned security duties; updated when system changes introduce new threats, procedures, or access controls that alter the security context in which personnel operate; and refreshed at least annually, or more frequently if the mission's security policy specifies a shorter interval. Training program effectiveness should be measurable, with completion records maintained and knowledge validation incorporated to confirm that training objectives are being achieved rather than simply delivered.	Functional	Intersects With	Cyber Threat Environment Situational Awareness	SAT-06	Mechanisms exist to provide role-based security, compliance and resilience awareness training that is current and relevant to the cyber threats that users might encounter in day-to-day business operations.	5		SAT-03.6
CM0041	User Training	Human factors represent one of the most consistently exploited attack vectors against space mission systems; adversaries routinely use spear phishing, social engineering, and other user-interaction techniques to gain initial access to mission networks, development environments, and operational infrastructure. User training reduces this attack surface by building personnel awareness of adversary techniques and the behaviors required to recognize and resist them. Training must be role-differentiated: general security awareness training is required for all personnel with access to mission systems, while personnel with assigned security roles and responsibilities require additional role-based training commensurate with the specific duties and access rights they hold. Training must be completed before a user is granted access to mission systems or authorized to perform assigned security duties; updated when system changes introduce new threats, procedures, or access controls that alter the security context in which personnel operate; and refreshed at least annually, or more frequently if the mission's security policy specifies a shorter interval. Training program effectiveness should be measurable, with completion records maintained and knowledge validation incorporated to confirm that training objectives are being achieved rather than simply delivered.	Functional	Intersects With	Suspicious Communications & Anomalous System Behavior	SAT-06.2	Mechanisms exist to provide training to personnel on organization-defined indicators of malware to recognize suspicious communications and anomalous behavior.	5		SAT-03.2

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0010	Update Software	Regular software and firmware updates are a primary mechanism for reducing exploitation risk by remediating known vulnerabilities before adversaries can leverage them against mission systems. Updates must undergo suitable regression testing to verify that security patches do not introduce functional defects or degrade safety-critical behaviors before deployment to operational systems. Release cadence should be governed by a mission-defined update frequency that balances vulnerability severity, exploitability, mission risk tolerance, testing requirements, and operational constraints. The program should define maximum allowable remediation or deployment timeframes (e.g., 30 days) for applicable vulnerability severities rather than applying a single update interval to all software and firmware. Update scheduling must account for operational constraints, coordinating patch deployment with mission downtime windows to avoid disrupting critical operations. Following a successful update, superseded software versions should be removed from active operational use unless retained as an authorized recovery image. Verified restoration images, commonly referred to as gold images, should be retained under access and integrity controls to support recovery when an update introduces unacceptable system behavior or an authorized rollback is	Functional	Intersects With	Software & Firmware Patching	VPM-08	Mechanisms exist to conduct software patching for all deployed Technology Assets, Applications and/or Services (TAAS), including firmware.	5		VPM-05
CM0054	Two-Person Rule	The two-person rule (TPR) requires that all access to and actions within systems possessing command-level authority over the spacecraft be conducted in the continuous presence of two separately authorized individuals, neither of whom alone can complete the access or action. This control reduces the opportunity for unauthorized, coerced, erroneous, or malicious commanding by requiring two independently authorized individuals to participate in the protected access or action. Its effectiveness depends on individual attribution, independent review, resistance to credential sharing or account misuse, and protection against collusion or bypass of the enforcement mechanism. The TPR requires two distinct, appropriately authorized individuals to participate in the same protected access or action. Participation may be physically co-located, remotely performed through authenticated technical controls, or implemented through a combination of physical and logical mechanisms, as defined by mission policy. The second individual must independently review and approve the specific access or action before it is completed. Under CM0054, the two-person rule applies to access and actions involving systems with command-level access to the spacecraft to include reprograming the flight computer/flight software. If the mission tailors the rule to selected commanding functions, the retained scope, excluded access and actions, compensating controls, and associated risk acceptance must be explicitly documented and approved.	Functional	Intersects With	Dual Authorization for Changes To Critical / Sensitive Technology Assets, Applications and/or Services (TAAS)	CHG-04.3	Mechanisms exist to enforce a two-person rule for implementing changes to critical and/or sensitive Technology Assets, Applications and/or Services (TAAS).	5		HRS-12.1
CM0029	TRANSEC	Transmission security (TRANSEC) is the component of communications security (COMSEC) concerned with protecting the characteristics of the transmission itself, as distinct from protecting the content of the information being communicated. TRANSEC controls reduce the likelihood, effectiveness, or operational impact of transmission interception, signal disruption, communications deception, and exploitation of transmission characteristics within the defined threat model. Applicable TRANSEC techniques include jam-resistant waveforms that increase resistance to jamming and communications deception, spread spectrum and frequency hopping techniques that reduce signal predictability and improve resistance to interception and disruption, low probability of intercept and low probability of detection (LPI/LPD) signal designs that reduce transmission observability, and transmission scheduling or pattern discipline that limits the intelligence value of traffic analysis. TRANSEC requirements should be applied to mission communication links according to link criticality (e.g., TT&C, crosslinks), threat exposure, and operational consequence. TRANSEC must be treated as a distinct layer of protection complementary to, but not a substitute for, cryptographic protection of	Functional	No Relationship	N/A	N/A	N/A	0		
CM0073	Traffic Flow Analysis Defense	Traffic flow analysis attacks enable adversaries to derive operationally significant intelligence from observable transmission characteristics, including message timing, volume, duration, periodicity, and routing information, without decrypting the content of communications. Even when link encryption is in place, unprotected traffic patterns can reveal spacecraft operational schedules, command activity, contact windows with specific ground stations, and anomalous events that provide adversaries with actionable mission intelligence. Traffic flow analysis defense encompasses a set of techniques applied to protect the confidentiality of transmission metadata on telemetry, tracking, and commanding (TT&C) and data links, as well as onboard communications where applicable. Applicable techniques include padding transmissions to normalize message lengths and volumes, introducing artificial traffic during idle periods to obscure true contact patterns and event timing, obfuscating routing information and endpoint identities, varying transmission periodicity to defeat statistical pattern recognition, and frustrating traffic volume and duration analysis through active obfuscation methods. These controls are a complement to, but distinct from, cryptographic content protection, and should be applied based on a threat-informed assessment of the value of traffic metadata to potential adversaries	Functional	Intersects With	Analyze Traffic for Covert Exfiltration	MON-20.1	Automated mechanisms exist to analyze network traffic to detect covert data exfiltration.	5		MON-11.1
CM0073	Traffic Flow Analysis Defense	Traffic flow analysis attacks enable adversaries to derive operationally significant intelligence from observable transmission characteristics, including message timing, volume, duration, periodicity, and routing information, without decrypting the content of communications. Even when link encryption is in place, unprotected traffic patterns can reveal spacecraft operational schedules, command activity, contact windows with specific ground stations, and anomalous events that provide adversaries with actionable mission intelligence. Traffic flow analysis defense encompasses a set of techniques applied to protect the confidentiality of transmission metadata on telemetry, tracking, and commanding (TT&C) and data links, as well as onboard communications where applicable. Applicable techniques include padding transmissions to normalize message lengths and volumes, introducing artificial traffic during idle periods to obscure true contact patterns and event timing, obfuscating routing information and endpoint identities, varying transmission periodicity to defeat statistical pattern recognition, and frustrating traffic volume and duration analysis through active obfuscation methods. These controls are a complement to, but distinct from, cryptographic content protection, and should be applied based on a threat-informed assessment of the value of traffic metadata to potential adversaries	Functional	Intersects With	Inbound & Outbound Communications Traffic	MON-22	Mechanisms exist to continuously monitor inbound and outbound communications traffic for unusual or unauthorized activities or conditions.	5		MON-01.3

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: SPARTA Countermeasure (4.0)
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0073	Traffic Flow Analysis Defense	Traffic flow analysis attacks enable adversaries to derive operationally significant intelligence from observable transmission characteristics, including message timing, volume, duration, periodicity, and routing information, without decrypting the content of communications. Even when link encryption is in place, unprotected traffic patterns can reveal spacecraft operational schedules, command activity, contact windows with specific ground stations, and anomalous events that provide adversaries with actionable mission intelligence. Traffic flow analysis defense encompasses a set of techniques applied to protect the confidentiality of transmission metadata on telemetry, tracking, and commanding (TT&C) and data links, as well as onboard communications where applicable. Applicable techniques include padding transmissions to normalize message lengths and volumes, introducing artificial traffic during idle periods to obscure true contact patterns and event timing, obfuscating routing information and endpoint identities, varying transmission periodicity to defeat statistical pattern recognition, and frustrating traffic volume and duration analysis through active obfuscation methods. These controls are a complement to, but distinct from, cryptographic content protection, and should be applied based on a threat-informed assessment of the value of traffic metadata to potential adversaries	Functional	Intersects With	Network Intrusion Detection / Prevention Systems (NIDS / NIPS)	NET-31	Mechanisms exist to employ Network Intrusion Detection / Prevention Systems (NIDS/NIPS) to detect and/or prevent intrusions into the network.	5		NET-08
CM0020	Threat modeling	Threat modeling is a structured analytical process that identifies, enumerates, and prioritizes potential threats to a system by systematically examining assets, trust boundaries, data flows, and adversary capabilities relative to the system's architecture. Applied in combination with attack surface analysis and vulnerability analysis, threat modeling produces an integrated picture of where the system is most exposed and what the consequences of successful exploitation would be. Analysis should draw on findings from similar systems, components, or services where applicable, leveraging documented threat experience from comparable missions or architectures to avoid re-learning known lessons. The outputs of threat modeling must directly inform design decisions throughout the development process, with attack surface reduction treated as a design objective rather than a post-development hardening activity: interfaces, services, protocols, and code paths that are not necessary to mission function should be eliminated or constrained before they become embedded in the architecture. Threat model artifacts should be treated as living documents, updated as the system design evolves and as new threat intelligence becomes available.	Functional	Intersects With	Threat Modeling	TDA-05.2	Mechanisms exist to perform threat modelling and other secure design techniques, to ensure that threats to software and solutions are identified and accounted for.	5		TDA-06.2
CM0009	Threat Intelligence Program	A threat intelligence program enables an organization to systematically collect, analyze, and apply information about adversary capabilities, infrastructure, and intent to inform defensive priorities and drive risk-informed security decisions across the mission lifecycle. For space missions specifically, this may include leveraging available all-source intelligence services or commercial satellite imagery to identify and monitor adversary infrastructure development and acquisition activities that may signal emerging threats to mission assets. Threat intelligence outputs should be operationalized into concrete adjustments to defensive architecture, monitoring priorities, and incident response posture rather than treated as informational products alone. Direct countermeasures against adversary infrastructure identified through this program will fall outside the scope of the mission in the majority of cases; the primary value of the program is in generating actionable awareness that sharpens the	Functional	Intersects With	Threat Intelligence Program	THR-02	Mechanisms exist to implement a threat intelligence program that includes a cross-organization information-sharing capability that can influence the development of the system and security architectures, selection of security solutions, monitoring, threat hunting, response and recovery activities.	5		THR-01
CM0003	TEMPEST	TEMPEST controls (i.e., emissions security (EMSEC)) protect spacecraft system components, internal data communications, and communication buses against side-channel and proximity-based attacks that exploit unintended electromagnetic, electrical, or acoustic emanations. Critical components must be enclosed within appropriate casings or shielding structures that attenuate unintended emissions to levels that deny adversaries the ability to reconstruct processed data or infer system state from externally observable signals. Shielding must extend to internal buses and data pathways, not only to individual processing elements, as inter-component communications represent a significant and often overlooked emanations surface. The physical enclosure strategy must be integrated with the broader system architecture so that shielding effectiveness is not degraded by penetrations, connectors, or cable routing that create unintended emissions paths. During sustainment & maintenance, Spacecraft TEMPEST and EMSEC protections are primarily established during design, fabrication, and integration, but sustainment remains applicable through configuration control, review of deployment-state or hardware changes, preservation of qualification evidence, assessment of relevant anomalies, and evaluation of refurbishment, replacement, or follow-on production changes. The guidance below addresses these spacecraft considerations as well as applicable ground-segment maintenance activities.	Functional	No Relationship	N/A	N/A	N/A	0		
CM0057	Tamper Resistant Body	A tamper-resistant physical enclosure increases the effort, time, and equipment required to physically probe, observe, remove, or modify protected spacecraft sensor nodes and embedded components. The enclosure must be designed for the specific physical-access and side-channel threats being addressed and should not be assumed to prevent every invasive or non-invasive attack. A passive tamper-resistant body can provide physical and side-channel protection without continuous processing or electrical power, which may make it suitable for resource-constrained sensor nodes. The design trade must also account for mass, volume, thermal performance, manufacturability, inspection, repairability, qualification, and lifecycle cost. Enclosures incorporating active sensing or response mechanisms require power and must be evaluated separately from fully passive designs. The physical security design must distinguish among tamper resistance, which impedes access; tamper evidence, which leaves observable indications of attempted access; tamper detection, which senses an attempt while it occurs; and tamper response, which protects designated sensitive assets after detection. The required properties and response behavior must be selected according to the protected component, threat model, and mission consequence of both successful tampering and false activation.	Functional	Intersects With	Logical Tampering Protection	AST-29.2	Mechanisms exist to assess the integrity of critical Technology Assets, Applications and/or Services (TAAS) to detect evidence of tampering, where: (1) Logical assessments evaluate the integrity of critical components (e.g., configuration settings); and (2) Physical assessments evaluate assets for evidence of unauthorized access and/or modifications.	5		AST-15

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0028	Tamper Protection	Tamper protection encompasses physical and logical controls that detect, deter, and respond to unauthorized modification of mission hardware and software throughout the acquisition, transit, storage, integration, and operational phases of the mission lifecycle. Physical tamper protection requires inspection of hardware at defined custody transfer points to detect evidence of unauthorized access or modification, and requires the use of tamper-evident packaging and sealing mechanisms during shipping and receiving to ensure that any interference with equipment in transit is detectable upon arrival. Logical tamper protection addresses the integrity of software and firmware through layered techniques including code obfuscation, integrity checks that verify software has not been altered from its approved state, and runtime integrity monitoring mechanisms such as self-checking code. Watchdog processes may supplement these controls by detecting abnormal execution or failure to complete expected processing, but do not by themselves verify software integrity. Physical and logical tamper protections are complementary and should be applied together where warranted by system criticality, threat exposure, and lifecycle conditions. Physical access to hardware can enable logical tampering, while failure to verify software integrity can reduce the assurance provided by physical controls.	Functional	Intersects With	Logical Tampering Protection	AST-29.2	Mechanisms exist to assess the integrity of critical Technology Assets, Applications and/or Services (TAAS) to detect evidence of tampering, where: (1) Logical assessments evaluate the integrity of critical components (e.g., configuration settings); and (2) Physical assessments evaluate assets for evidence of unauthorized access and/or modifications.	5		AST-15
CM0028	Tamper Protection	Tamper protection encompasses physical and logical controls that detect, deter, and respond to unauthorized modification of mission hardware and software throughout the acquisition, transit, storage, integration, and operational phases of the mission lifecycle. Physical tamper protection requires inspection of hardware at defined custody transfer points to detect evidence of unauthorized access or modification, and requires the use of tamper-evident packaging and sealing mechanisms during shipping and receiving to ensure that any interference with equipment in transit is detectable upon arrival. Logical tamper protection addresses the integrity of software and firmware through layered techniques including code obfuscation, integrity checks that verify software has not been altered from its approved state, and runtime integrity monitoring mechanisms such as self-checking code. Watchdog processes may supplement these controls by detecting abnormal execution or failure to complete expected processing, but do not by themselves verify software integrity. Physical and logical tamper protections are complementary and should be applied together where warranted by system criticality, threat exposure, and lifecycle conditions. Physical access to hardware can enable logical tampering, while failure to verify software integrity can reduce the assurance provided by physical controls.	Functional	Intersects With	Product Tampering and Counterfeiting (PTC)	TDA-09	Mechanisms exist to maintain awareness of component authenticity by developing and implementing Product Tampering and Counterfeiting (PTC) practices that include the means to detect and prevent counterfeit components.	5		TDA-11
CM0025	Supplier Review	A supplier review is a structured pre-contract assessment conducted before entering into any agreement with a contractor or subcontractor for the acquisition of systems, system components, or system services. The review evaluates the prospective supplier's security posture, trustworthiness, and capability to deliver components or services that meet the mission's integrity and assurance requirements, before contractual commitments are made and before the supplier gains access to mission information or influence over mission systems. Supplier reviews reduce the risk of introducing supply chain vulnerabilities through poorly qualified, compromised, or adversary-influenced suppliers at any tier of the acquisition chain. The rigor and depth of the review should be calibrated to the criticality of the components or services being acquired, with suppliers of mission-critical hardware, software, or services subject to the most intensive assessment. Supplier review findings should inform not only the decision to contract but also the specific security requirements, oversight provisions, and flow-down obligations included in the resulting agreement.	Functional	Intersects With	Third-Party Management	TPM-02	Mechanisms exist to facilitate the implementation of third-party management controls.	5		TPM-01
CM0025	Supplier Review	A supplier review is a structured pre-contract assessment conducted before entering into any agreement with a contractor or subcontractor for the acquisition of systems, system components, or system services. The review evaluates the prospective supplier's security posture, trustworthiness, and capability to deliver components or services that meet the mission's integrity and assurance requirements, before contractual commitments are made and before the supplier gains access to mission information or influence over mission systems. Supplier reviews reduce the risk of introducing supply chain vulnerabilities through poorly qualified, compromised, or adversary-influenced suppliers at any tier of the acquisition chain. The rigor and depth of the review should be calibrated to the criticality of the components or services being acquired, with suppliers of mission-critical hardware, software, or services subject to the most intensive assessment. Supplier review findings should inform not only the decision to contract but also the specific security requirements, oversight provisions, and flow-down obligations included in the resulting agreement.	Functional	Intersects With	Third-Party Risk Assessments & Approvals	TPM-10	Mechanisms exist to conduct a risk assessment prior to the acquisition or outsourcing of technology-related Technology Assets, Applications and/or Services (TAAS).	5		TPM-04.1
CM0080	Stealth Technology	Spacecraft stealth encompasses design and operational techniques that reduce a satellite's detectability and trackability by adversary space surveillance systems, increasing the cost and difficulty of adversary targeting, tracking, and characterization efforts. Design-based approaches include reducing physical size to decrease radar cross-section (RCS), applying radar-absorbing coatings, using radar-deflecting geometric shapes, and controlling the emission or reflection of radar, optical, and infrared (IR) energy to minimize the observable signatures that surveillance sensors rely upon. Operational stealth techniques include optimizing maneuver profiles to avoid detection by known ground-based or space-based tracking sensors, executing maneuvers at unexpected times or with trajectories that complicate orbit determination, and employing active measures such as radar jamming or spoofing to degrade tracking accuracy. These approaches collectively raise the adversary's intelligence collection burden, degrade the accuracy of targeting solutions, and reduce the predictability of the spacecraft's future position, complicating the planning and execution of both kinetic and directed energy counterspace attacks. Stealth is a design philosophy and operational discipline that must be balanced against mission functional requirements, as size reductions and coating applications that reduce observability may affect payload capacity, thermal management, and	Functional	No Relationship	N/A	N/A	N/A	0		
CM0019	Static Analysis	Static source code analysis examines software without executing it, identifying security-relevant weaknesses in the codebase before they can be exploited at runtime. Static analysis must be performed across all available source code. The mission's weakness prioritization (i.e., CM0016) process should guide tool configuration and finding triage without unnecessarily excluding other weakness classes supported by the tools. The static analysis program must employ no fewer than two distinct tools with complementary language and weakness-detection capabilities. The tools should be applied at defined points throughout the development and maintenance lifecycle to provide overlapping and complementary analysis coverage and reduce the likelihood that exploitable weaknesses persist into integration or deployment. Tool selection should account for language compatibility, weakness class coverage, and the ability to produce findings in formats that support triage and remediation tracking. Static analysis should be integrated into the software build pipeline as an automated, recurring activity rather than conducted as a periodic manual exercise, ensuring that new code contributions are analyzed continuously throughout	Functional	Intersects With	Static Code Analysis	TDA-17.1	Mechanisms exist to require the developers of Technology Assets, Applications and/or Services (TAAS) to employ static code analysis tools to identify and remediate common flaws and document the results of the analysis.	5		TDA-09.2

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0078	Space-Based Radio Frequency Mapping	Space-based radio frequency (RF) mapping can provide broad-area monitoring and analysis of RF activity affecting space systems in orbit and on the ground. Depending on the sensor architecture, it may support recurring or persistent detection, signal characterization, and geolocation of interference sources, with performance determined by factors such as frequency coverage, signal strength, antenna pattern, sensor geometry, and revisit rate. By correlating RF observations with mission link performance, operators can better distinguish potential jamming or spoofing from unintentional interference and estimate relevant signal characteristics and source location. Although these observations do not independently establish intent, attribution, or a precise emitter location in every case, they provide an important intelligence layer that supports faster investigation and more informed defensive decisions when communications degradation occurs.	Functional	No Relationship	N/A	N/A	N/A	0		
CM0077	Space Domain Awareness	Space domain awareness (SDA) enables mission owners to detect and characterize objects, behaviors, environmental conditions, and anomalous events that may affect their space systems. When correlated with other intelligence and mission data, SDA can also support assessment of possible threats and attribution. SDA encompasses the tracking and cataloging of space objects, prediction of future object positions, monitoring of the space environment and space weather, and characterization of the capabilities and behaviors of on-orbit objects. SDA data must provide the accuracy, timeliness, coverage, and characterization needed for the mission's defined decisions. Publicly available data may support general awareness but may be insufficient for time-sensitive conjunction, proximity, or threat assessment; appropriate government, commercial, partner, or owner-operator data should be obtained where required. SDA is generated by a diverse sensor architecture spanning terrestrial optical, infrared, and radar systems and space-based sensors including inspector satellites capable of close-approach observation. The SDA landscape is increasingly populated by national space agencies, military programs, allied partners, commercial providers, and amateur tracking communities, making the space environment progressively more transparent and creating opportunities for mission owners to leverage diverse data sources to build a more complete operational picture.	Functional	No Relationship	N/A	N/A	N/A	0		
CM0007	Software Version Numbers	Software version information for commercial off-the-shelf (COTS), open-source software (OSS), firmware, operating systems, libraries, middleware, bootloaders, software-defined radios, and other software-enabled components used in spacecraft and ground systems must be protected from unauthorized or unnecessary disclosure. Exact component versions can allow adversaries to correlate an identified product or library with public vulnerability databases, vendor advisories, exploit repositories, known configuration weaknesses, and software-specific backdoor or supply-chain opportunities. Version information may be exposed directly through telemetry fields, network and service banners, diagnostic commands, management interfaces, error messages, log outputs, crash reports, configuration files, package manifests, filenames, firmware headers, debug symbols, update metadata, and publicly released documentation. Versions may also be inferred indirectly through protocol behavior, command responses, file hashes, default configurations, timing characteristics, or software-specific error conditions. These disclosure paths should be identified and controlled according to mission risk. Authoritative version information must remain available through controlled mechanisms to authorized developers, maintainers, operators, assessors, and incident responders. CM0007 is intended to limit unauthorized disclosure of exact software versions, not to eliminate internal software identification, configuration tracking, or diagnostic capability. Version-number protection increases the effort required for adversary reconnaissance but does not prevent active fingerprinting or mitigate vulnerabilities present in the software.	Functional	Intersects With	Commercial Off-The-Shelf (COTS) Security Solutions	TDA-11.3	Mechanisms exist to utilize only Commercial Off-the-Shelf (COTS) security products.	5		TDA-03
CM0007	Software Version Numbers	Software version information for commercial off-the-shelf (COTS), open-source software (OSS), firmware, operating systems, libraries, middleware, bootloaders, software-defined radios, and other software-enabled components used in spacecraft and ground systems must be protected from unauthorized or unnecessary disclosure. Exact component versions can allow adversaries to correlate an identified product or library with public vulnerability databases, vendor advisories, exploit repositories, known configuration weaknesses, and software-specific backdoor or supply-chain opportunities. Version information may be exposed directly through telemetry fields, network and service banners, diagnostic commands, management interfaces, error messages, log outputs, crash reports, configuration files, package manifests, filenames, firmware headers, debug symbols, update metadata, and publicly released documentation. Versions may also be inferred indirectly through protocol behavior, command responses, file hashes, default configurations, timing characteristics, or software-specific error conditions. These disclosure paths should be identified and controlled according to mission risk. Authoritative version information must remain available through controlled mechanisms to authorized developers, maintainers, operators, assessors, and incident responders. CM0007 is intended to limit unauthorized disclosure of exact software versions, not to eliminate internal software identification, configuration tracking, or diagnostic capability. Version-number protection increases the effort required for adversary reconnaissance but does not prevent active fingerprinting or mitigate vulnerabilities present in the software.	Functional	Intersects With	Vulnerability & Patch Management Program (VPMP)	VPM-02	Mechanisms exist to facilitate the implementation and monitoring of risk-based vulnerability management capabilities.	5		VPM-01
CM0015	Software Source Control	Binary or machine-executable code obtained from sources that provide no warranty and no access to the corresponding source code must not be incorporated into spacecraft or ground systems. This prohibition addresses a fundamental software assurance gap: without source code, missions cannot perform source-based static analysis or independently review and modify implementation details, and may have reduced ability to assess, repair, or extend the software. Code from sources with limited or no warranty and no source code provision may be difficult to independently analyze, repair, or extend and leaves the mission dependent on supplier assurances and remediation capabilities. This countermeasure applies throughout the software supply chain, including components integrated by subcontractors.	Functional	No Relationship	N/A	N/A	N/A	0		

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0021	Software Digital Signature	Each software or firmware image subject to installation or update control shall be verified using an approved digital signature and an approved trust anchor before installation or activation. The trust anchor may be represented by a certificate or by a directly provisioned verification key, depending on the approved trust architecture. Successful verification establishes that the image was signed by an authorized signing identity and has not been modified since signing; it does not establish that the signed code is non-malicious, vulnerability-free, or operationally safe. Signature verification confirms the integrity and approved origin of the software but does not, by itself, prevent installation of an older validly signed version; update authorization and rollback protections must be enforced separately. The verification mechanism must be implemented such that it cannot be bypassed through operational commands, configuration changes, or software updates, and must reject any component whose signature is absent or invalid, or whose signing certificate or trust anchor is not recognized and approved by the mission. Digital signature enforcement complements but is distinct from the boot-time chain of trust established through secure boot; it applies to software installation and update events throughout the operational lifecycle, not only at system	Functional	Intersects With	Signed Components	CHG-13	Mechanisms exist to prevent the installation of software and firmware components without verification that the component has been digitally signed using an organization-approved certificate authority.	5		CHG-04.2
CM0012	Software Bill of Materials	A software bill of materials (SBOM) is a structured inventory of software components, libraries, dependencies, and associated metadata comprising a delivered system, spanning first-party code and available third-party and open-source supply-chain information. The SBOM serves as the foundational reference for continuous vulnerability management: by cross-correlating the component inventory against known vulnerability databases, such as those cataloging common vulnerabilities and exposures (CVEs), mission owners and operators can rapidly identify which specific system components are affected by newly disclosed vulnerabilities and prioritize remediation accordingly. SBOM generation must cover the full software supply chain, including transitive dependencies that are not explicitly declared in top-level manifests, as these indirect inclusions represent a persistent and frequently exploited blind spot in software inventory programs. An SBOM may reveal component composition and vulnerability-relevant information that could assist adversary reconnaissance. Its classification, sensitivity, dissemination, and handling requirements shall be determined using applicable mission guidance, contractual requirements, and a documented disclosure-risk assessment. If deemed to have sensitive information then the handling controls applied should align to other mission-critical security documentation as defined in CM0001.	Functional	Intersects With	Software Bill of Materials (SBOM)	TDA-21.4	Mechanisms exist to generate, or obtain, a Software Bill of Materials (SBOM) for Technology Assets, Applications and/or Services (TAAS) that lists software packages in use, including versions and applicable licenses.	5		TDA-04.2
CM0067	Smart Contracts	Smart contracts provide an automated, consensus-based enforcement mechanism for security protocols governing data exchange across a spacecraft bus, particularly in hosted payload architectures where multiple independently operated payloads or subsystems share common infrastructure. By encoding security rules as deterministic contract logic, participating validator nodes can evaluate authenticated observations of bus activity and reach consensus on whether a defined violation occurred. A trusted bus controller, gateway, or access-control mechanism must then enforce any approved restriction because the smart contract itself cannot directly observe or block physical bus traffic. This approach is directly applicable to scenarios where a hosted payload is compromised or behaves maliciously, as the smart contract framework can detect the violation, achieve consensus among the remaining bus participants, and enforce exclusion of the offending payload before the attack propagates to other bus components or the host spacecraft. The consensus mechanism should prevent a single participating node from causing payload exclusion when the approved quorum and membership rules require agreement from multiple independent validators. Its effectiveness depends on the validator architecture, quorum threshold, communication assumptions, and number of compromised or unavailable validators. Smart contract-based enforcement is particularly valuable in multi-tenant spacecraft architectures where payload operators are organizationally distinct from the spacecraft bus operator and cannot be governed through a single chain of	Functional	No Relationship	N/A	N/A	N/A	0		
CM0040	Shared Resource Leakage	Shared system resources (e.g., processor registers, main memory, secondary storage, cache) may retain residual data or security-relevant state after a process releases them for reuse. If a subsequent process can access that residual information, it may obtain data from the prior process, including sensitive information or encrypted representations of information that were not intended to cross process or partition boundaries. This countermeasure requires that shared resources be sanitized, zeroed, or otherwise cleared of prior process data before being allocated to a new process, ensuring that information transfer between processes occurs only through explicitly authorized channels. The protection must apply to encrypted representations as well as plaintext because encrypted data remains information belonging to the prior process and must not be transferred to another process solely because its contents are not immediately readable. This is particularly significant in space system environments where multiple processes of varying criticality and trust levels may share the same hardware resources.	Functional	Intersects With	Information In Shared Resources	SEA-14	Mechanisms exist to prevent unauthorized and unintended information transfer via shared system resources.	5		SEA-05
CM0036	Session Termination	Where session-oriented communications are used between ground systems and spacecraft, between ground system components, or across internal spacecraft interfaces, the associated connection or session must be terminated upon completion or after a period of inactivity exceeding a threshold defined in the mission's concept of operations (CONOPS). Failure to terminate idle or completed sessions leaves authenticated connections open and exploitable, providing adversaries with an opportunity to inject commands or data into an existing authenticated session without needing to complete the authentication process independently. Inactivity timeout thresholds must be established through the CONOPS process rather than set arbitrarily, balancing the security benefit of rapid session termination against the operational consequences of terminating a session that an operator or automated process may legitimately require across a contact gap or processing delay. Session termination must invalidate the associated session identifiers, authorization state, and cryptographic session state so that the terminated session cannot be reused. Where secure session resumption is operationally required, it must use a mission-approved, cryptographically protected mechanism with defined validity limits and must not permit reuse of expired, revoked, or invalidated session state.	Functional	Intersects With	Session Termination	CFG-04.5	Automated mechanisms exist to log out users, both locally on the network and for remote sessions, at the end of the session or after an organization-defined period of inactivity.	5		IAC-25

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0038	Segmentation	Segmentation establishes physical or logical isolation boundaries between spacecraft system components and functional domains to limit the propagation of compromise, contain the consequences of a security failure, and enforce controlled information flow across the mission architecture. Mission-critical functions must be isolated from non-mission-critical functions through enforced partition boundaries that control access to and protect the integrity of the hardware, software, and firmware implementing those functions. Information flow between partitioned components or applications must be explicitly authorized by security policy; any flow not affirmatively permitted is denied by default. Boundary protections must be implemented to separate spacecraft bus, communications, and payload components, preventing a compromise in one domain from directly affecting the others. Information crossing the spacecraft boundary shall receive confidentiality protection where required by its classification, sensitivity, or mission risk. Command-bearing and other security-critical exchanges shall receive the mission-required authentication, integrity, authorization, and anti-replay protections regardless of whether confidentiality is required. These controls collectively implement a defense-in-depth architecture in which an adversary who gains a foothold in one partition faces enforced barriers before reaching mission-critical assets.	Functional	Intersects With	Network Segmentation (macrosegmentation)	NET-06	Mechanisms exist to implement network segmentation within network architectures to isolate Technology Assets, Applications and/or Services (TAAS) from other network resources.	5		NET-06
CM0008	Security Testing Results	Penetration testing, vulnerability scanning, fuzzing, code analysis, adversarial testing, and other security assessment activities are essential for validating the security posture of space systems. However, the artifacts generated by these activities are themselves sensitive mission information requiring protection. Security testing artifacts may include final reports, raw scanner output, working notes, exploit code, proof-of-concept scripts, packet captures, RF recordings, screenshots, logs, memory or core dumps, test credentials, command sequences, telemetry mappings, architecture diagrams, test configurations, remediation records, and copies of software or data collected during testing. Security testing artifacts may reveal exploitable weaknesses, attack paths, security-control gaps, spacecraft and ground-system interfaces, command and telemetry behavior, link characteristics, safety interlocks, test accounts, and system-specific vulnerability conditions. Detailed evidence identifying an exploitable weakness, affected interface, and validated attack method may provide an adversary with much of the information needed to reproduce an attack. Security testing results shall be categorized, marked, handled, stored, transmitted, shared, retained, and disposed of according to their information classification, CUI category, proprietary restrictions, contractual requirements, and mission sensitivity. Protection shall address both unauthorized disclosure and unauthorized modification, because alteration or deletion of test results could conceal vulnerabilities, misrepresent remediation status, or undermine security decisions. Access shall be limited by role and need to know throughout the artifact lifecycle, beginning when the data is generated and continuing through authorized disposition.	Functional	Intersects With	Vulnerability Scanning	VPM-12	Mechanisms exist to detect vulnerabilities and configuration errors by routine vulnerability scanning of systems and applications.	5		VPM-06
CM0008	Security Testing Results	Penetration testing, vulnerability scanning, fuzzing, code analysis, adversarial testing, and other security assessment activities are essential for validating the security posture of space systems. However, the artifacts generated by these activities are themselves sensitive mission information requiring protection. Security testing artifacts may include final reports, raw scanner output, working notes, exploit code, proof-of-concept scripts, packet captures, RF recordings, screenshots, logs, memory or core dumps, test credentials, command sequences, telemetry mappings, architecture diagrams, test configurations, remediation records, and copies of software or data collected during testing. Security testing artifacts may reveal exploitable weaknesses, attack paths, security-control gaps, spacecraft and ground-system interfaces, command and telemetry behavior, link characteristics, safety interlocks, test accounts, and system-specific vulnerability conditions. Detailed evidence identifying an exploitable weakness, affected interface, and validated attack method may provide an adversary with much of the information needed to reproduce an attack. Security testing results shall be categorized, marked, handled, stored, transmitted, shared, retained, and disposed of according to their information classification, CUI category, proprietary restrictions, contractual requirements, and mission sensitivity. Protection shall address both unauthorized disclosure and unauthorized modification, because alteration or deletion of test results could conceal vulnerabilities, misrepresent remediation status, or undermine security decisions. Access shall be limited by role and need to know throughout the artifact lifecycle, beginning when the data is generated and continuing through authorized disposition.	Functional	Intersects With	Penetration Testing	VPM-18	Mechanisms exist to conduct penetration testing on Technology Assets, Applications and/or Services (TAAS).	5		VPM-07
CM0055	Secure Command Mode(s)	Secure command modes provide additional layers of restriction on spacecraft command acceptance beyond standard authentication and encryption, constraining when, where, and under what operational conditions the spacecraft will process commands. These supplemental controls reduce the window of opportunity for unauthorized commanding by limiting command receptivity to defined parameters that an adversary would need to satisfy simultaneously with authentication requirements, substantially increasing the difficulty of a successful command injection attack. Specific implementations include geographic restriction, in which the spacecraft accepts commands only when in contact with designated ground station locations; operational mode restrictions, in which special flight software (FSW) modes must be active before certain command categories are accepted; and temporal controls, in which the spacecraft enforces time-bounded windows during which commands are valid. These mechanisms complement command authentication, integrity protection, anti-replay controls, and authorization and do not replace them. Encryption should also be applied where command confidentiality is required. Secure command modes may combine geographic, temporal, operational-state, source, or other mission-defined conditions according to the active command policy. Secure command modes helps create a multi-dimensional command acceptance policy that an adversary must defeat in its entirety to achieve unauthorized command execution.	Functional	No Relationship	N/A	N/A	N/A	0		
CM0014	Secure boot	Secure boot establishes and enforces a cryptographically verified chain of trust from a hardware-anchored root of trust (RoT) through each applicable stage of the startup sequence to the operating system or flight software image. Each stage in the chain must verify the integrity and authenticity of the next before transferring execution control. The boot policy must also prevent execution of unauthorized or revoked images, including unauthorized rollback to an older but validly signed software version. The trust anchor and initial verification function should be immutable after provisioning or protected by hardware-enforced mechanisms that prevent unauthorized modification and preserve their integrity. Components implementing the RoT must also be qualified for the expected mission radiation environment. Radiation tolerance addresses the reliability of the trust anchor, while immutability or protected update mechanisms address its resistance to unauthorized modification. This is particularly critical where radiation-induced bit flips and the physical inaccessibility of on-orbit hardware make a tamper-resistant, immutable hardware anchor essential to sustained boot integrity across the mission lifetime.	Functional	Intersects With	Boot Process Integrity	END-11.4	Automated mechanisms exist to verify the integrity of the boot process of systems.	5		END-06.5

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0014	Secure boot	Secure boot establishes and enforces a cryptographically verified chain of trust from a hardware-anchored root of trust (RoT) through each applicable stage of the startup sequence to the operating system or flight software image. Each stage in the chain must verify the integrity and authenticity of the next before transferring execution control. The boot policy must also prevent execution of unauthorized or revoked images, including unauthorized rollback to an older but validly signed software version. The trust anchor and initial verification function should be immutable after provisioning or protected by hardware-enforced mechanisms that prevent unauthorized modification and preserve their integrity. Components implementing the RoT must also be qualified for the expected mission radiation environment. Radiation tolerance addresses the reliability of the trust anchor, while immutability or protected update mechanisms address its resistance to unauthorized modification. This is particularly critical where radiation-induced bit flips and the physical inaccessibility of on-orbit hardware make a tamper-resistant, immutable hardware anchor essential to sustained boot integrity across the mission lifetime.	Functional	Intersects With	Protection of Boot Firmware	END-11.5	Automated mechanisms exist to protect the integrity of boot firmware in systems.	5		END-06.6
CM0060	Secret Shares	Secret sharing for side-channel protection, commonly implemented as masking, represents each sensitive value or intermediate computation using multiple randomized shares. A masking scheme of order dtypically uses d+1 shares and is designed so that observation of up to dshares or covered intermediate values does not reveal information about the underlying sensitive value within the defined security model. The complete set of shares reconstructs the original value and therefore must not be considered mutually independent. Computation must be performed on the shares using masking operations or gadgets designed to preserve the required security order. Masking can prevent straightforward first-order exploitation and increase the observations or attack complexity required to recover the secret, but joint leakage from multiple shares, unintended recombination, register transitions, glitches, memory activity, or other implementation effects may remain exploitable. Higher-order attacks specifically attempt to combine leakage associated with multiple shares and are not eliminated merely by dividing a value into shares. The primary operational tradeoff is a significant increase in computational operations, approaching a doubling of the number of operations required, which translates directly into increased power consumption; this overhead must be evaluated carefully against the spacecraft's power budget and the processing constraints of the target hardware. Masking should be applied to the complete set of operations, intermediate values, key schedules, conversions, memory transfers, and control paths whose leakage could disclose the protected secret. Selective masking may be used when analysis demonstrates that unmasked operations and transitions do not expose secret-dependent information and that the resulting implementation remains secure within the approved leakage	Functional	No Relationship	N/A	N/A	N/A	0		
CM0042	Robust Fault Management	The fault management system is a high-privilege, autonomous spacecraft function that adversaries may attempt to exploit as an attack vector, triggering protective responses that place the spacecraft in a degraded or more vulnerable operational state. Attack scenarios include manipulating sensor, state, or telemetry information to induce onboard or ground-directed safing actions; creating false fault conditions through sensor spoofing or proximity operations; exploiting safe-mode configurations that reduce security protections; and inducing autonomous maneuver responses through crafted fault indications. Robust fault management requires that safing procedures and autonomous responses be designed with explicit security analysis confirming that each protective action does not introduce a more exploitable system state than the fault condition it responds to. The integrity and authenticity of sensor data, state information, commands, and telemetry used by onboard or ground-based fault management functions must be protected to prevent falsified inputs from triggering unintended responses. Every fault response, including mode transitions, actuator commands, and communication reconfigurations, must be evaluated against the question of whether an adversary could deliberately induce that response and whether the resulting system state provides the adversary with meaningful advantage.	Functional	No Relationship	N/A	N/A	N/A	0		
CM0048	Resilient Position, Navigation, and Timing	Where compatible authentication services are available, GNSS receivers used for spacecraft position, navigation, and timing (PNT) must authenticate the navigation information and its asserted GNSS system source before treating that information as trusted. Navigation-message authentication must not be treated as complete protection against spoofing because it may not authenticate the ranging signal or prevent all replay, meaconing, or signal-manipulation scenarios. Authenticated GNSS information should therefore be combined with PNT integrity monitoring and alternate navigation or timing sources appropriate to mission risk. The spacecraft must maintain a fault-tolerant authoritative time architecture capable of maintaining time within mission-defined accuracy and uncertainty limits when the primary source is degraded, rejected, or unavailable. The architecture should support the time-dependent cryptographic controls, command sequencing, telemetry correlation, fault-management logic, and other functions that rely on synchronized time. Each onboard processor must synchronize its internal clock to the authoritative time source whenever the measured time difference exceeds a threshold defined in the flight software (FSW), preventing clock drift from accumulating to levels that corrupt time-dependent functions. Where SpaceWire is used to distribute time, the spacecraft must implement the mission-defined synchronization protocol and achieve the accuracy required by the functions that consume that time. An accuracy of approximately one microsecond should be applied where required by the mission architecture and verified for the applicable SpaceWire nodes and operational configurations.	Functional	No Relationship	N/A	N/A	N/A	0		
CM0033	Relay Protection	Replay attacks capture and later retransmit previously valid authentication messages, frames, credentials, or tokens. Relay attacks forward an authentication exchange in real time between legitimate endpoints, causing an endpoint to authenticate a connection or action that the adversary is relaying without requiring compromise of the underlying cryptographic keys. Relay- and replay-resistant authentication mechanisms must be implemented when establishing remote connections or security associations with the spacecraft and for authenticated communications on spacecraft internal buses. The protections must prevent previously accepted authentication material or authenticated traffic from being reused outside its authorized context. Replay resistance should use freshness and anti-reuse mechanisms such as nonces, sequence numbers, timestamps where operationally suitable, and managed anti-replay windows. Relay resistance additionally requires authentication to be cryptographically bound to the intended endpoints, security association, session or channel context, and authenticated action. Challenge-response authentication provides replay resistance when fresh challenges are used but does not, by itself, prevent an adversary from relaying the challenge and response between legitimate endpoints. These protections must be applied at both external interface boundaries, including ground-to-spacecraft command links and crosslinks, and internal spacecraft bus connections where component-to-component	Functional	Intersects With	Network Security Controls (NSC)	NET-02	Mechanisms exist to develop, govern & update procedures to facilitate the implementation of Network Security Controls (NSC).	5		NET-01

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0068	Reinforcement Learning	A reinforcement learning (RL) agent deployed within the spacecraft or ground system can provide an adaptive, autonomous anomaly detection and response capability that identifies anomalous events, including malicious data inputs and injected commands, and redirects affected processes to proceed safely by ignoring or isolating the malicious input. An RL agent learns a response policy that maps observations to actions according to its training environment and reward function. It may generalize to scenarios not explicitly included in training, but its ability to detect or respond correctly to novel attacks or conditions outside the validated operational envelope must not be assumed. Anomaly detection may be incorporated into the RL architecture or provided by a separate monitoring function. Effective deployment requires separate protections against compromise of the training process and manipulation of observations presented to the deployed agent. Online learning or policy adaptation should be disabled unless specifically authorized, bounded, and validated. Agent-selected responses must be constrained by a trusted safety	Functional	No Relationship	N/A	N/A	N/A	0		
CM0072	Protocol Update / Refactoring	Communication and data exchange protocols governing spacecraft, ground system, and inter-system interfaces may contain specification-level vulnerabilities that cannot be resolved through implementation hardening alone, requiring updates to or refactoring of the protocol itself to eliminate the underlying weakness. Protocol vulnerabilities may arise from design flaws in the original specification, from advances in adversarial capability that render previously adequate security assumptions insufficient, or from emerging threats such as quantum computing that threaten the cryptographic primitives upon which protocol security depends. Protocol update or refactoring encompasses the deliberate modification of the rules, formats, and procedures governing system communications to address known vulnerabilities, improve security properties, or maintain adequate protection against the evolving threat environment over the mission's operational lifetime. Because space system protocols are often tightly coupled to hardware interfaces, flight software implementations, and ground system processing pipelines, protocol changes carry significant integration risk and must be managed through rigorous engineering change processes; this complexity makes proactive protocol security assessment and planned update capability more cost-effective than reactive refactoring under operational urgency.	Functional	No Relationship	N/A	N/A	N/A	0		
CM0001	Protect Sensitive Information	Space mission sensitive information spans a broad attack surface and must be inventoried, classified, and protected at a level commensurate with its sensitivity across every location where it resides, including ground systems, contractor networks, and remote access environments. Sensitive material typically includes functional and performance specifications, interface control documents (ICDs), command and telemetry (C&T) databases, uplink protection schemes including disable and bypass features, fault management logic, scripts, simulation and rehearsal results, failure and anomaly resolution records, and architecture and software documentation. Each information type must be assigned a protection level, such as unclassified, controlled, proprietary, or classified, and access must be restricted to personnel with defined roles and a verified need to know. Sensitive data shall be protected at rest and in transit using encryption or other mission-approved safeguards commensurate with its classification, sensitivity, threat exposure, and operational constraints. DLP capabilities shall be applied to systems and data flows where they are technically feasible and effective, with alternative access controls, monitoring, or information-flow protections used where conventional DLP technology is not suitable. Ongoing configuration management must track, control, and document all changes to command procedures and critical database content to prevent unauthorized modification and mission degradation.	Functional	Intersects With	Asset Scope Classification	AST-16	Mechanisms exist to determine security, compliance and resilience control applicability by identifying, assigning and documenting the appropriate asset scope categorization for all Technology Assets, Applications and/or Services (TAAS) and personnel (internal and third-parties).	5		AST-04.1
CM0001	Protect Sensitive Information	Space mission sensitive information spans a broad attack surface and must be inventoried, classified, and protected at a level commensurate with its sensitivity across every location where it resides, including ground systems, contractor networks, and remote access environments. Sensitive material typically includes functional and performance specifications, interface control documents (ICDs), command and telemetry (C&T) databases, uplink protection schemes including disable and bypass features, fault management logic, scripts, simulation and rehearsal results, failure and anomaly resolution records, and architecture and software documentation. Each information type must be assigned a protection level, such as unclassified, controlled, proprietary, or classified, and access must be restricted to personnel with defined roles and a verified need to know. Sensitive data shall be protected at rest and in transit using encryption or other mission-approved safeguards commensurate with its classification, sensitivity, threat exposure, and operational constraints. DLP capabilities shall be applied to systems and data flows where they are technically feasible and effective, with alternative access controls, monitoring, or information-flow protections used where conventional DLP technology is not suitable. Ongoing configuration management must track, control, and document all changes to command procedures and critical database content to prevent unauthorized modification and mission degradation.	Functional	Intersects With	Data Protection	DCH-02	Mechanisms exist to facilitate the implementation of data protection controls.	5		DCH-01
CM0001	Protect Sensitive Information	Space mission sensitive information spans a broad attack surface and must be inventoried, classified, and protected at a level commensurate with its sensitivity across every location where it resides, including ground systems, contractor networks, and remote access environments. Sensitive material typically includes functional and performance specifications, interface control documents (ICDs), command and telemetry (C&T) databases, uplink protection schemes including disable and bypass features, fault management logic, scripts, simulation and rehearsal results, failure and anomaly resolution records, and architecture and software documentation. Each information type must be assigned a protection level, such as unclassified, controlled, proprietary, or classified, and access must be restricted to personnel with defined roles and a verified need to know. Sensitive data shall be protected at rest and in transit using encryption or other mission-approved safeguards commensurate with its classification, sensitivity, threat exposure, and operational constraints. DLP capabilities shall be applied to systems and data flows where they are technically feasible and effective, with alternative access controls, monitoring, or information-flow protections used where conventional DLP technology is not suitable. Ongoing configuration management must track, control, and document all changes to command procedures and critical database content to prevent unauthorized modification and mission degradation.	Functional	Intersects With	Data & Asset Classification	DCH-04	Mechanisms exist to ensure data and assets are categorized in accordance with applicable statutory, regulatory and contractual requirements.	5		DCH-02

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0001	Protect Sensitive Information	Space mission sensitive information spans a broad attack surface and must be inventoried, classified, and protected at a level commensurate with its sensitivity across every location where it resides, including ground systems, contractor networks, and remote access environments. Sensitive material typically includes functional and performance specifications, interface control documents (ICDs), command and telemetry (C&T) databases, uplink protection schemes including disable and bypass features, fault management logic, scripts, simulation and rehearsal results, failure and anomaly resolution records, and architecture and software documentation. Each information type must be assigned a protection level, such as unclassified, controlled, proprietary, or classified, and access must be restricted to personnel with defined roles and a verified need to know. Sensitive data shall be protected at rest and in transit using encryption or other mission-approved safeguards commensurate with its classification, sensitivity, threat exposure, and operational constraints. DLP capabilities shall be applied to systems and data flows where they are technically feasible and effective, with alternative access controls, monitoring, or information-flow protections used where conventional DLP technology is not suitable. Ongoing configuration management must track, control, and document all changes to command procedures and critical database content to prevent unauthorized modification and mission degradation.	Functional	Intersects With	Sensitive / Regulated Data Protection	DCH-07	Mechanisms exist to protect sensitive and/or regulated data wherever it is processed and/or stored.	5		DCH-01.2
CM0035	Protect Authenticators	Authenticators and associated authenticator material, including passwords, secret and private keys, tokens, biometric templates, shared secrets, certificates, and trust-store entries, must be protected against unauthorized modification and, where the material is confidential, unauthorized disclosure throughout their lifecycle. Disclosure of secret authenticator material may enable adversaries to impersonate legitimate users or systems. Unauthorized modification of authenticators, certificates, or trust information can deny access to legitimate entities, substitute adversary-controlled credentials, or corrupt the trust basis of mission authentication mechanisms. Protection must apply to authenticators at rest, in transit, and in use, and must extend to all forms and storage locations, including credential databases, configuration files, embedded device credentials, hardware security tokens, and cryptographic key stores. Authenticator protection is a prerequisite for the effectiveness of any authentication-based access control; an authentication system whose authenticators are unprotected provides no meaningful security regardless of the strength of the underlying authentication protocol.	Functional	Intersects With	Protection of Authenticators	IAC-14.2	Mechanisms exist to protect authenticators commensurate with the sensitivity of the information to which use of the authenticator permits access.	5		IAC-10.5
CM0075	Proliferated Constellations	Proliferated satellite constellations increase mission resilience by deploying a larger number of functionally equivalent satellites in similar orbits, expanding overall constellation capacity and raising the number of assets an adversary must successfully attack to achieve meaningful mission degradation. Unlike distribution, in which multiple satellites or payloads work together to provide a complete capability, proliferation increases the number of systems performing the same or substantially equivalent mission. Its resilience benefit is primarily derived from additional capacity and reduced dependence on any individual satellite rather than from architectural diversity. Proliferation also supports resilience through on-orbit spare maintenance, in which additional satellites are held in reserve or parked in accessible orbits to replace operational assets without requiring new launches. The cost implications of proliferation are significant and architecture-dependent. Designs optimized for repeatable production may achieve lower unit costs through learning and economies of scale, but those savings depend on design stability, production quantity, supplier capacity, and the amount of non-recurring change between production lots. The choice to proliferate must be made as a mission architecture decision early in the program, as it determines the spacecraft design philosophy, production strategy, launch architecture, and ground system scalability requirements.	Functional	No Relationship	N/A	N/A	N/A	0		
CM0069	Process White Listing	Process whitelisting establishes an approved set of executable images, tasks, applications, or process types that may run on the spacecraft computing platform. Enforcement should occur within a trusted layer capable of controlling process or task creation, which may be implemented in firmware, a secure monitor, a hypervisor, the operating system, or the real-time executive according to the platform architecture. The enforcement mechanism must verify an approved process identity before execution and deny unapproved process creation unless a mission-approved recovery mechanism applies. This countermeasure limits the creation or execution of software identities not present in the approved baseline. It does not prevent an adversary from abusing an approved process, altering its runtime memory, redirecting its control flow, or exploiting permitted scripting or loading functionality unless those behaviors are addressed by complementary integrity and execution controls.	Functional	Intersects With	Explicitly Allow / Deny Applications	CFG-14.1	Mechanisms exist to explicitly allow (allowlist / whitelist) and/or block (denylist / blacklist) applications that are authorized to execute on systems.	5		CFG-03.3
CM0058	Power Randomization	Power randomization is a hardware-level countermeasure against power analysis side-channel attacks, in which an adversary monitors a device's power consumption during cryptographic or other security-sensitive operations to extract secret information such as cryptographic keys by correlating power traces with internal computational states. The technique uses an on-chip hardware mechanism to add data-independent or randomized power activity intended to reduce the observable signal-to-noise ratio between measured power consumption and security-sensitive internal computations. Power randomization increases the number or sophistication of measurements required for power analysis but does not eliminate the underlying leakage or guarantee resistance against averaging, profiling, multi-trace, or higher-order analysis. Power randomization must be incorporated into the chip architecture or selected as an existing capability of the target device. Its implementation can increase dynamic power consumption, die area, thermal load, design and verification complexity, non-recurring engineering cost, and potentially unit fabrication cost. Resulting spacecraft-level mass or volume impacts depend on packaging, power-delivery, and thermal-management consequences. These tradeoffs must be evaluated during the system design phase against the mission's threat model and the availability of alternative or complementary side-channel countermeasures, with power	Functional	No Relationship	N/A	N/A	N/A	0		

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)	
Reference Document:	Secure Controls Framework (SCF) version 2026.3
STRM Guidance:	https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document:	SPARTA Countermeasure (4.0)
Focal Document URL:	https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL:	https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0061	Power Masking	Power masking is a side-channel countermeasure in which secret-dependent values and intermediate computations are represented using multiple randomized shares. A masking scheme of order d is designed so that observations involving up to dcovered intermediate values do not reveal information about the protected secret under the scheme's defined leakage and adversary model. Correctly implemented masking can prevent straightforward lower-order exploitation and increase the complexity or number of observations required for successful power or electromagnetic analysis. It does not guarantee protection regardless of the number of measurements: higher-order, profiled, multivariate, or implementation-specific attacks may combine leakage from multiple shares or observations and recover the protected secret. The masking scheme generates randomized shares and performs the protected computation using masking operations designed to preserve the required security order. Reconstruction or conversion to an unmasked representation must occur only at an explicitly authorized boundary and must not expose secret-dependent values through registers, memory, buses, transitions, glitches, control flow, or other observable implementation state. Power masking applies secret-sharing principles to the secret key, cryptographic state, and other secret-dependent intermediate values throughout a computation. CM0060 may describe the general share-based protection concept, while CM0061 should focus on implementing and preserving that sharing across cryptographic operations to reduce exploitable power and electromagnetic leakage. Effective masking requires correct implementation across the entire cryptographic execution path, as a single unmasked intermediate value anywhere in the computation can restore exploitable correlation and	Functional	No Relationship	N/A	N/A	N/A	0		
CM0059	Power Consumption Obfuscation	Power consumption obfuscation encompasses hardware circuit design techniques and architectural obfuscation strategies that mask the relationship between a device's internal operations and its observable power consumption profile, increasing the difficulty and cost of power analysis side-channel attacks. Unlike power randomization, which adds noise to the power signal, obfuscation techniques actively obscure the underlying power consumption pattern through circuit-level design approaches such as dual-rail logic, balanced circuit topologies, and constant-power execution paths that decouple observable power draw from data-dependent computational activity. These techniques increase manufacturing cost and design complexity for sensor nodes and other embedded hardware, representing a one-time investment at fabrication that must be justified against the mission's physical threat environment and the sensitivity of the data processed by the protected device. Power consumption obfuscation is most effectively applied to hardware executing cryptographic operations or other security-sensitive functions where correlation between power traces and internal state would be most damaging if	Functional	No Relationship	N/A	N/A	N/A	0		
CM0084	Physical Seizure	Physical seizure capability employs spacecraft equipped with docking, manipulation, or proximity maneuvering systems to counter space-based threats and mitigate post-attack effects through direct physical interaction with other on-orbit objects. Primary applications include seizing or neutralizing a threatening satellite actively attacking or endangering other spacecraft, capturing a satellite that has been disabled or hijacked and is being operated for hostile purposes, and collecting and disposing of harmful orbital debris resulting from a kinetic attack. The effectiveness of a physical seizure system is fundamentally constrained by propellant and time: a seizure asset stored in a particular orbital regime cannot efficiently reach objects in significantly different orbits due to the delta-v required for large orbital plane changes or altitude transfers, making geostationary Earth orbit (GEO) assets poorly positioned to respond to threats in low Earth orbit (LEO) and vice versa. This constraint drives a basing trade between pre-positioned on-orbit assets and ground-based responsive-launch assets. On-orbit assets may provide shorter response times but remain limited by their current orbit, propellant reserves, and readiness state. Ground-based assets may be launched closer to the required orbital plane and altitude but remain constrained by launch readiness, vehicle performance, launch-site geometry, and the time required to reach and rendezvous with the target.	Functional	No Relationship	N/A	N/A	N/A	0		
CM0053	Physical Security Controls	Physical security controls form the outermost defensive layer protecting systems that can command the spacecraft, limiting access to commanding infrastructure to personnel who are both identity-verified and specifically authorized to be in those environments. Unauthorized physical access may enable theft, tampering, connection to exposed interfaces, use of unattended sessions, or attempts to extract or alter protected information. Physical security must therefore complement authentication, encryption, session management, tamper protection, and other technical controls rather than assume those controls will remain sufficient without protection of the underlying equipment. Physical security measures for commanding facilities and infrastructure must use layered controls selected according to the facility threat environment, system criticality, site characteristics, and consequence of unauthorized access. Controls may include perimeter barriers, controlled entry points, interior protected areas, identity and authorization verification, locks, intrusion detection, surveillance, and security personnel. Access to the facility must not, by itself, authorize physical access to commanding systems. Physical security controls must be commensurate with the sensitivity and criticality of the commanding functions they protect, with the most sensitive commanding capabilities requiring the most stringent physical access restrictions.	Functional	Intersects With	Physical & Environmental Protections	PES-02	Mechanisms exist to facilitate the operation of physical and environmental protection controls.	5		PES-01
CM0065	OSAM Dual Authorization	Before a cooperative OSAM engagement enters a protected proximity, capture, docking, mating, or servicing phase, the servicing spacecraft must be authenticated and the specific engagement authorized by both the serviced asset's mission control authority and, where technically capable, the serviced asset itself. Authentication establishes the identity of the servicer, while dual authorization requires two independent approval decisions for the proposed activity. The approvals may be sequential rather than simultaneous but must both remain valid and be bound to the same servicer, client, service scope, mission phase, interfaces, operational constraints, and validity period. Failure to obtain or maintain authorization must cause the serviced asset to withhold cooperation and interface enablement and must invoke the approved hold, retreat, or abort response. These controls reduce the risk of unauthorized servicing but cannot physically prevent a hostile or non-cooperative spacecraft from approaching the asset.	Functional	No Relationship	N/A	N/A	N/A	0		

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0026	Original Component Manufacturer	Hardware components that cannot be sourced directly from the original component manufacturer (OCM) or an authorized franchised distributor, and software that cannot be obtained from the original publisher, developer, or an authorized distribution channel, represent elevated supply-chain risk and must not be procured or incorporated into the mission system without documented approval from the program's supply-chain governance authority. Sourcing hardware from the OCM or authorized franchised distributors provides greater assurance of component authenticity, traceability, and conformance to specification. Obtaining software from the original publisher, developer, or an authorized distribution channel similarly reduces the risk of unauthorized, altered, fraudulent, or malicious software. Deviations from these approved sourcing channels introduce additional supply-chain risk that must be assessed before acceptance. The approval process for non-OCM-sourced items must evaluate the specific risk posed by the alternative source, the criticality of the component or software to mission function, the availability and adequacy of compensating inspection and authentication measures, and whether a compliant source can be identified before accepting the deviation. This governance requirement applies to hardware components, firmware, and software, although the applicable sourcing and authentication methods differ. Hardware controls should address component authenticity and traceability, while software controls should address publisher or developer provenance, distribution-channel integrity, license legitimacy, and cryptographic verification	Functional	Intersects With	Provenance	AST-14	Mechanisms exist to track the origin, development, ownership, location and changes to Technology Assets, Applications, Services and/or Data (TAASD).	5		AST-03.2
CM0026	Original Component Manufacturer	Hardware components that cannot be sourced directly from the original component manufacturer (OCM) or an authorized franchised distributor, and software that cannot be obtained from the original publisher, developer, or an authorized distribution channel, represent elevated supply-chain risk and must not be procured or incorporated into the mission system without documented approval from the program's supply-chain governance authority. Sourcing hardware from the OCM or authorized franchised distributors provides greater assurance of component authenticity, traceability, and conformance to specification. Obtaining software from the original publisher, developer, or an authorized distribution channel similarly reduces the risk of unauthorized, altered, fraudulent, or malicious software. Deviations from these approved sourcing channels introduce additional supply-chain risk that must be assessed before acceptance. The approval process for non-OCM-sourced items must evaluate the specific risk posed by the alternative source, the criticality of the component or software to mission function, the availability and adequacy of compensating inspection and authentication measures, and whether a compliant source can be identified before accepting the deviation. This governance requirement applies to hardware components, firmware, and software, although the applicable sourcing and authentication methods differ. Hardware controls should address component authenticity and traceability, while software controls should address publisher or developer provenance, distribution-channel integrity, license legitimacy, and cryptographic verification	Functional	Intersects With	Supply Chain Risk Management (SCRM) Plan	RSK-20	Mechanisms exist to develop a plan for Supply Chain Risk Management (SCRM) associated with the development, acquisition, maintenance and disposal of Technology Assets, Applications and/or Services (TAAS), including documenting selected mitigating actions and monitoring performance against those plans.	5		RSK-09
CM0026	Original Component Manufacturer	Hardware components that cannot be sourced directly from the original component manufacturer (OCM) or an authorized franchised distributor, and software that cannot be obtained from the original publisher, developer, or an authorized distribution channel, represent elevated supply-chain risk and must not be procured or incorporated into the mission system without documented approval from the program's supply-chain governance authority. Sourcing hardware from the OCM or authorized franchised distributors provides greater assurance of component authenticity, traceability, and conformance to specification. Obtaining software from the original publisher, developer, or an authorized distribution channel similarly reduces the risk of unauthorized, altered, fraudulent, or malicious software. Deviations from these approved sourcing channels introduce additional supply-chain risk that must be assessed before acceptance. The approval process for non-OCM-sourced items must evaluate the specific risk posed by the alternative source, the criticality of the component or software to mission function, the availability and adequacy of compensating inspection and authentication measures, and whether a compliant source can be identified before accepting the deviation. This governance requirement applies to hardware components, firmware, and software, although the applicable sourcing and authentication methods differ. Hardware controls should address component authenticity and traceability, while software controls should address publisher or developer provenance, distribution-channel integrity, license legitimacy, and cryptographic verification	Functional	Intersects With	Supply Chain Risk Management (SCRM)	TPM-04	Mechanisms exist to: (1) Evaluate security risks and threats associated with Technology Assets, Applications and/or Services (TAAS) supply chains; and (2) Take appropriate remediation actions to minimize the organization's exposure to those risks and threats, as necessary.	5		TPM-03
CM0088	Organizational Policy	Documented cybersecurity policies establish the foundational governance framework that defines how an organization protects its information assets, assigns security responsibilities, and ensures consistent security behavior across all personnel and organizational levels. For space mission organizations, these policies must address the unique threat environment, operational constraints, and asset types associated with spacecraft, ground systems, and mission data, providing a coherent governance layer that connects organizational security objectives to the technical controls and operational practices implemented throughout the mission lifecycle. Well-documented policies ensure that personnel at all levels, from executive leadership through program management to operations staff, understand their security roles and responsibilities, reducing the probability of security failures attributable to ambiguity, inconsistency, or lack of guidance. Policies establish organizational security objectives, authorities, responsibilities, and required outcomes. Risk assessments, system requirements, standards, plans, and procedures translate those policies into technical controls and operational practices. During a security incident, approved incident response plans and procedures provide the actionable guidance needed to implement organizational policy and support timely decision-making. Mission organizations must identify the legal, regulatory, contractual, policy, and licensing requirements applicable to their activities and ensure that their cybersecurity policies address those obligations. Documented policies may therefore serve both organizational governance and compliance	Functional	Subset Of	Publishing Security, Compliance & Resilience Documentation	GOV-04	Mechanisms exist to establish, maintain and disseminate policies, standards and procedures necessary for secure, compliant and resilient capabilities.	10		GOV-02

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0088	Organizational Policy	Documented cybersecurity policies establish the foundational governance framework that defines how an organization protects its information assets, assigns security responsibilities, and ensures consistent security behavior across all personnel and organizational levels. For space mission organizations, these policies must address the unique threat environment, operational constraints, and asset types associated with spacecraft, ground systems, and mission data, providing a coherent governance layer that connects organizational security objectives to the technical controls and operational practices implemented throughout the mission lifecycle. Well-documented policies ensure that personnel at all levels, from executive leadership through program management to operations staff, understand their security roles and responsibilities, reducing the probability of security failures attributable to ambiguity, inconsistency, or lack of guidance. Policies establish organizational security objectives, authorities, responsibilities, and required outcomes. Risk assessments, system requirements, standards, plans, and procedures translate those policies into technical controls and operational practices. During a security incident, approved incident response plans and procedures provide the actionable guidance needed to implement organizational policy and support timely decision-making. Mission organizations must identify the legal, regulatory, contractual, policy, and licensing requirements applicable to their activities and ensure that their cybersecurity policies address those obligations. Documented policies may therefore serve both organizational governance and compliance	Functional	Subset Of	Publishing Security, Compliance & Resilience Documentation	GOV-04	Mechanisms exist to establish, maintain and disseminate policies, standards and procedures necessary for secure, compliant and resilient capabilities.	10		GOV-02
CM0047	Operating System Security	The spacecraft operating system (OS) must be subjected to software assurance scrutiny commensurate with its mission criticality and security role. Prior assurance and flight-history evidence may be reused only after its applicability to the selected OS version, configuration, target hardware, operational use, and current mission security requirements has been evaluated. The OS must be analyzed for its attack surface, and all features, services, libraries, and interfaces not required for the mission's defined operational functions must be stripped, disabled, or removed before the OS is integrated into the flight software stack. This hardening requirement is particularly significant for some real-time operating systems (RTOS), which commonly include networking stacks, file systems, shell interfaces, diagnostic services, and other general-purpose capabilities that are unnecessary for spacecraft operations but expand the exploitable attack surface available to an adversary. Only an organization-approved OS product, version, build, and configuration baseline may be used in the delivered spacecraft system. The approved baseline must be enforced through reproducible build controls, configuration management, component inventories, and automated verification where supported.	Functional	Intersects With	Least Functionality	CFG-03	Mechanisms exist to configure Technology Assets, Applications and/or Services (TAAS) to provide only essential capabilities by specifically prohibiting or restricting the use of ports, protocols, and/or services.	5		CFG-03
CM0047	Operating System Security	The spacecraft operating system (OS) must be subjected to software assurance scrutiny commensurate with its mission criticality and security role. Prior assurance and flight-history evidence may be reused only after its applicability to the selected OS version, configuration, target hardware, operational use, and current mission security requirements has been evaluated. The OS must be analyzed for its attack surface, and all features, services, libraries, and interfaces not required for the mission's defined operational functions must be stripped, disabled, or removed before the OS is integrated into the flight software stack. This hardening requirement is particularly significant for some real-time operating systems (RTOS), which commonly include networking stacks, file systems, shell interfaces, diagnostic services, and other general-purpose capabilities that are unnecessary for spacecraft operations but expand the exploitable attack surface available to an adversary. Only an organization-approved OS product, version, build, and configuration baseline may be used in the delivered spacecraft system. The approved baseline must be enforced through reproducible build controls, configuration management, component inventories, and automated verification where supported.	Functional	Intersects With	Secure Baseline Configurations	CFG-04	Mechanisms exist to develop, document and maintain secure baseline configurations for Technology Assets, Applications and/or Services (TAAS) that are consistent with industry-accepted hardening standards.	5		CFG-02
CM0047	Operating System Security	The spacecraft operating system (OS) must be subjected to software assurance scrutiny commensurate with its mission criticality and security role. Prior assurance and flight-history evidence may be reused only after its applicability to the selected OS version, configuration, target hardware, operational use, and current mission security requirements has been evaluated. The OS must be analyzed for its attack surface, and all features, services, libraries, and interfaces not required for the mission's defined operational functions must be stripped, disabled, or removed before the OS is integrated into the flight software stack. This hardening requirement is particularly significant for some real-time operating systems (RTOS), which commonly include networking stacks, file systems, shell interfaces, diagnostic services, and other general-purpose capabilities that are unnecessary for spacecraft operations but expand the exploitable attack surface available to an adversary. Only an organization-approved OS product, version, build, and configuration baseline may be used in the delivered spacecraft system. The approved baseline must be enforced through reproducible build controls, configuration management, component inventories, and automated verification where supported.	Functional	Intersects With	Configure Technology Assets, Applications and/or Services (TAAS) for High-Risk Areas	CFG-05	Mechanisms exist to configure Technology Assets, Applications and/or Services (TAAS) utilized in high-risk areas with more restrictive secure baseline configurations.	5		CFG-02.5
CM0047	Operating System Security	The spacecraft operating system (OS) must be subjected to software assurance scrutiny commensurate with its mission criticality and security role. Prior assurance and flight-history evidence may be reused only after its applicability to the selected OS version, configuration, target hardware, operational use, and current mission security requirements has been evaluated. The OS must be analyzed for its attack surface, and all features, services, libraries, and interfaces not required for the mission's defined operational functions must be stripped, disabled, or removed before the OS is integrated into the flight software stack. This hardening requirement is particularly significant for some real-time operating systems (RTOS), which commonly include networking stacks, file systems, shell interfaces, diagnostic services, and other general-purpose capabilities that are unnecessary for spacecraft operations but expand the exploitable attack surface available to an adversary. Only an organization-approved OS product, version, build, and configuration baseline may be used in the delivered spacecraft system. The approved baseline must be enforced through reproducible build controls, configuration management, component inventories, and automated verification where supported.	Functional	Intersects With	Explicitly Allow / Deny Applications	CFG-14.1	Mechanisms exist to explicitly allow (allowlist / whitelist) and/or block (denylist / blacklist) applications that are authorized to execute on systems.	5		CFG-03.3

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0050	On-board Message Encryption	Authentication controls on the spacecraft internal bus verify the identity of communicating components but do not protect the confidentiality of the data in transit; an adversary with access to the bus, whether through a compromised component, a hardware implant, or a physical access event, can observe all unencrypted inter-component communications regardless of whether authentication is enforced. Encrypting data traversing the spacecraft internal bus protects the confidentiality of selected message content from entities that can observe the bus but do not possess authorization and the applicable cryptographic keys. The protection does not prevent disclosure to a compromised component that legitimately possesses the decryption key, and it may not conceal unencrypted protocol headers, addressing information, message timing, or traffic volume. Bus encryption should be considered for bus segments or message types carrying information whose unauthorized disclosure would create unacceptable mission, security, privacy, or operational risk. Criticality alone does not establish a confidentiality requirement. Where confidentiality is required, encryption must be combined with message integrity, source authentication, and replay protection through an approved authenticated-encryption mechanism or an appropriately	Functional	Intersects With	Use of Cryptographic Controls	CRY-02	Mechanisms exist to facilitate the implementation of cryptographic protections controls using known public standards and trusted cryptographic technologies.	5		CRY-01
CM0032	On-board Intrusion Detection & Prevention	An on-board intrusion detection and prevention system (IDS/IPS) monitors mission-critical spacecraft components and systems, generates and stores audit records, and supports mission-approved responses to detected threats. Depending on the mission architecture, threat, and availability of ground support, responses may be autonomous, ground-directed, or a combination of both. The system should address both known attack patterns and previously unseen anomalous behavior through complementary signature-based and behavior- or anomaly-based detection methods. Machine learning or adaptive technologies may be used when their performance, resource consumption, and failure behavior have been validated for the mission environment. Detection and response coverage should address applicable adversary activities across the attack lifecycle, including initial access, execution, persistence, defense evasion, and exfiltration. The on-board IDS/IPS must be integrated with the spacecraft's traditional fault management system to provide a unified approach to anomaly response, ensuring that cyber-triggered responses are compatible with fault management logic and do not produce unintended effects or fratricide against the spacecraft's own systems; countermeasures that are incompatible with fault management are considered unsafe and must not be executed autonomously. The response hierarchy must prioritize vehicle safety and continued mission operations. Advanced containment or deception responses may be considered when they can be executed without unacceptable mission risk. The system should preserve evidence that supports post-event analysis, threat characterization, and potential attribution by authorized ground support.	Functional	Intersects With	Network Intrusion Detection & Prevention Systems (NIDS & NIPS) Monitoring	MON-23	Mechanisms exist to monitor Network Intrusion Detection / Prevention Systems (NIDS / NIPS) technologies on critical systems, key network segments and network choke points.	5		MON-01.1
CM0034	Monitor Critical Telemetry Points	Monitoring defined spacecraft telemetry points provides a key source of evidence for detecting adversary activity against on-orbit systems, where observability is largely limited to the events and conditions the spacecraft can sense, record, and report. Monitored telemetry must include both accepted and rejected commands, command mode transitions, command counters, and other indicators of commanding activity, enabling detection of unauthorized command attempts that fail authentication as well as anomalous patterns in legitimate command traffic. Monitoring scope should include RF and link-quality indicators that support detection and triage of interference or suspected jamming. These indicators should be correlated with expected link conditions and other available evidence before hostile activity is concluded. Security-relevant telemetry should be integrated and time-correlated with ground-based defensive cyber operations infrastructure, including security information and event management (SIEM) and audit platforms, to provide unified space-system cybersecurity situational awareness. The resulting view should correlate spacecraft observations with relevant ground-system security events while accounting for telemetry latency, contact availability, and	Functional	No Relationship	N/A	N/A	N/A	0		
CM0066	Model-based System Verification	Model-based system verification compares observed spacecraft behavior with behavior predicted by a physics-based or hybrid model to identify discrepancies inconsistent with mission-defined physical, configuration, and operational constraints. It can detect unexpected or physically implausible sensor values, state transitions, actuator effects, and command outcomes, but it does not independently establish that a command was authorized or that an anomaly was caused by a cyberattack. The verification architecture should use an independently protected model, diagnosis engine, configuration baseline, and, where feasible, diverse or separately validated input sources. A model driven solely by the same compromised sensor values, state estimates, command history, or software pathways as the monitored system may reproduce the adversary-controlled state rather than detect it. Model-based verification therefore complements authentication, command authorization, data integrity, fault management, and security monitoring rather than replacing them. The fidelity of the physics model determines the sensitivity and specificity of the verification, with higher-fidelity models capable of detecting subtler anomalies at the cost of greater computational resources. The model should provide sufficient fidelity for the defined verification objectives without introducing unnecessary complexity or sensitivity to poorly characterized parameters. Higher fidelity may improve detection of some anomalies but does not automatically improve sensitivity or specificity and may increase computational cost, model-maintenance burden, or false alerts caused by model mismatch.	Functional	No Relationship	N/A	N/A	N/A	0		
CM0079	Maneuverability	Spacecraft maneuverability provides an active physical defense capability against kinetic and certain directed energy threats by enabling the satellite to relocate from a predicted intercept trajectory when a threat is detected with sufficient warning time. Against unguided projectiles, maneuvering out of the predicted impact trajectory can be effective, requiring only sufficient delta-v and warning time to execute a displacement maneuver before impact. Against guided threats, including direct-ascent anti-satellite (ASAT) weapons and co-orbital ASAT platforms equipped with onboard sensors, maneuverability is significantly more constrained in its effectiveness; evasion requires displacing the satellite beyond the seeker or sensor acquisition range of the guided warhead, which demands larger delta-v margins and more precise threat characterization than unguided intercept scenarios. The effectiveness of maneuverability as a countermeasure is therefore strongly dependent on the warning time provided by space domain awareness (SDA) capabilities, the propulsion capacity of the spacecraft, the fidelity of threat trajectory characterization, and whether the threat employs passive or active terminal guidance. Maneuverability also provides operational flexibility for avoiding predictable orbital slots that adversaries may have targeted in advance, complicating targeting planning even in the	Functional	No Relationship	N/A	N/A	N/A	0		

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: SPARTA Countermeasure (4.0)
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0049	Machine Learning Data Integrity	When artificial intelligence (AI) or machine learning (ML) is employed for mission-critical spacecraft or ground system operations, the integrity of the training data set is a foundational security requirement, not merely a data quality concern. Data poisoning attacks introduce or modify training samples, labels, or other data-pipeline inputs to alter resulting model behavior. Poisoning may cause general performance degradation, targeted misclassification, or backdoor behavior that appears only when specific operational inputs or triggers are present. Remediation may require identifying and removing affected data, retraining or fine-tuning from a trusted dataset, or rolling back to a validated model. Because remediation can be difficult and may not identify every affected behavior, controls should prioritize preventing unauthorized data changes and detecting suspicious inputs before training while maintaining recoverable trusted dataset and model versions. Detection and prevention techniques include validity checking of data sets to identify statistically anomalous or out-of-distribution inputs, statistical analysis to detect injected samples that deviate from expected data distributions, manual analysis of flagged inputs, and regression testing over time to identify model behavioral drift that may indicate successful historical poisoning. These controls must be applied as part of a continuous data governance process spanning the full training pipeline, from data collection and curation through model validation and operational deployment.	Functional	Intersects With	Data Source Identification	AAT-18.1	Mechanisms exist to identify and document data sources utilized in the training and/or operation of Artificial Intelligence and Autonomous Technologies (AAT).	5		AAT-12.1
CM0049	Machine Learning Data Integrity	When artificial intelligence (AI) or machine learning (ML) is employed for mission-critical spacecraft or ground system operations, the integrity of the training data set is a foundational security requirement, not merely a data quality concern. Data poisoning attacks introduce or modify training samples, labels, or other data-pipeline inputs to alter resulting model behavior. Poisoning may cause general performance degradation, targeted misclassification, or backdoor behavior that appears only when specific operational inputs or triggers are present. Remediation may require identifying and removing affected data, retraining or fine-tuning from a trusted dataset, or rolling back to a validated model. Because remediation can be difficult and may not identify every affected behavior, controls should prioritize preventing unauthorized data changes and detecting suspicious inputs before training while maintaining recoverable trusted dataset and model versions. Detection and prevention techniques include validity checking of data sets to identify statistically anomalous or out-of-distribution inputs, statistical analysis to detect injected samples that deviate from expected data distributions, manual analysis of flagged inputs, and regression testing over time to identify model behavioral drift that may indicate successful historical poisoning. These controls must be applied as part of a continuous data governance process spanning the full training pipeline, from data collection and curation through model validation and operational deployment.	Functional	Intersects With	Data Source Integrity	AAT-18.2	Mechanisms exist to protect the integrity of source data to prevent accidental contamination or malicious corruption (e.g., data poisoning) that could compromise the performance of Artificial Intelligence and Autonomous Technologies (AAT).	8		AAT-12.2
CM0049	Machine Learning Data Integrity	When artificial intelligence (AI) or machine learning (ML) is employed for mission-critical spacecraft or ground system operations, the integrity of the training data set is a foundational security requirement, not merely a data quality concern. Data poisoning attacks introduce or modify training samples, labels, or other data-pipeline inputs to alter resulting model behavior. Poisoning may cause general performance degradation, targeted misclassification, or backdoor behavior that appears only when specific operational inputs or triggers are present. Remediation may require identifying and removing affected data, retraining or fine-tuning from a trusted dataset, or rolling back to a validated model. Because remediation can be difficult and may not identify every affected behavior, controls should prioritize preventing unauthorized data changes and detecting suspicious inputs before training while maintaining recoverable trusted dataset and model versions. Detection and prevention techniques include validity checking of data sets to identify statistically anomalous or out-of-distribution inputs, statistical analysis to detect injected samples that deviate from expected data distributions, manual analysis of flagged inputs, and regression testing over time to identify model behavioral drift that may indicate successful historical poisoning. These controls must be applied as part of a continuous data governance process spanning the full training pipeline, from data collection and curation through model validation and operational deployment.	Functional	Intersects With	Provenance	AST-14	Mechanisms exist to track the origin, development, ownership, location and changes to Technology Assets, Applications, Services and/or Data (TAASD).	5		AST-03.2
CM0046	Long Duration Testing	Long duration testing subjects spacecraft software, firmware, hardware, and relevant integrated ground interfaces, or representative simulation and emulation environments, to extended test execution of 30 days or more to expose security and reliability defects that may manifest only after prolonged operation or specific time-dependent conditions. Race conditions, memory or resource leaks, time-dependent state corruption, resource exhaustion, counter rollover, and time-triggered malicious behavior may not manifest during short-duration testing because their activation depends on accumulated runtime, rare timing interactions, or gradual changes in system state. Long duration testing increases the opportunity to expose these conditions before deployment and complements static analysis, formal analysis, stress testing, fault injection, and targeted rollover testing. Testing should use the highest-fidelity environment appropriate to the test objectives. Flight-representative hardware should be used where hardware timing, device behavior, or integration effects are material; validated simulation or emulation may be used for conditions that it represents with sufficient fidelity. Differences between the test environment and operational system must be documented and	Functional	No Relationship	N/A	N/A	N/A	0		
CM0039	Least Privilege	The principle of least privilege requires that every process, user account, service, and system component be granted only the permissions and access rights necessary to perform its defined function, with no additional privileges retained beyond what the assigned task requires. Applied to spacecraft and ground systems, this means that processes executing on flight computers, operating system services, ground system applications, and inter-system communication handlers are each confined to the minimum privilege level needed for their specific function, preventing a compromised or malfunctioning component from leveraging excess permissions to affect other system resources or functions. Separate execution domains should be used where supported to reinforce least privilege and contain process failures or compromise. Process isolation does not replace explicit access controls, because authorized communication and shared resources may still cross execution-domain boundaries. Least privilege is a foundational design principle that reduces the consequence of any individual component compromise by limiting what an adversary can accomplish within that	Functional	Intersects With	Least Privilege	IAC-30	Mechanisms exist to utilize the concept of least privilege, allowing only authorized access to processes necessary to accomplish assigned tasks in accordance with organizational business functions.	5		IAC-21

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0052	Insider Threat Protection	Insider threats represent a distinct and particularly difficult risk category for space mission security, as individuals with authorized access to mission systems, facilities, and commanding infrastructure can cause significant damage without needing to overcome the external access controls that defend against outside adversaries. An insider threat program establishes the organizational, procedural, and technical controls necessary to deter, detect, and respond to malicious or negligent actions by personnel with legitimate access, including attempts by insiders to masquerade as other authorized individuals to access commanding functions or sensitive mission infrastructure under a false identity. The program must address both the technical controls that limit what any individual can do with their authorized access and the procedural and behavioral controls that create accountability, reduce opportunity, and enable early detection of concerning patterns. Effective insider threat protection requires integration across personnel security, access management, monitoring, and incident response functions, treating the insider threat as an ongoing operational risk to be managed continuously rather than a problem solved by initial personnel vetting alone.	Functional	Intersects With	Insider Threats	MON-17.3	Mechanisms exist to monitor internal personnel activity for potential security incidents.	5		MON-16.1
CM0052	Insider Threat Protection	Insider threats represent a distinct and particularly difficult risk category for space mission security, as individuals with authorized access to mission systems, facilities, and commanding infrastructure can cause significant damage without needing to overcome the external access controls that defend against outside adversaries. An insider threat program establishes the organizational, procedural, and technical controls necessary to deter, detect, and respond to malicious or negligent actions by personnel with legitimate access, including attempts by insiders to masquerade as other authorized individuals to access commanding functions or sensitive mission infrastructure under a false identity. The program must address both the technical controls that limit what any individual can do with their authorized access and the procedural and behavioral controls that create accountability, reduce opportunity, and enable early detection of concerning patterns. Effective insider threat protection requires integration across personnel security, access management, monitoring, and incident response functions, treating the insider threat as an ongoing operational risk to be managed continuously rather than a problem solved by initial personnel vetting alone.	Functional	Intersects With	Insider Threat Program	THR-15	Mechanisms exist to implement an insider threat program that includes a cross-discipline insider threat incident handling team.	5		THR-04
CM0052	Insider Threat Protection	Insider threats represent a distinct and particularly difficult risk category for space mission security, as individuals with authorized access to mission systems, facilities, and commanding infrastructure can cause significant damage without needing to overcome the external access controls that defend against outside adversaries. An insider threat program establishes the organizational, procedural, and technical controls necessary to deter, detect, and respond to malicious or negligent actions by personnel with legitimate access, including attempts by insiders to masquerade as other authorized individuals to access commanding functions or sensitive mission infrastructure under a false identity. The program must address both the technical controls that limit what any individual can do with their authorized access and the procedural and behavioral controls that create accountability, reduce opportunity, and enable early detection of concerning patterns. Effective insider threat protection requires integration across personnel security, access management, monitoring, and incident response functions, treating the insider threat as an ongoing operational risk to be managed continuously rather than a problem solved by initial personnel vetting alone.	Functional	Intersects With	Insider Threat Awareness	THR-16	Mechanisms exist to utilize security awareness training on recognizing and reporting potential indicators of insider threat.	5		THR-05
CM0063	Increase Clock Cycles/Timing	Timing side-channel attacks exploit observable differences in execution time to infer information about secret values, such as cryptographic keys, by correlating measured execution durations with data-dependent branching paths or memory access patterns. This countermeasure reduces timing leakage by ensuring that execution time, control flow, instruction selection, and memory-access behavior do not vary as a function of secret values within the defined implementation and threat model. Additional computation or delay may be used where appropriate, but constant-time behavior should primarily be achieved by eliminating secret-dependent branches, memory accesses, and variable-latency operations. Memory accesses involving secret-dependent values or indices shall be implemented so that observable access patterns and timing do not vary as a function of the protected secret within the defined threat model. Non-secret-dependent memory accesses need not be normalized solely for this countermeasure. Where mission timing requirements permit, access time normalization can be achieved by adding deliberate delays to faster accesses to equalize timing across all operations. Constant-time implementation may increase execution time, code size, memory use, power consumption, or design complexity, depending on the algorithm, implementation technique, processor, and memory architecture. These impacts must be measured for the target platform and evaluated against timing, power, thermal, and throughput mission requirements.	Functional	No Relationship	N/A	N/A	N/A	0		
CM0005	Ground-based Countermeasures	Ground-based countermeasures protect the terrestrial capabilities that develop, launch, command, monitor, operate, secure, and sustain space missions. These capabilities may be distributed across mission-owned systems, contractor environments, external partners, and commercial service providers. Because ground segment architectures and responsibilities vary, cybersecurity protections should be selected through threat-informed analysis of the mission functions being performed rather than through indiscriminate application of a single control set. The SPARTA Ground Segment Cyber Defenses guidance provides an interactive functional decomposition that maps ground segment function groups to applicable Defense-in-Depth sub-layers and countermeasure targets. The supporting report, VTR-2026-00702 Rev A, Ground Segment Cyber Defenses and Risk-Based Tiering, provides the methodology, definitions, and risk-tier rationale. The accompanying Ground Segment Cyber Defenses Excel Workbook consolidates the function mappings and Baseline and Enhanced countermeasure guidance into a resource that can be tailored to a specific mission architecture.	Functional	Intersects With	Security, Compliance & Resilience Program (SCRP)	GOV-02	Mechanisms exist to facilitate the design, implementation, and monitoring of security, compliance and resilience governance controls.	5		GOV-01

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0086	Filtering and Shuttering	Optical filters and shutters are passive and active protective mechanisms for remote sensing spacecraft sensors against laser dazzling and blinding attacks that exploit the same optical pathways used for legitimate mission collection. Optical filters selectively attenuate light outside the sensor's design wavelength bands, blocking laser energy at wavelengths that fall outside the mission's collection bands while preserving sensitivity to legitimate return signals; however, filters provide no protection against lasers operating at the same wavelengths the sensor is designed to detect, as filtering those wavelengths would simultaneously block the sensor from its intended mission function. Shutters provide complementary broadband protection by blocking or diverting light from the protected detector path when an exposure threshold or other anomaly criterion is met. Protection depends on the threat-detection latency, decision time, shutter actuation time, optical leakage, and duration of the incident illumination. A shutter may limit exposure from sustained or repeated illumination, but it must not be assumed to prevent damage from a short pulse that deposits damaging energy before closure. Together, filters and shutters provide layered protection: filters continuously attenuate selected wavelengths, while shutters can limit additional exposure from threats that cannot be sufficiently rejected by filtering. The fundamental tradeoff of shuttering is that it trades temporary collection interruption for sensor preservation, making the shutter activation threshold a critical design parameter that must balance sensor protection against mission data continuity requirements. Sustainment activities are limited because the filter and shutter hardware is generally fixed after launch, but operational monitoring, calibration assessment, mechanism exercising, threshold configuration control, and post-event analysis remain applicable.	Functional	No Relationship	N/A	N/A	N/A	0		
CM0051	Fault Injection Redundancy	Fault injection attacks deliberately induce errors in executing hardware or software, typically by manipulating voltage, clock signals, electromagnetic fields, or radiation, to corrupt computation and extract sensitive information or bypass security controls through observable fault effects. For mission-critical functions that must be protected against fault-based side-channel attacks, particularly cryptographic operations, redundancy-based countermeasures provide a robust detection and mitigation mechanism. The approach executes a protected function through two or more spatially or temporally redundant computations and compares their results. A mismatch indicates a computation fault or implementation discrepancy but does not by itself establish that fault injection occurred. With two results, the system can generally detect disagreement but cannot determine which result is valid. A voter or selection mechanism may provide fault masking only when the number of replicas, independence assumptions, and defined fault model support reliable adjudication; otherwise, the result must be rejected and a mission-approved protective response initiated. Although spacecraft sensor nodes and embedded processors operate under constrained resources, the functions most critical to protect through redundancy are typically cryptographic routines, whose execution footprint is bounded and whose compromise would have disproportionate mission security	Functional	No Relationship	N/A	N/A	N/A	0		
CM0045	Error Detection and Correcting Memory	Error detection and correcting (EDAC) memory provides a foundational defense against radiation-induced corruption in spacecraft onboard memory. The selected EDAC scheme must detect and correct error patterns within its designed capability and identify errors that exceed that capability. Common single-error-correction, double-error-detection schemes correct single-bit errors and detect double-bit errors, but other schemes may provide different correction and detection capabilities. The EDAC scheme must be integrated with both the fault management system and the spacecraft's cyber-protection mechanisms, enabling coordinated responses to uncorrectable multi-bit errors that go beyond time-delayed ground monitoring of EDAC telemetry. This integration is security-relevant because multi-bit memory errors, whether radiation-induced or adversarially induced through deliberate fault injection, can corrupt flight software, configuration data, or security-critical parameters in ways that create exploitable system states if not detected and responded to promptly and autonomously. The spacecraft must use the selected EDAC architecture to detect and correct errors during memory access and, where applicable, perform periodic memory scrubbing to detect and remove latent correctable errors. The implementation must identify the affected memory address or region for detected uncorrectable errors involving two or more bits, to the extent supported by the memory architecture, with higher-order detection or correction provided where required by mission risk. Detection of an uncorrectable error must trigger a timely onboard fault-management or cyber-protection response that prevents continued use or propagation of suspect data and autonomously minimizes adverse effects without waiting for ground detection. Subsequent diagnosis and recovery may be autonomous, ground-directed, or combined according to mission	Functional	No Relationship	N/A	N/A	N/A	0		
CM0085	Electromagnetic Shielding	Spacecraft electronics are vulnerable to natural ionizing particle radiation and intentional electromagnetic threats such as high-power microwave and electromagnetic pulse effects. Both may cause transient upset or permanent damage, but they act through different physical mechanisms and require distinct protections. Conductive enclosures and associated electromagnetic protection reduce fields and induced transients from HPM or EMP, while particle-radiation shielding reduces the dose or particle environment reaching susceptible components. The spacecraft design must address particle-radiation protection and HPM or EMP protection as coordinated but separately verified requirements. Enclosure materials, geometry, penetrations, bonding, and component placement should be evaluated together so that protection against one environment does not create unacceptable mass, thermal, electrical, or secondary-radiation effects in another. Shielding is primarily a design- and integration-phase hardware control and generally cannot be increased after launch. It must be combined with component hardness assurance and electrical protection measures sufficient to meet the mission's residual susceptibility requirements.	Functional	No Relationship	N/A	N/A	N/A	0		

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0018	Dynamic Testing	Dynamic analysis subjects executing software and firmware to active testing conditions to identify weaknesses and vulnerabilities that static analysis alone cannot detect, including runtime memory corruption, logic errors, timing vulnerabilities, and interface behaviors that only manifest under operational conditions. Applicable techniques include simulation-based testing, penetration testing, fuzz testing, and adversarial emulation using realistic threat actor tactics, techniques, and procedures (TTPs). Dynamic testing must be applied to all software and firmware categories: mission-developed code, open-source components, commercial off-the-shelf (COTS) software, and third-party developed code. Dynamic testing should be performed on potential system elements before acceptance, include realistic simulation of known adversary TTPs, and tools, and continue throughout the lifecycle on applicable physical and logical systems, elements, and processes. Full-scale hardware integration testbeds, commonly referred to as flat satellite (FLATSAT) environments, provide high-fidelity physical test platforms for dynamic analysis; digital twins implemented via instruction set simulation, also known as emulation, provide a flexible and scalable alternative environment capable of supporting TTP execution across a broad range of test scenarios without requiring physical hardware availability.	Functional	Intersects With	Dynamic Code Analysis	TDA-17.2	Mechanisms exist to require the developers of Technology Assets, Applications and/or Services (TAAS) to employ dynamic code analysis tools to identify and remediate common flaws and document the results of the analysis.	5		TDA-09.3
CM0062	Dummy Process - Aggregator Node	In spacecraft architectures containing aggregator or relay nodes, observable computation and communication patterns may reveal valid aggregation cycles and traffic-flow relationships involving critical nodes, root nodes, or ground termination points. This countermeasure is applicable when the threat model identifies a credible adversary capability to observe node power or electromagnetic activity and correlate those observations with RF transmission activity. While camouflaging all network traffic through constant high-power transmission is energy-prohibitive, selectively obscuring aggregator node behavior through dummy process execution provides a practical alternative. This countermeasure requires aggregator nodes to execute dummy workloads whose observable characteristics are sufficiently similar to genuine aggregation cycles to make reliable classification difficult within the mission-defined adversary model. Evaluation should consider applicable power, electromagnetic, execution-duration, processor-activity, memory-access, and RF-correlated features rather than matching only an average power-consumption curve. Two properties are essential for effectiveness: first, dummy processes must vary in their execution pattern, using a different dummy process each time or maintaining a low repetition rate, to prevent adversaries from identifying a distinguishable signature that differentiates dummy from genuine execution; second, timing of dummy execution must be carefully controlled, with a dummy process executed every time the aggregator receives a transmission and randomly during idle periods, to prevent adversaries from correlating the presence or absence of radio frequency (RF) transmissions with power consumption curves to identify and discard dummy activity. Together, these properties are intended to reduce an observer's ability to distinguish valid aggregation cycles and infer traffic flow toward a critical root or base-station node. They do not guarantee that aggregator nodes, network topology, or the	Functional	No Relationship	N/A	N/A	N/A	0		
CM0064	Dual Layer Protection	Dual-layer physical enclosure protection reduces thermal side-channel leakage by combining an inner heat-spreading layer with an outer thermally insulating barrier. The design is intended to attenuate the spatial and temporal relationship between internal computational activity and externally observable surface-temperature patterns; it must not be assumed to make thermal activity completely unobservable. Thermal side-channel attacks observe temperature or infrared-emission patterns associated with device activity to infer information such as workload, active functional regions, execution behavior, or, under applicable adversary and measurement conditions, security-sensitive computation. The inner conductive layer spreads heat laterally to reduce localized temperature gradients and shorten the persistence of internal hot spots. Residual gradients and transient patterns may remain because of component placement, enclosure geometry, interface conductance, workload, and the available heat-rejection path. The outer thermally insulating layer limits direct access to the inner heat-spreading surface and attenuates the propagation of short-duration temperature gradients to the observable exterior. The external surface may still exhibit temperature or infrared-emission variations that must be evaluated against the mission-defined adversary capability. This countermeasure can operate passively without direct computational or electrical-power consumption. However, it can impose mass, volume, thermal-resistance, heat-rejection, structural, integration, and qualification impacts and may indirectly increase demand on active thermal-control systems. Its suitability must therefore be evaluated against both security and	Functional	No Relationship	N/A	N/A	N/A	0		
CM0076	Diversified Architectures	A diversified mission architecture provides a capability through multiple systems, platforms, payloads, orbital regimes, or domains to reduce the mission impact of losing any individual element and increase the range of adversary capabilities required to achieve mission denial. Diversification differs from proliferation in that it employs heterogeneous systems, potentially across different orbits, domains, operators, and technologies, rather than deploying more units of the same design. This heterogeneity imposes asymmetric costs on adversaries: attacking systems across different orbital regimes requires different physical and electronic capabilities for each regime, and kinetic attacks on space assets in diverse orbits carry differentiated collateral debris consequences that increase the political and economic cost of a broad attack campaign. Domain diversification, extending mission capability delivery across space, airborne, and terrestrial layers, further reduces adversary incentive by ensuring that defeating the space layer alone does not deny the end user the underlying capability. Diversification can preserve minimum mission capability following the loss of individual elements when the remaining systems provide sufficient coverage, capacity, interoperability, and operational availability to compensate for the	Functional	No Relationship	N/A	N/A	N/A	0		
CM0074	Distributed Constellations	A distributed constellation architecture deploys mission capability across multiple spacecraft nodes operating collectively, such that the end user is not dependent on any single satellite to derive the intended capability. This architectural approach directly complicates adversary counterspace planning by multiplying the number of assets that must be successfully degraded or destroyed to achieve mission denial effects equivalent to those achievable against a concentrated, single-node architecture. The resilience benefit depends on how much mission capability remains available following the loss or degradation of specified nodes. A constellation that can satisfy defined minimum mission requirements through multiple combinations of surviving nodes generally requires an adversary to affect more assets or shared dependencies to achieve mission denial. GPS exemplifies this principle: a receiver generally uses signals from at least four healthy satellites with suitable geometry to determine three-dimensional position and time. Loss of one satellite does not ordinarily eliminate the service where sufficient healthy satellites remain visible; resilience to ground-system failures depends separately on the redundancy and distribution of the control segment. Distribution is a mission architecture decision that must be made early in the program lifecycle, as it fundamentally shapes spacecraft design, ground system architecture, launch strategy, and operational concepts.	Functional	Intersects With	Distributed Processing & Storage	SEA-25	Mechanisms exist to distribute processing and storage across multiple physical locations.	5		SEA-15

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0037	Disable Physical Ports	Physical data connection, debug, programming, and maintenance interfaces (e.g., joint test action group (JTAG)) that are not required for spacecraft operations must be disabled, removed, or otherwise made inaccessible before spacecraft operations begin. Interfaces required for operational functions must be explicitly identified and protected against unauthorized physical access and use. These interfaces, essential during development for programming, debugging, and testing, represent persistent attack surfaces in the operational environment: an adversary with physical access to the spacecraft before launch, during ground handling, or at a shared launch facility could exploit active debug interfaces to read memory, modify firmware, bypass security controls, or implant persistent malicious code without leaving detectable traces in software-visible logs. Disabling or removing unused physical interfaces closes a direct hardware-access pathway and reduces reliance on procedural controls or physical security alone. The capability to disable these interfaces must be designed into the system from the outset, as physical removal or reliable hardware-enforced disablement cannot be easily retrofitted into a completed	Functional	Intersects With	Secure Baseline Configurations	CFG-04	Mechanisms exist to develop, document and maintain secure baseline configurations for Technology Assets, Applications and/or Services (TAAS) that are consistent with industry-accepted hardening standards.	5		CFG-02
CM0037	Disable Physical Ports	Physical data connection, debug, programming, and maintenance interfaces (e.g., joint test action group (JTAG)) that are not required for spacecraft operations must be disabled, removed, or otherwise made inaccessible before spacecraft operations begin. Interfaces required for operational functions must be explicitly identified and protected against unauthorized physical access and use. These interfaces, essential during development for programming, debugging, and testing, represent persistent attack surfaces in the operational environment: an adversary with physical access to the spacecraft before launch, during ground handling, or at a shared launch facility could exploit active debug interfaces to read memory, modify firmware, bypass security controls, or implant persistent malicious code without leaving detectable traces in software-visible logs. Disabling or removing unused physical interfaces closes a direct hardware-access pathway and reduces reliance on procedural controls or physical security alone. The capability to disable these interfaces must be designed into the system from the outset, as physical removal or reliable hardware-enforced disablement cannot be easily retrofitted into a completed	Functional	Intersects With	Configure Technology Assets, Applications and/or Services (TAAS) for High-Risk Areas	CFG-05	Mechanisms exist to configure Technology Assets, Applications and/or Services (TAAS) utilized in high-risk areas with more restrictive secure baseline configurations.	5		CFG-02.5
CM0037	Disable Physical Ports	Physical data connection, debug, programming, and maintenance interfaces (e.g., joint test action group (JTAG)) that are not required for spacecraft operations must be disabled, removed, or otherwise made inaccessible before spacecraft operations begin. Interfaces required for operational functions must be explicitly identified and protected against unauthorized physical access and use. These interfaces, essential during development for programming, debugging, and testing, represent persistent attack surfaces in the operational environment: an adversary with physical access to the spacecraft before launch, during ground handling, or at a shared launch facility could exploit active debug interfaces to read memory, modify firmware, bypass security controls, or implant persistent malicious code without leaving detectable traces in software-visible logs. Disabling or removing unused physical interfaces closes a direct hardware-access pathway and reduces reliance on procedural controls or physical security alone. The capability to disable these interfaces must be designed into the system from the outset, as physical removal or reliable hardware-enforced disablement cannot be easily retrofitted into a completed	Functional	Intersects With	Interface Security	EMB-07	Mechanisms exist to protect embedded devices against unauthorized use of the physical factory diagnostic and test interface(s).	5		EMB-04
CM0037	Disable Physical Ports	Physical data connection, debug, programming, and maintenance interfaces (e.g., joint test action group (JTAG)) that are not required for spacecraft operations must be disabled, removed, or otherwise made inaccessible before spacecraft operations begin. Interfaces required for operational functions must be explicitly identified and protected against unauthorized physical access and use. These interfaces, essential during development for programming, debugging, and testing, represent persistent attack surfaces in the operational environment: an adversary with physical access to the spacecraft before launch, during ground handling, or at a shared launch facility could exploit active debug interfaces to read memory, modify firmware, bypass security controls, or implant persistent malicious code without leaving detectable traces in software-visible logs. Disabling or removing unused physical interfaces closes a direct hardware-access pathway and reduces reliance on procedural controls or physical security alone. The capability to disable these interfaces must be designed into the system from the outset, as physical removal or reliable hardware-enforced disablement cannot be easily retrofitted into a completed	Functional	Intersects With	Prevent Alterations	EMB-09	Mechanisms exist to protect embedded devices by preventing the unauthorized installation and execution of software.	5		EMB-06
CM0004	Development Environment Security	A secure development environment requires a current and sufficiently complete inventory of the people, devices, software, services, credentials, and automated identities capable of accessing or influencing the environment. The development environment includes source-code repositories, developer workstations, build servers, CI/CD runners, compiler and linker toolchains, container and virtual machine images, package registries, artifact repositories, signing systems, test environments, integration laboratories, and release-staging systems. For space systems, these environments may produce or manage flight software images, firmware, FPGA bitstreams, software-defined radio waveforms, command and telemetry databases, configuration tables, ephemerides, calibration products, fault-management logic, and on-orbit update packages. Unmanaged assets, unauthorized access, compromised dependencies, altered toolchains, and untrusted build services are significant pathways through which adversaries may maliciously modify software or other mission artifacts during development and build activities. All personnel and assets touching the development environment must be inventoried and actively managed. MFA shall be enforced for human access, while non-human identities shall use managed workload identities, scoped credentials, protected secrets, and defined rotation or expiration period, with particular rigor applied to code repositories, where threat actors may attempt to inject malicious code into software under development without detection. Zero-trust access controls should govern repository access, with protected branch and tag policies shall restrict direct modification, prohibit unauthorized force pushes or deletion, require successful security checks, and require independent review before merging or releasing critical code. Effective development environment security also requires integrated change management, privilege management, comprehensive audit logging, and continuous in-depth monitoring	Functional	Intersects With	Secure Development Environments	TDA-15	Mechanisms exist to maintain a segmented development network to ensure a secure development environment.	5		TDA-07

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0013	Dependency Confusion	Dependency confusion is a software supply chain attack in which an adversary publishes a malicious package to a public repository using the same name as an organization's internal private package, exploiting build tools that may inadvertently resolve the public version over the intended internal one. Mitigating this attack requires a layered set of controls: internal dependencies must be sourced exclusively from private, controlled repositories rather than public package registries; build and continuous integration and continuous delivery (CI/CD) pipeline configurations must explicitly define trusted repository sources and resolution order to prevent inadvertent public package substitution; and dependency integrity must be validated at the point of consumption by verifying that cryptographic checksums match those of the known-good official packages. A secure build environment is a prerequisite for these controls to be effective, as a compromised build environment can undermine dependency controls regardless of repository configuration. Together, these measures ensure that only verified, intended dependencies are incorporated into flight software, ground system software, and supporting toolchains.	Functional	Intersects With	Asset-Service Dependencies	AST-06	Mechanisms exist to identify and assess the security of Technology Assets, Applications and/or Services (TAAS) that support more than one critical business function.	5		AST-01.1
CM0081	Defensive Jamming and Spoofing	Defensive jamming and spoofing are active electronic countermeasures that may disrupt or deceive the terminal guidance sensors of an incoming kinetic anti-satellite weapon. When combined with evasive maneuvering, these measures may reduce the accuracy of the threat's targeting solution and lower the probability of a successful intercept. Effectiveness depends on timely threat detection, knowledge of the relevant sensor characteristics, available transmit power and geometry, and the threat's ability to recognize or overcome the countermeasure. Development, testing, and employment of these capabilities must occur only under applicable governmental authorization, spectrum authority, rules of engagement, and information-protection requirements. The design must limit unintended interference and account for effects on friendly, civil, and safety-related	Functional	No Relationship	N/A	N/A	N/A	0		
CM0087	Defensive Dazzling/Blinding	Defensive dazzling and blinding employs directed laser energy to degrade or defeat the optical or infrared (IR) sensors of adversary systems, providing an active countermeasure against kinetic anti-satellite (ASAT) weapons and adversary reconnaissance platforms in the space domain. Against kinetic ASAT threats, laser energy directed toward the terminal guidance sensor of an incoming weapon may temporarily saturate, disrupt, or damage the sensor and reduce the accuracy of terminal guidance. When coordinated with evasive maneuvering, this capability may reduce the probability of a successful intercept. Effectiveness depends on timely threat detection, target-sensor characteristics, engagement geometry, laser performance, pointing accuracy, dwell time, and the threat's ability to maintain or recover guidance. Against adversary inspector satellites or SDA collection platforms, defensive dazzling may temporarily degrade or deny optical or infrared collection of the protected spacecraft. Permanent sensor damage is a materially different effect that requires separate authorization, targeting criteria, and escalation analysis. Dazzling generally produces temporary sensor degradation, while blinding produces permanent sensor damage. Both effects carry legal, policy, safety, and escalation implications, but their legal characterization depends on the target, circumstances, intended effect, actual consequences, and applicable national and international authorities. Employment authorities and prohibited target or effect categories must be established before the capability is operationally relied upon.	Functional	No Relationship	N/A	N/A	N/A	0		
CM0082	Deception and Decoys	Deception and decoy techniques can reduce the accuracy or confidence of adversary assessments concerning spacecraft location, capability, operational status, mission type, or constellation robustness. Ground segment honeypots, such as HoneySat, extend deception into the cyber domain by simulating realistic satellite ground infrastructure and mission control systems to attract, deceive, and collect intelligence on adversaries attempting network-based compromise of satellite operations. Their effectiveness depends on whether the deception remains credible when evaluated across the observable signatures and intelligence sources available to the adversary. Strategic deception encompasses information operations approaches such as controlled public messaging and launch announcements that limit disclosure or actively introduce uncertainty about satellite capabilities, as well as operational practices that conceal spacecraft functions through careful management of observable behaviors and emissions. On-orbit capability deception, enabled by swappable payload modules and on-orbit servicing vehicles that periodically transfer payloads between satellites, creates persistent uncertainty in the adversary's intelligence picture about which capabilities are resident on which platform at any given time, directly complicating targeting calculus. Tactical decoys provide active point defense by creating false targets that confuse the sensors of anti-satellite (ASAT) weapons and space domain awareness (SDA) surveillance systems; physical decoys, such as deployable inflatable devices that replicate a satellite's size and radar cross-section, and electromagnetic decoys that mimic a spacecraft's radio frequency (RF) signature, can each divert adversary attention and degrade the reliability of tracking and targeting solutions. Multiple decoys stored onboard for sequential deployment extend the utility of the capability across engagement scenarios. Cyber-layer deception through satellite	Functional	No Relationship	N/A	N/A	N/A	0		
CM0056	Data Backup	A mission's ability to recover from a cyber incident, hardware failure, or adversary action depends directly on the availability of verified, uncorrupted backups of critical data that are stored independently from the primary systems those backups are intended to restore. Data backup procedures must be defined within a broader disaster recovery plan that specifies what data is backed up, at what frequency, through what process, and under what conditions restoration will be initiated. At least one recoverable backup copy must be stored outside the primary system's administrative and failure domains and protected so that compromise of ordinary production systems, credentials, or management services does not provide the ability to modify or destroy that copy. Separation may use offline media, physically separate infrastructure, isolated storage systems, separate cloud accounts or security domains, immutable retention controls, or an approved combination of these mechanisms. Backup storage must be protected against the methods adversaries commonly use to target recovery capability, including ransomware that encrypts or deletes backup repositories, credential attacks against backup management systems, and physical access to backup media. Backup integrity must be verifiable, as a backup that has been silently corrupted or tampered with provides no recovery capability when needed.	Functional	Intersects With	Resilient Data Backup Architecture	BCD-23	Mechanisms exist to maintain a secure, immutable backup architecture.	8		

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0056	Data Backup	A mission's ability to recover from a cyber incident, hardware failure, or adversary action depends directly on the availability of verified, uncorrupted backups of critical data that are stored independently from the primary systems those backups are intended to restore. Data backup procedures must be defined within a broader disaster recovery plan that specifies what data is backed up, at what frequency, through what process, and under what conditions restoration will be initiated. At least one recoverable backup copy must be stored outside the primary system's administrative and failure domains and protected so that compromise of ordinary production systems, credentials, or management services does not provide the ability to modify or destroy that copy. Separation may use offline media, physically separate infrastructure, isolated storage systems, separate cloud accounts or security domains, immutable retention controls, or an approved combination of these mechanisms. Backup storage must be protected against the methods adversaries commonly use to target recovery capability, including ransomware that encrypts or deletes backup repositories, credential attacks against backup management systems, and physical access to backup media. Backup integrity must be verifiable, as a backup that has been silently corrupted or tampered with provides no recovery capability when needed.	Functional	Intersects With	Data Backups	BCD-24	Mechanisms exist to create recurring backups of data, software and/or system images, as well as verify the integrity of these backups, to ensure the availability of the data to satisfy Recovery Time Objectives (RTOs) and Recovery Point Objectives (RPOs).	5		BCD-11
CM0044	Cyber-safe Mode	Cyber-safe mode is a dedicated, configuration-controlled spacecraft operating state entered autonomously or by authorized ground command when mission-defined conditions indicate a credible threat to platform integrity. In this state, nonessential functions are shut down or isolated and the spacecraft operates from an integrity-protected, validated software and configuration baseline. Unlike traditional safe mode, which addresses hardware faults and operational anomalies, cyber-safe mode is specifically designed to respond to cyber threats, providing a secure recovery baseline from which the spacecraft can reconstitute compromised functions. Authentication and encryption must remain enabled within cyber-safe mode, ensuring that the reduced operational state does not degrade the security posture of the vehicle. The cyber-safe mode software and configuration must be stored onboard using hardware-based protections that prevent modification by nominal flight software, ordinary commands, and other untrusted execution paths. Where baseline updates are permitted, they must use a separately authorized and integrity-verified maintenance process that preserves a recoverable trusted version. Following entry into cyber-safe mode, the spacecraft must be capable of reconstituting firmware and software functions to pre-attack capability levels, either autonomously through self-healing mechanisms or with ground assistance, and must be capable of replanning operations based on whatever equipment remains available after the cyber event. The primary recovery objective is restoration of full mission capability; where that is not achievable, the spacecraft should attain the maximum reduced mission capability available given the post-attack	Functional	Intersects With	Fail Secure	SEA-15	Mechanisms exist to enable systems to fail to an organization-defined known-state for types of failures, preserving system state information in failure.	5		SEA-07.2
CM0016	CWE List	A prioritized list of software weakness classes, drawn from a recognized weakness enumeration taxonomy such as the common weakness enumeration (CWE) framework, provides the analytical foundation for targeted, mission-relevant static code analysis. Rather than treating all weakness classes as equally significant, mission teams should derive a system-specific priority ranking based on the spacecraft or ground system's architecture, threat environment, operational constraints, and consequence of exploitation for each weakness class. This prioritized list serves two functions: it directs static analysis tool configuration toward the weakness classes most relevant to the mission, and it provides a defensible, documented basis for triaging and ranking analysis findings so that remediation resources are applied where mission risk is highest. The prioritization rationale should be documented and maintained as a controlled artifact, as it reflects security-relevant design decisions about the system's risk	Functional	Intersects With	Vulnerability & Patch Management Program (VPMP)	VPM-02	Mechanisms exist to facilitate the implementation and monitoring of risk-based vulnerability management capabilities.	5		VPM-01
CM0016	CWE List	A prioritized list of software weakness classes, drawn from a recognized weakness enumeration taxonomy such as the common weakness enumeration (CWE) framework, provides the analytical foundation for targeted, mission-relevant static code analysis. Rather than treating all weakness classes as equally significant, mission teams should derive a system-specific priority ranking based on the spacecraft or ground system's architecture, threat environment, operational constraints, and consequence of exploitation for each weakness class. This prioritized list serves two functions: it directs static analysis tool configuration toward the weakness classes most relevant to the mission, and it provides a defensible, documented basis for triaging and ranking analysis findings so that remediation resources are applied where mission risk is highest. The prioritization rationale should be documented and maintained as a controlled artifact, as it reflects security-relevant design decisions about the system's risk	Functional	Intersects With	Vulnerability Exploitation Analysis	VPM-04	Mechanisms exist to identify, assess, prioritize and document the potential impact(s) and likelihood(s) of applicable internal and external threats exploiting known vulnerabilities.	5		VPM-03.1
CM0016	CWE List	A prioritized list of software weakness classes, drawn from a recognized weakness enumeration taxonomy such as the common weakness enumeration (CWE) framework, provides the analytical foundation for targeted, mission-relevant static code analysis. Rather than treating all weakness classes as equally significant, mission teams should derive a system-specific priority ranking based on the spacecraft or ground system's architecture, threat environment, operational constraints, and consequence of exploitation for each weakness class. This prioritized list serves two functions: it directs static analysis tool configuration toward the weakness classes most relevant to the mission, and it provides a defensible, documented basis for triaging and ranking analysis findings so that remediation resources are applied where mission risk is highest. The prioritization rationale should be documented and maintained as a controlled artifact, as it reflects security-relevant design decisions about the system's risk	Functional	Intersects With	Vulnerability Ranking	VPM-05	Mechanisms exist to identify and assign a risk ranking to newly discovered security vulnerabilities using reputable outside sources for security vulnerability information.	5		VPM-03
CM0030	Crypto Key Management	Effective cryptographic key management is a foundational requirement for all mission encryption and authentication functions; the security of cryptographic implementations is only as strong as the protection afforded to the keys those implementations rely upon. Key management must conform to recognized cryptographic guidance and address the full lifecycle applicable to each key type, including generation or establishment, distribution, storage, activation, use, replacement, deactivation or revocation, recovery where authorized, compromise response, and destruction. Only approved cryptographic algorithms, key generation methods, key distribution techniques, and authentication mechanisms may be used; the use of unapproved, deprecated, or custom cryptographic primitives is prohibited regardless of perceived functional adequacy. Encryption key handling must be performed outside of onboard software and protected through dedicated cryptographic mechanisms, preventing keys from being exposed through software vulnerabilities, memory inspection, or software-level debugging interfaces. Secret and private key material must not be retrievable in plaintext through telecommands, telemetry, diagnostic outputs, debugging interfaces, or other externally accessible mechanisms, regardless of the privilege level of the requesting entity.	Functional	Intersects With	Cryptographic Key Management	CRY-10	Mechanisms exist to facilitate cryptographic key management controls to protect the confidentiality, integrity and availability of keys.	5		CRY-09

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0022	Criticality Analysis	Criticality analysis is a structured engineering process that identifies the mission functions, system components, and data flows whose compromise, degradation, or loss would most severely impact mission success, crew safety, or operational continuity. The outputs of this analysis directly drive security investment prioritization: components and functions assessed as most critical receive the most rigorous design-phase protections, supply chain scrutiny, and operational security controls, while lower-criticality elements are protected proportionately. Criticality analysis findings should inform the application of complementary security design principles, including network and functional segmentation and least-privilege access control, to isolate critical components from less-trusted system elements and reduce the consequence of compromise elsewhere in the system. Supply chain protection resources and oversight rigor should be explicitly allocated in proportion to component criticality, ensuring that the most mission-essential hardware and software receive the most intensive sourcing controls, provenance verification, and supplier oversight. Criticality analysis must be initiated early in the system design process and updated as the architecture evolves, threat intelligence changes, or operational experience reveals previously	Functional	Intersects With	Asset-Service Dependencies	AST-06	Mechanisms exist to identify and assess the security of Technology Assets, Applications and/or Services (TAAS) that support more than one critical business function.	5		AST-01.1
CM0022	Criticality Analysis	Criticality analysis is a structured engineering process that identifies the mission functions, system components, and data flows whose compromise, degradation, or loss would most severely impact mission success, crew safety, or operational continuity. The outputs of this analysis directly drive security investment prioritization: components and functions assessed as most critical receive the most rigorous design-phase protections, supply chain scrutiny, and operational security controls, while lower-criticality elements are protected proportionately. Criticality analysis findings should inform the application of complementary security design principles, including network and functional segmentation and least-privilege access control, to isolate critical components from less-trusted system elements and reduce the consequence of compromise elsewhere in the system. Supply chain protection resources and oversight rigor should be explicitly allocated in proportion to component criticality, ensuring that the most mission-essential hardware and software receive the most intensive sourcing controls, provenance verification, and supplier oversight. Criticality analysis must be initiated early in the system design process and updated as the architecture evolves, threat intelligence changes, or operational experience reveals previously	Functional	Intersects With	Asset Categorization	AST-15	Mechanisms exist to categorize Technology Assets, Applications and/or Services (TAAS).	5		AST-31
CM0022	Criticality Analysis	Criticality analysis is a structured engineering process that identifies the mission functions, system components, and data flows whose compromise, degradation, or loss would most severely impact mission success, crew safety, or operational continuity. The outputs of this analysis directly drive security investment prioritization: components and functions assessed as most critical receive the most rigorous design-phase protections, supply chain scrutiny, and operational security controls, while lower-criticality elements are protected proportionately. Criticality analysis findings should inform the application of complementary security design principles, including network and functional segmentation and least-privilege access control, to isolate critical components from less-trusted system elements and reduce the consequence of compromise elsewhere in the system. Supply chain protection resources and oversight rigor should be explicitly allocated in proportion to component criticality, ensuring that the most mission-essential hardware and software receive the most intensive sourcing controls, provenance verification, and supplier oversight. Criticality analysis must be initiated early in the system design process and updated as the architecture evolves, threat intelligence changes, or operational experience reveals previously	Functional	Intersects With	Network Diagrams & Data Flow Diagrams (DFDs)	AST-17	Mechanisms exist to maintain network architecture diagrams that: (1) Contain sufficient detail to assess the security of the network's architecture; (2) Reflect the current architecture of the network environment; and (3) Document all sensitive and/or regulated data flows.	5		AST-04
CM0022	Criticality Analysis	Criticality analysis is a structured engineering process that identifies the mission functions, system components, and data flows whose compromise, degradation, or loss would most severely impact mission success, crew safety, or operational continuity. The outputs of this analysis directly drive security investment prioritization: components and functions assessed as most critical receive the most rigorous design-phase protections, supply chain scrutiny, and operational security controls, while lower-criticality elements are protected proportionately. Criticality analysis findings should inform the application of complementary security design principles, including network and functional segmentation and least-privilege access control, to isolate critical components from less-trusted system elements and reduce the consequence of compromise elsewhere in the system. Supply chain protection resources and oversight rigor should be explicitly allocated in proportion to component criticality, ensuring that the most mission-essential hardware and software receive the most intensive sourcing controls, provenance verification, and supplier oversight. Criticality analysis must be initiated early in the system design process and updated as the architecture evolves, threat intelligence changes, or operational experience reveals previously	Functional	Intersects With	Criticality Analysis During Development	TDA-05.1	Mechanisms exist to require the developer of the Technology Asset, Application and/or Service (TAAS) to perform a criticality analysis at organization-defined decision points in the Secure Development Life Cycle (SDLC).	5		TDA-06.1
CM0022	Criticality Analysis	Criticality analysis is a structured engineering process that identifies the mission functions, system components, and data flows whose compromise, degradation, or loss would most severely impact mission success, crew safety, or operational continuity. The outputs of this analysis directly drive security investment prioritization: components and functions assessed as most critical receive the most rigorous design-phase protections, supply chain scrutiny, and operational security controls, while lower-criticality elements are protected proportionately. Criticality analysis findings should inform the application of complementary security design principles, including network and functional segmentation and least-privilege access control, to isolate critical components from less-trusted system elements and reduce the consequence of compromise elsewhere in the system. Supply chain protection resources and oversight rigor should be explicitly allocated in proportion to component criticality, ensuring that the most mission-essential hardware and software receive the most intensive sourcing controls, provenance verification, and supplier oversight. Criticality analysis must be initiated early in the system design process and updated as the architecture evolves, threat intelligence changes, or operational experience reveals previously	Functional	Intersects With	Third-Party Criticality Assessments	TPM-09	Mechanisms exist to identify, prioritize and assess suppliers and partners of critical Technology Assets, Applications and/or Services (TAAS) using a supply chain risk assessment process relative to their importance in supporting the delivery of high-value services.	5		TPM-02
CM-NA	Countermeasure Not Identified	No Applicable Countermeasures	Functional	No Relationship	N/A	N/A	N/A	0		

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0090	Continuous Monitoring	Continuous monitoring maintains persistent, real-time or near-real-time visibility into the security posture of spacecraft, ground systems, and mission networks, providing the ongoing situational awareness required to support informed risk management decisions throughout the mission lifecycle. Unlike point-in-time assessments that capture a snapshot of security posture at a specific moment, continuous monitoring detects changes in system configuration, software vulnerabilities, threat indicators, and control effectiveness as they occur, enabling faster detection of and response to security-relevant events before they escalate into mission-impacting incidents. For space missions, continuous monitoring spans both the cyber domain, including ground system network activity, software configuration compliance, vulnerability status, and access control events, and the physical and operational domains, including spacecraft telemetry indicators of anomalous behavior, link quality indicators of potential radio frequency (RF) interference, and space domain awareness data indicating proximity threats. The output of continuous monitoring feeds directly into risk management decision-making, providing mission owners and security teams with the current information needed to prioritize remediation actions, authorize changes, and adjust defensive posture in response to the evolving threat and vulnerability landscape.	Functional	Intersects With	Continuous Monitoring	MON-02	Mechanisms exist to facilitate the implementation of enterprise-wide monitoring controls.	5		MON-01
CM0023	Configuration Management	Configuration management (CM) for space systems requires automated mechanisms to establish, maintain, validate, and report on the approved baseline configuration of spacecraft and supporting ground systems, ensuring the baseline remains current, complete, and accurate throughout the mission lifecycle. Automated mechanisms should support configuration management by improving the accuracy, currency, and validation of configuration records. Automation does not replace formal change control, engineering review, approval, or other configuration management activities that require human judgment. The approved baseline must be readily accessible to authorized personnel for operational decision-making, anomaly investigation, and change impact assessment. Deviations between the documented baseline and the actual system configuration represent both security and operational risk because undocumented changes may introduce vulnerabilities, mask adversary activity, or produce unpredictable system behavior. Automated validation should identify and report configuration drift at mission-defined intervals or following relevant change events, based on system observability and mission risk.	Functional	Intersects With	Configuration Management Program	CFG-02	Mechanisms exist to facilitate the implementation of configuration management controls.	5		CFG-01
CM0023	Configuration Management	Configuration management (CM) for space systems requires automated mechanisms to establish, maintain, validate, and report on the approved baseline configuration of spacecraft and supporting ground systems, ensuring the baseline remains current, complete, and accurate throughout the mission lifecycle. Automated mechanisms should support configuration management by improving the accuracy, currency, and validation of configuration records. Automation does not replace formal change control, engineering review, approval, or other configuration management activities that require human judgment. The approved baseline must be readily accessible to authorized personnel for operational decision-making, anomaly investigation, and change impact assessment. Deviations between the documented baseline and the actual system configuration represent both security and operational risk because undocumented changes may introduce vulnerabilities, mask adversary activity, or produce unpredictable system behavior. Automated validation should identify and report configuration drift at mission-defined intervals or following relevant change events, based on system observability and mission risk.	Functional	Intersects With	Automated Baseline Configuration Management & Verification	CFG-07.1	Automated mechanisms exist to govern and report on secure baseline configurations of Technology Assets, Applications and/or Services (TAAS) through Continuous Diagnostics and Mitigation (CDM), or similar technologies.	5		CFG-02.2
CM0002	COMSEC	Communications security (COMSEC) denies unauthorized parties access to information derived from telecommunications while ensuring the authenticity of those communications. COMSEC is commonly defined as a broad discipline that may encompass cryptographic security, transmission security, emissions security, cryptographic key management, traffic-flow security, and physical security of COMSEC material. Within SPARTA, these areas are further broken down through separate countermeasures, including CM0029 | TRANSEC, CM0030 | Crypto Key Management, CM0003 | TEMPEST/EMSEC, and CM0073 | Traffic Flow Analysis Defense. CM0002 provides the overarching communications-security context and supports the coordinated application of these specialized countermeasures. All mission links, particularly telemetry, tracking, and commanding (TT&C) links, should employ communications-security protections appropriate to the sensitivity, criticality, operational environment, and threat exposure of the information being exchanged. These protections may include cryptographic protection, transmission security, emissions security, traffic-flow protection, secure key management, and physical protection of COMSEC material, as addressed by the applicable specialized countermeasures. Spacecraft should not provide an operational mode that permits required cryptographic protection or command authentication on TT&C links to be bypassed or disabled. Operational, maintenance, test, recovery, and contingency modes should be considered when evaluating whether communications-security protections can be unintentionally or improperly circumvented. Communication receivers and associated signal-processing or TRANSEC mechanisms should detect and, when mission-defined criteria are met, reject or otherwise safely handle transmissions exhibiting anomalous signal characteristics consistent with communications deception. Cryptographic mechanisms should	Functional	Intersects With	Network Security Controls (NSC)	NET-02	Mechanisms exist to develop, govern & update procedures to facilitate the implementation of Network Security Controls (NSC).	5		NET-01
CM0071	Communication Physical Medium	The physical medium selected for ground segment networking directly affects the mission's vulnerability to passive interception, traffic flow analysis, and electromagnetic eavesdropping, making medium selection a security design decision that must be informed by the mission's threat model. Fiber optic cabling transmits data optically and does not produce the same conducted or radiated electrical emanations as copper cabling, reducing exposure to proximity-based electromagnetic collection. Fiber may still be intercepted through physical access, optical coupling, compromised patch points, or endpoint equipment, and medium selection alone does not conceal traffic timing or volume from an observer with access to the link or its endpoints. Fiber should therefore be considered where the threat model identifies electromagnetic emanation or physical-medium interception risk, while cryptographic and physical protections remain applicable, particularly for links carrying mission-sensitive data such as command uplink traffic, cryptographic key material, or mission planning data. The selection of physical medium should be treated as a threat-informed engineering decision evaluated for each network segment based on the sensitivity of the data carried, the physical accessibility of the cabling, and the threat environment of the facility in	Functional	No Relationship	N/A	N/A	N/A	0		

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0017	Coding Standard	A formally defined coding standard establishes the rules, conventions, and constraints that govern how software is written across the mission's development program, directly influencing the security, maintainability, and verifiability of the delivered system. The standard must specify acceptable programming language types, with language selection driven by a documented evaluation of security requirements, application complexity, scalability needs, available development resources, schedule constraints, and the availability of security-relevant language features such as memory safety, type safety, and bounds checking. Language choices that introduce classes of vulnerability by design, such as languages without memory safety guarantees used in contexts where memory corruption is a plausible attack vector, require explicit justification and compensating controls. The coding standard must include security-relevant rules for input validation, error handling, cryptographic usage, memory management, and concurrency, as applicable to the selected languages and system design. Adherence should be evaluated through automated means where supported, supplemented by manual review for requirements that cannot be reliably automated.	Functional	Intersects With	Software Assurance Maturity Model (SAMM)	TDA-03.1	Mechanisms exist to utilize a Software Assurance Maturity Model (SAMM) to govern a secure development lifecycle for the development of Technology Assets, Applications and/or Services (TAAS).	5		TDA-06.3
CM0017	Coding Standard	A formally defined coding standard establishes the rules, conventions, and constraints that govern how software is written across the mission's development program, directly influencing the security, maintainability, and verifiability of the delivered system. The standard must specify acceptable programming language types, with language selection driven by a documented evaluation of security requirements, application complexity, scalability needs, available development resources, schedule constraints, and the availability of security-relevant language features such as memory safety, type safety, and bounds checking. Language choices that introduce classes of vulnerability by design, such as languages without memory safety guarantees used in contexts where memory corruption is a plausible attack vector, require explicit justification and compensating controls. The coding standard must include security-relevant rules for input validation, error handling, cryptographic usage, memory management, and concurrency, as applicable to the selected languages and system design. Adherence should be evaluated through automated means where supported, supplemented by manual review for requirements that cannot be reliably automated.	Functional	Intersects With	Secure Software Development Practices (SSDP)	TDA-05	Mechanisms exist to develop applications based on Secure Software Development Practices (SSDP).	5		TDA-06
CM0006	Cloaking Safe-mode	Safe-mode entry represents a high-risk transition point at which a spacecraft enters a reduced-capability state to preserve vehicle safety and support anomaly recovery. This transition must not create a less secure command, telemetry, or onboard processing environment. To the extent permitted by mission safety and recovery requirements, the spacecraft should avoid unnecessary or uniquely identifying changes in transmission characteristics, beacon content, communication cadence, and externally observable behavior that would allow an adversary to reliably identify and exploit the safe-mode state. Safe-mode shall preserve the mission-defined minimum security posture for every communication path and command mechanism that remains active. This posture should include authentication, data integrity, anti-replay protection, command authorization, command validation, cryptographic key protection, security-relevant logging, and encryption where confidentiality is required. The spacecraft shall not enter a crypto-bypass or unauthenticated command state solely because safe-mode has been activated. The safe-mode software and configuration baseline shall explicitly define the security controls, command dictionaries, alternate receivers, contingency communication paths, rate and size limits, command counters, time tag requirements, interlocks, logging functions, and monitoring capabilities that remain active during safe-mode. These protections shall be designed and verified as part of the safe-mode baseline rather than treated as discretionary functions that may be removed without security impact	Functional	No Relationship	N/A	N/A	N/A	0		
CM0043	Backdoor Commands	All commands capable of being executed on the spacecraft must be known, documented, and accounted for by the mission and spacecraft owner; the existence of undisclosed or undocumented commands, whether introduced by developers, component suppliers, or subsystem vendors, represents an unacceptable and unmanageable risk to mission integrity. Commands capable of adversely affecting mission success if misused must be identified through deliberate analysis and protected by mission-defined authorization and execution controls commensurate with their consequences. Commands that bypass normal operational safeguards require additional justification and restrictions appropriate to their emergency or contingency purpose. Backdoor, residual, hardware-level, test, diagnostic, or override commands that bypass normal operational pathways or safeguards should be retained only where mission-approved emergency or contingency access requires them. Their inclusion must be explicitly justified, and their commanding authority must be appropriately restricted. Hazardous commands required for normal mission operations are not prohibited by this countermeasure but must be governed by approved authorization, prerequisite, sequencing, and safety controls. Any command capability introduced by a subsystem supplier or component vendor that was not explicitly requested or authorized by the mission owner must be identified, evaluated, and either removed or brought under formal mission command governance	Functional	Intersects With	Software Assurance Maturity Model (SAMM)	TDA-03.1	Mechanisms exist to utilize a Software Assurance Maturity Model (SAMM) to govern a secure development lifecycle for the development of Technology Assets, Applications and/or Services (TAAS).	5		TDA-06.3
CM0043	Backdoor Commands	All commands capable of being executed on the spacecraft must be known, documented, and accounted for by the mission and spacecraft owner; the existence of undisclosed or undocumented commands, whether introduced by developers, component suppliers, or subsystem vendors, represents an unacceptable and unmanageable risk to mission integrity. Commands capable of adversely affecting mission success if misused must be identified through deliberate analysis and protected by mission-defined authorization and execution controls commensurate with their consequences. Commands that bypass normal operational safeguards require additional justification and restrictions appropriate to their emergency or contingency purpose. Backdoor, residual, hardware-level, test, diagnostic, or override commands that bypass normal operational pathways or safeguards should be retained only where mission-approved emergency or contingency access requires them. Their inclusion must be explicitly justified, and their commanding authority must be appropriately restricted. Hazardous commands required for normal mission operations are not prohibited by this countermeasure but must be governed by approved authorization, prerequisite, sequencing, and safety controls. Any command capability introduced by a subsystem supplier or component vendor that was not explicitly requested or authorized by the mission owner must be identified, evaluated, and either removed or brought under formal mission command governance	Functional	Intersects With	Secure Software Development Practices (SSDP)	TDA-05	Mechanisms exist to develop applications based on Secure Software Development Practices (SSDP).	5		TDA-06

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0043	Backdoor Commands	All commands capable of being executed on the spacecraft must be known, documented, and accounted for by the mission and spacecraft owner; the existence of undisclosed or undocumented commands, whether introduced by developers, component suppliers, or subsystem vendors, represents an unacceptable and unmanageable risk to mission integrity. Commands capable of adversely affecting mission success if misused must be identified through deliberate analysis and protected by mission-defined authorization and execution controls commensurate with their consequences. Commands that bypass normal operational safeguards require additional justification and restrictions appropriate to their emergency or contingency purpose. Backdoor, residual, hardware-level, test, diagnostic, or override commands that bypass normal operational pathways or safeguards should be retained only where mission-approved emergency or contingency access requires them. Their inclusion must be explicitly justified, and their commanding authority must be appropriately restricted. Hazardous commands required for normal mission operations are not prohibited by this countermeasure but must be governed by approved authorization, prerequisite, sequencing, and safety controls. Any command capability introduced by a subsystem supplier or component vendor that was not explicitly requested or authorized by the mission owner must be identified, evaluated, and either removed or brought under formal mission command governance	Functional	Intersects With	Software Design Review	TDA-05.4	Mechanisms exist to have an independent review of the software design to validate: (1) Applicable security, compliance and resilience requirements are met; and (2) Identified risks are remediated.	5		TDA-06.5
CM0043	Backdoor Commands	All commands capable of being executed on the spacecraft must be known, documented, and accounted for by the mission and spacecraft owner; the existence of undisclosed or undocumented commands, whether introduced by developers, component suppliers, or subsystem vendors, represents an unacceptable and unmanageable risk to mission integrity. Commands capable of adversely affecting mission success if misused must be identified through deliberate analysis and protected by mission-defined authorization and execution controls commensurate with their consequences. Commands that bypass normal operational safeguards require additional justification and restrictions appropriate to their emergency or contingency purpose. Backdoor, residual, hardware-level, test, diagnostic, or override commands that bypass normal operational pathways or safeguards should be retained only where mission-approved emergency or contingency access requires them. Their inclusion must be explicitly justified, and their commanding authority must be appropriately restricted. Hazardous commands required for normal mission operations are not prohibited by this countermeasure but must be governed by approved authorization, prerequisite, sequencing, and safety controls. Any command capability introduced by a subsystem supplier or component vendor that was not explicitly requested or authorized by the mission owner must be identified, evaluated, and either removed or brought under formal mission command governance	Functional	Intersects With	Static Code Analysis	TDA-17.1	Mechanisms exist to require the developers of Technology Assets, Applications and/or Services (TAAS) to employ static code analysis tools to identify and remediate common flaws and document the results of the analysis.	5		TDA-09.2
CM0043	Backdoor Commands	All commands capable of being executed on the spacecraft must be known, documented, and accounted for by the mission and spacecraft owner; the existence of undisclosed or undocumented commands, whether introduced by developers, component suppliers, or subsystem vendors, represents an unacceptable and unmanageable risk to mission integrity. Commands capable of adversely affecting mission success if misused must be identified through deliberate analysis and protected by mission-defined authorization and execution controls commensurate with their consequences. Commands that bypass normal operational safeguards require additional justification and restrictions appropriate to their emergency or contingency purpose. Backdoor, residual, hardware-level, test, diagnostic, or override commands that bypass normal operational pathways or safeguards should be retained only where mission-approved emergency or contingency access requires them. Their inclusion must be explicitly justified, and their commanding authority must be appropriately restricted. Hazardous commands required for normal mission operations are not prohibited by this countermeasure but must be governed by approved authorization, prerequisite, sequencing, and safety controls. Any command capability introduced by a subsystem supplier or component vendor that was not explicitly requested or authorized by the mission owner must be identified, evaluated, and either removed or brought under formal mission command governance	Functional	Intersects With	Dynamic Code Analysis	TDA-17.2	Mechanisms exist to require the developers of Technology Assets, Applications and/or Services (TAAS) to employ dynamic code analysis tools to identify and remediate common flaws and document the results of the analysis.	5		TDA-09.3
CM0031	Authentication	All command-bearing sessions, frames, or messages involving spacecraft command links, crosslinks, or relay services shall provide cryptographic authentication of the command origin and integrity verification before commands are accepted. Mutual or bidirectional authentication shall be required where both endpoints must authenticate one another and the link and protocol architecture support that exchange. Acquisition requirements should mandate cryptographically based, bidirectional authentication for all command sessions across external links, including ground-to-spacecraft uplinks, spacecraft-to-spacecraft crosslinks, and any relay or intermediary ground station connections, with authentication required aBidirectional authentication enables both communicating entities to verify each other's identity and helps prevent impersonation. Authentication establishes identity but does not by itself authorize a command, protect mission-data confidentiality, or prevent session hijacking. Command acceptance must also enforce authorization, and authenticated sessions or security associations must maintain integrity and replay resistance so that subsequent traffic remains bound to the authenticated entities. Beyond external links, authentication is strongly recommended for spacecraft internal bus communications and onboard inter-component connections, as an adversary with access to internal interfaces, whether through a compromised component or a physical access event, should face the same authentication barrier as an external adversary attempting to inject commands from outside the spacecraft.	Functional	Intersects With	Authenticate, Authorize and Audit (AAA)	IAC-03	Mechanisms exist to strictly govern the use of Authenticate, Authorize and Audit (AAA) solutions, both on-premises and those hosted by an External Service Provider (ESP).	5		IAC-01.2
CM0031	Authentication	All command-bearing sessions, frames, or messages involving spacecraft command links, crosslinks, or relay services shall provide cryptographic authentication of the command origin and integrity verification before commands are accepted. Mutual or bidirectional authentication shall be required where both endpoints must authenticate one another and the link and protocol architecture support that exchange. Acquisition requirements should mandate cryptographically based, bidirectional authentication for all command sessions across external links, including ground-to-spacecraft uplinks, spacecraft-to-spacecraft crosslinks, and any relay or intermediary ground station connections, with authentication required aBidirectional authentication enables both communicating entities to verify each other's identity and helps prevent impersonation. Authentication establishes identity but does not by itself authorize a command, protect mission-data confidentiality, or prevent session hijacking. Command acceptance must also enforce authorization, and authenticated sessions or security associations must maintain integrity and replay resistance so that subsequent traffic remains bound to the authenticated entities. Beyond external links, authentication is strongly recommended for spacecraft internal bus communications and onboard inter-component connections, as an adversary with access to internal interfaces, whether through a compromised component or a physical access event, should face the same authentication barrier as an external adversary attempting to inject commands from outside the spacecraft.	Functional	Intersects With	Identification & Authentication for Organizational Users	IAC-32	Mechanisms exist to uniquely identify and centrally Authenticate, Authorize and Audit (AAA) organizational users and processes acting on behalf of organizational users.	5		IAC-02

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0089	Assessment & Authorization	Assessment and authorization (A&A) is a structured, formal process through which an organization evaluates the extent to which a system's design and implementation satisfy a defined set of security requirements, and grants or denies authorization for that system to operate based on the assessed risk. For space mission systems, A&A may apply to spacecraft, ground systems, mission networks, supporting infrastructure, common controls, and the interfaces and dependencies among them. The authorization scope and boundary must be defined by the governing risk management framework, mission architecture, information types, applicable requirements, and organizational risk decisions. The assessment phase produces evidence concerning whether selected controls are implemented correctly, operating as intended, and producing the required security outcomes. The resulting authorization package should contain the system security plan, assessment reports, plan of action and milestones, executive risk summary, and other evidence required by the authorizing authority. The authorization decision, made by a designated authority with accountability for accepting the residual risk of operating the system, formally records the organization's acceptance of that risk and establishes the conditions under which the system may operate. Authorization must be supported throughout the system lifecycle by continuous monitoring, security impact analysis, updated risk information, and maintenance of the authorization evidence. Proposed system changes must be assessed before implementation when they could affect the authorization boundary, control implementation, or accepted risk. Significant changes or material deviations from the authorization basis must be reported to the authorizing authority, who determines whether additional assessment, modified authorization conditions, or reauthorization is required.	Functional	Intersects With	Information Assurance (IA) Operations	IAO-02	Mechanisms exist to facilitate the implementation of security, compliance and resilience assessment and authorization controls.	5		IAO-01
CM0027	ASIC/FPGA Manufacturing	Custom application-specific integrated circuits (ASICs) should be fabricated through accredited trusted foundries, and field-programmable gate array (FPGA) devices should be procured through trusted suppliers with documented fabrication provenance, to reduce the risk of hardware Trojan insertion or unauthorized modification. Unlike software, hardware trojans embedded during semiconductor manufacturing are extremely difficult to detect through functional testing alone, as they may be designed to activate only under specific operational conditions or remain dormant indefinitely; the integrity of the fabrication source is therefore a primary defense. Trusted foundry accreditation provides assurance that the accredited fabrication activities are subject to security controls intended to reduce the risk of unauthorized modification. Assurance for design, intellectual property, aggregation, packaging, assembly, testing, and distribution must be addressed through trusted suppliers or other controls applicable to those lifecycle stages. This requirement applies to custom ASICs and to the base silicon used in FPGA implementations. The programmable design loaded onto an FPGA requires separate protection because trusted fabrication of the device does not establish the integrity or authenticity of the configured bitstream.	Functional	Intersects With	Acquisition Strategies, Tools & Methods	TPM-03	Mechanisms exist to utilize tailored acquisition strategies, contract tools and procurement methods for the purchase of unique Technology Assets, Applications and/or Services (TAAS).	5		TPM-03.1
CM0027	ASIC/FPGA Manufacturing	Custom application-specific integrated circuits (ASICs) should be fabricated through accredited trusted foundries, and field-programmable gate array (FPGA) devices should be procured through trusted suppliers with documented fabrication provenance, to reduce the risk of hardware Trojan insertion or unauthorized modification. Unlike software, hardware trojans embedded during semiconductor manufacturing are extremely difficult to detect through functional testing alone, as they may be designed to activate only under specific operational conditions or remain dormant indefinitely; the integrity of the fabrication source is therefore a primary defense. Trusted foundry accreditation provides assurance that the accredited fabrication activities are subject to security controls intended to reduce the risk of unauthorized modification. Assurance for design, intellectual property, aggregation, packaging, assembly, testing, and distribution must be addressed through trusted suppliers or other controls applicable to those lifecycle stages. This requirement applies to custom ASICs and to the base silicon used in FPGA implementations. The programmable design loaded onto an FPGA requires separate protection because trusted fabrication of the device does not establish the integrity or authenticity of the configured bitstream.	Functional	Intersects With	Supply Chain Risk Management (SCRM)	TPM-04	Mechanisms exist to: (1) Evaluate security risks and threats associated with Technology Assets, Applications and/or Services (TAAS) supply chains; and (2) Take appropriate remediation actions to minimize the organization's exposure to those risks and threats, as necessary.	5		TPM-03
CM0024	Anti-counterfeit Hardware	Counterfeit electronic components represent a direct supply chain threat to space mission integrity, introducing hardware that may fail prematurely, perform outside specification, or contain malicious functionality deliberately embedded by an adversary during manufacture or distribution. A formal anti-counterfeit program must establish policy and procedures that span the entire component acquisition and integration lifecycle, from supplier qualification and procurement through incoming inspection, storage, and installation. The program must address two distinct but related risks: counterfeit components that fail to perform their intended function, degrading mission reliability; and deliberately tampered components that introduce malicious hardware functionality or create pathways for malicious code execution. Anti-counterfeit controls must include measures appropriate to component criticality and supply chain risk to authenticate components, detect evidence of tampering, and resist unauthorized modification. Detection and prevention must be treated as complementary objectives: prevention through qualified sourcing and procurement controls, detection through inspection and authentication techniques applied before components enter the	Functional	Intersects With	Anti-Counterfeit Training	SAT-06.4	Mechanisms exist to train personnel to detect counterfeit system components, including hardware, software and firmware.	5		TDA-11.1
CM0024	Anti-counterfeit Hardware	Counterfeit electronic components represent a direct supply chain threat to space mission integrity, introducing hardware that may fail prematurely, perform outside specification, or contain malicious functionality deliberately embedded by an adversary during manufacture or distribution. A formal anti-counterfeit program must establish policy and procedures that span the entire component acquisition and integration lifecycle, from supplier qualification and procurement through incoming inspection, storage, and installation. The program must address two distinct but related risks: counterfeit components that fail to perform their intended function, degrading mission reliability; and deliberately tampered components that introduce malicious hardware functionality or create pathways for malicious code execution. Anti-counterfeit controls must include measures appropriate to component criticality and supply chain risk to authenticate components, detect evidence of tampering, and resist unauthorized modification. Detection and prevention must be treated as complementary objectives: prevention through qualified sourcing and procurement controls, detection through inspection and authentication techniques applied before components enter the	Functional	Intersects With	Product Tampering and Counterfeiting (PTC)	TDA-09	Mechanisms exist to maintain awareness of component authenticity by developing and implementing Product Tampering and Counterfeiting (PTC) practices that include the means to detect and prevent counterfeit components.	5		TDA-11

NIST IR 8477-Based Set Theory Relationship Mapping (STRM)

Reference Document: Secure Controls Framework (SCF) version 2026.3
STRM Guidance: https://securecontrolsframework.com/start-here/set-theory-relationship-mapping-strm/

Focal Document: **SPARTA Countermeasure (4.0)**
Focal Document URL: https://sparta.aerospace.org/countermeasures/SPARTA
Published STRM URL: https://content.securecontrolsframework.com/strm/scf-strm-general-sparta-4-0.pdf

FDE #	FDE Name	Focal Document Element (FDE) Description	STRM Rationale	STRM Relationship	SCF Control	SCF #	Secure Controls Framework (SCF) Control Description	Strength of Relationship	Notes	Legacy SCF #
CM0083	Antenna Nulling and Adaptive Filtering	Antenna nulling and adaptive filtering are complementary electronic protection techniques that reduce the effects of jamming on spacecraft communication and sensing links while preserving access to legitimate signals. Antenna nulling dynamically adjusts the receive antenna pattern to reduce sensitivity in the estimated direction of arrival of a jammer. For terrestrial interference, the affected geographic region depends on the spacecraft's position and attitude, antenna geometry, and uncertainty in the jammer's location. Nulling is most effective against a limited number of discrete, detectable interference sources, but it may also attenuate legitimate signals arriving from the same or a nearby direction. Adaptive filtering suppresses interference based on its spectral or signal characteristics, such as by placing adaptive notches around narrowband or slowly varying interference. It can preserve operation within unaffected portions of the received bandwidth but may also remove or distort legitimate signal energy that overlaps the rejected frequencies. Its effectiveness decreases against wideband, rapidly changing, or multiple simultaneous jammers when too little usable bandwidth remains to support mission requirements. Used together, antenna nulling and adaptive filtering can address a broader range of jamming conditions than either technique alone. Both techniques depend on the interference remaining within the operating range of the antenna and receiver chain. If a jammer saturates or damages the low-noise amplifier, analog front end, or analog-to-digital converter, downstream digital processing may be unable to recover the legitimate signal. Receiver dynamic range and front-end protection must therefore be incorporated into the overall electronic protection	Functional	No Relationship	N/A	N/A	N/A	0		
CM0070	Alternate Communications Paths	Establishing alternate communications paths for spacecraft and ground system operations reduces the likelihood that a single adversarial event, physical disruption, or technical failure will deny all mission communications. Reliance on a single communications pathway creates a single point of failure that adversaries can exploit through jamming, denial of service against ground infrastructure, physical disruption of a ground station, or compromise of network connectivity, any of which could result in complete loss of commanding and telemetry capability. Alternate paths must be assessed end to end for shared failure modes and operational dependencies. Differences in frequency, ground station, relay service, or network provider provide meaningful resilience only when the paths do not remain dependent on the same critical spacecraft, ground, management, or service infrastructure. The selection and configuration of alternate paths must be governed by the mission's concept of operations (CONOPS), which defines the conditions under which each path is used, the priority and switchover procedures between paths, and the minimum communications capability that must be maintained to satisfy mission safety and operational continuity requirements.	Functional	Intersects With	Alternate Communications Channels	BCD-06.1	Mechanisms exist to maintain command and control capabilities via alternate communications channels and designating alternative decision makers if primary decision makers are unavailable during a designated crisis situation.	5		BCD-10.4