Conducting Cyber Tabletop Exercises for Space Systems Using SPARTA

March 20, 2026

Brandon T. Bailey
Cybersecurity and Advanced Platforms Subdivision, Information Systems and Cyber Division

AI Disclosure Statement: This report was developed with the assistance of generative AI tools that supported technical editor needs such as refining structure, adding clarity, and ensuring consistency of expert-derived content. All technical editor content was validated by subject matter experts. Models used: OpenAI o3-mini-high, Claude 4.5 Sonnet.

1. Introduction

1.1. Purpose

This Standard Operating Procedure (SOP) provides a repeatable approach for planning and conducting cyber tabletop exercises (TTX) for space systems.

A cyber tabletop exercise is a discussion-based scenario where participants work through a hypothetical cyber incident affecting a mission. The objective is to evaluate how teams detect, respond to, and recover from a cyber-attack while maintaining mission operations.

For space missions, cyber incidents can affect multiple segments of the system, including:

- **Space segment** (spacecraft, hosted payloads, and flight software)
- **Ground segment** (mission control systems and command infrastructure)
- **Link segment** (telemetry, tracking, and command communications)

This SOP uses the Space Attack Research and Tactic Analysis (SPARTA) framework as the primary structure for developing and evaluating scenarios. SPARTA is used throughout the exercise lifecycle to build attack scenarios, identify indicators of adversary behavior, evaluate countermeasures, and assess mission risk. Using SPARTA keeps the exercise grounded in realistic spacecraft attack techniques and produces findings that can be tied to defensive improvements.

Space systems involve cyber-physical interactions that are not well represented in traditional enterprise cybersecurity frameworks. SPARTA focuses on adversary behaviors that directly affect spacecraft subsystems, allowing tabletop exercises to evaluate attack paths against the system.

This SOP is intended to help organizations:

- Test detection and response capabilities against documented space cyber threats
- Validate mission protection procedures
- Identify gaps in operational playbooks and system defenses
- Improve coordination between engineering, operations, and cybersecurity teams

The goal is to translate exercise outcomes into concrete improvements that strengthen mission resilience.

1.2. Overview of SPARTA

SPARTA is a cybersecurity framework developed by The Aerospace Corporation to characterize how cyber threats apply to space systems. It organizes adversary behavior across the cyber-attack lifecycle in a way similar to MITRE ATT&CK but tailored for space systems.

At a high level, SPARTA breaks adversary behavior down into the following elements:

Component	Description
Tactics	High level adversary objectives during an attack
Techniques	Specific methods used to achieve those objectives
Sub-techniques	More detailed variations of a technique
Procedures	Realistic ways an adversary might implement a technique
Indicators of Behavior (IOBs)	Observable system behaviors that could indicate malicious activity
Countermeasures	Technical or operational defenses that mitigate the technique

SPARTA tactics generally follow a familiar adversary lifecycle:

SPARTA Tactic	Example Objective
Reconnaissance	Identify mission architecture and ground infrastructure
Initial Access	Gain access to mission systems
Execution	Execute malicious commands or software
Persistence	Maintain access over time across mission operations
Defense Evasion	Avoid detection by mission monitoring or security systems
Lateral Movement	Move between different mission components or systems
Exfiltration	Exfiltrate mission data or intellectual property
Impact	Disrupt spacecraft behavior or mission operations

SPARTA is useful for tabletop exercises because it provides space-specific attack scenarios that reflect how adversaries could realistically target spacecraft. For example, SPARTA includes techniques such as:

- IA-0006 Compromise Hosted Payload
- EX-0012 Modify On-Board Values
- LM-0002 Exploit Lack of Bus Segregation

Using SPARTA helps ensure tabletop scenarios are grounded in known or plausible adversary behavior, aligned with the actual spacecraft architecture, and traceable to specific defensive countermeasures.

1.3. Scope

This SOP is intended for organizations responsible for the design, operation, or defense of space systems, including government programs, civil space agencies, and commercial satellite operators and developers.

The process is designed to support tabletop exercises that address cyber threats across all mission segments:

Segment	Example Systems
Space Segment	Spacecraft avionics, flight software, and payload controllers
Ground Segment	Mission operations centers, command systems, engineering networks
Link Segment	Telemetry, tracking and command (TT&C), and crosslinks

Exercises can be conducted at different points in the mission lifecycle, including system design, pre-launch readiness, on-orbit operations, and anomaly response preparation.

1.4. Benefits of Conducting Space Cyber Tabletop Exercises

Cyber tabletop exercises provide several practical benefits for space missions.

- Improved threat awareness
  - Participants gain a better understanding of how adversaries could target space missions using specific attack techniques.
- Threat modeling grounded in operationally realistic attack scenarios
  - SPARTA provides a structured way to describe space system attack methods. Using it helps ensure tabletop exercises reflect realistic adversary behaviors instead of drifting into generic or hypothetical scenarios.
- Validation of response procedures
  - Exercises help determine whether existing procedures enable teams to detect and respond effectively to cyber incidents.
- Identification of system vulnerabilities
  - Exercise discussions often surface architectural weaknesses or gaps in monitoring and visibility.
- Cross team coordination
  - Space cyber incidents often involve multiple teams. Exercises help improve coordination between mission operators, cybersecurity analysts, spacecraft engineers, and decision-makers.
- Improved mission resilience
  - Insights from these exercises can be used to strengthen system protections, improve procedures, and increase overall mission resilience.

In practice, many of these benefits only become clear once teams work through a realistic scenario and see where assumptions break down.

1.5. Common Pitfalls in Space Cyber Tabletop Exercises

Organizations conducting tabletop exercises for the first time often encounter similar challenges. Understanding these pitfalls helps improve exercise effectiveness.

1.5.1. Scenarios That Are Too Generic or Oversimplified

A common issue in cyber tabletop exercises is the use of generic scenarios that lack real operational context. In some exercises, facilitators introduce scenarios with vague prompts like “you’ve been

hacked" or "there's a cyber incident." These prompts might get the conversation started, but they don't give enough technical or operational context to make the discussion meaningful.

These scenarios often end up looking like typical enterprise cybersecurity incidents, like phishing campaigns or general network compromises. While those scenarios might make sense for corporate IT environments, they don't reflect how an adversary would realistically target a space mission, its subsystems, or mission operations.

Effective tabletop exercises should instead be built around specific adversary behaviors and observable actions that could realistically occur in the mission. Scenarios should describe concrete events, like anomalous command activity, unexpected telemetry behavior, communication disruptions, or other suspicious system interactions. Providing participants with realistic indicators and system behaviors allows teams to work through how the incident would be detected, investigated, and mitigated within the context of actual mission operations.

Frameworks like SPARTA help address this by providing structured adversary tactics, techniques, and mission-relevant behaviors specific to space systems. By building exercise injects from SPARTA techniques and Indicators of Behavior, designers can create scenarios that reflect credible adversary actions against spacecraft and communication links. This approach moves the exercise away from abstract cyber events and toward realistic, threat-informed scenarios that better evaluate mission resilience and operational decision-making.

1.5.2. Lack of Mission Context

Participants will struggle to evaluate a cyber scenario if they don't understand how the mission operates. Before starting the scenario, facilitators should provide enough context for participants to understand the mission (e.g., this includes mission objectives, spacecraft capabilities, ground infrastructure, and operational timelines). Without this context, participants end up guessing. With it, they can evaluate how adversary actions would affect mission operations.

1.5.3. Overly Technical Discussions

Tabletop exercises lose value when discussions become overly focused on how a specific exploit would work. Participants often get pulled into debates about whether a vulnerability really exists, how an adversary would gain initial access, or whether a specific attack has ever been seen before.

Those discussions can be useful in other settings, but they're not the goal of a tabletop exercise. The purpose of a tabletop exercise is to evaluate operational response, decision making, and mission impact but not to prove whether a specific exploit path is technically achievable.

For this reason, tabletop exercises often rely on controlled assumptions, or "white card" injects, where facilitators introduce a condition that is assumed to be true for the purpose of the scenario. This allows participants to focus on how the organization would detect, respond to, and manage the operational consequences of the event.

By accepting these assumptions, teams can focus on questions like:

- How anomalies would be recognized
- What actions operators would take
- How mission operations would be affected
- Which teams would be responsible for response actions

Maintaining this balance keeps the exercise focused on what actually matters which is revealing gaps in procedures, coordination, monitoring, and mission resilience, rather than getting stuck trying to prove whether an exploit is feasible.

1.5.4. Lack of Clear Findings

Without structure, tabletop exercises often produce vague observations instead of actionable outcomes. Using structured methods such as NIST CSF categories and SPARTA technique mappings helps turn those observations into concrete improvements. When missing the structure, it's common to walk away from a tabletop with good discussion but no clear path forward.

1.5.5. Failure to Follow Up on Recommendations

A tabletop exercise only provides value if the findings lead to actual improvements. Recommended actions should be assigned to responsible teams, prioritized based on mission risk, and tracked through existing engineering or cybersecurity processes. Integrating exercise outcomes into mission security programs helps ensure those findings actually lead to measurable improvements.

1.6. Key Definitions

Cyber Tabletop Exercise (TTX): A facilitated discussion-based exercise where participants work through a simulated cyber incident to evaluate response procedures and decision making.

Tactics, Techniques, and Procedures (TTPs): A structured description of adversary behavior. Tactics describe objectives, techniques describe methods, and procedures describe specific implementations.

- Tactics represent the threat actor's tactical goal and the reason(s) they are performing a technique. For example, a threat actor may want to achieve initial access on a spacecraft via cyber means. Other tactics include reconnaissance, resource development, execution, defense evasion, exfiltration, lateral movement, and impact.
- Techniques represent "how" a threat actor achieves a tactical goal by performing a threat action. For example, a threat actor may exploit trusted relationships to achieve initial access. SPARTA maps a range of techniques that threat actors can use to execute tactics.

- Sub-techniques represent a variation or more specific instance of the threat actor's behavior used to achieve a goal. Sub-techniques typically describe behavior at lower levels than a technique and are considered children of the parent technique. For example, a threat actor may compromise mission collaborators (academia, international, etc.) to achieve their initial access.

Indicator of Behavior (IOB): Observable system activity that may indicate adversary behavior. Examples include unexpected command sequences or unusual telemetry patterns. Unlike traditional Indicators of Compromise (IOCs), which focus on detecting known malicious artifacts or signatures, IOBs are designed to identify behavioral patterns that could indicate emerging threats. This proactive approach allows space system developers to build detection mechanisms for anomalies and suspicious activities onboard the spacecraft, making IOBs a crucial element in modern spacecraft defense.

Countermeasure: According to NIST, countermeasures are defined as "protective measures prescribed to meet the security objectives, i.e., confidentiality, integrity, and availability, specified for an information system. Safeguards may include security features, management controls, personnel security, and security of physical structures, areas, and devices". In SPARTA's context, countermeasures represent security concepts and classes of technologies that can be used to prevent a technique or sub-technique from being successfully executed.

Attack Chain: A sequence of adversary actions used to achieve an attacker's objective, often spanning multiple TTPs from initial reconnaissance to achieving their final goal.

Inject: A piece of information introduced during a tabletop exercise to move the scenario forward. Example injects include anomaly reports, security alerts, and operator observations.

2. SPARTA-Driven Threat Modeling Lifecycle

SPARTA provides a structured way to analyze cyber threats to space systems across the full lifecycle of mission security activities. When used in tabletop exercises, SPARTA connects threat modeling, operational evaluation, and mission risk analysis into a single workflow. The lifecycle described in this SOP links mission architecture, adversary behaviors, operational response, and risk prioritization.

2.1. Step 1: Understand the Mission Architecture

Start by understanding the mission architecture and how the system is broken down across space, ground, link, and user segments. This includes identifying key systems such as spacecraft subsystems, ground stations, communication links, and supporting infrastructure. Without this context, it's difficult to tie adversary behavior to anything meaningful in the mission. SPARTA has the tool Spacecraft Functional Decomposition to enable this step by providing common spacecraft subsystems and components. In practice, this can be where many exercises fall short because teams jump into scenarios without a shared understanding of how the system actually operates.

2.2. Step 2: Map SPARTA Techniques to the Architecture

Once the architecture is understood, select SPARTA TTPs that are relevant to the mission. This step identifies how an adversary could interact with the system, which subsystems may be targeted, and which mission functions could be affected. The result is a set of adversary actions aligned with the actual mission design. SPARTA provides the Spacecraft Functional Decomposition tool, which maps techniques directly to common spacecraft subsystems and components. The goal is to focus on adversary behavior that makes sense for the system being evaluated. Not every TTP applies to every mission and forcing them in usually leads to unrealistic scenarios.

2.3. Step 3: Build a SPARTA Attack Chain

Mapped techniques are organized into an attack chain that represents how an adversary might progress through the mission environment. The attack chain follows the SPARTA adversary lifecycle, beginning with initial access and progressing through execution, persistence, and mission impact. Indicators of Behavior associated with each technique are identified so that participants can evaluate detection during the exercise. SPARTA has the tool SPARTA Navigator to enable this step by providing a GUI to build the attack chains and keep track of the mapped SPARTA countermeasures.

2.4. Step 4: Evaluate Detection and Response During the Tabletop Exercise

The tabletop exercise walks participants through the attack chain using scenario injects derived from SPARTA techniques and indicators. Participants evaluate how the mission would detect adversary behavior, what operational responses would occur, and how existing procedures or controls influence the outcome. Using the export feature in SPARTA Navigator, associated Indicators of Behavior (IOB) can be exported to help correlate adversary actions with observable system behaviors.

2.5. Step 5: Assess Countermeasures and Mission Impact

Following the scenario, the exercise examines how implemented countermeasures, or lack thereof, affected the attack progression. Participants evaluate whether defensive controls prevented the attack, enabled detection, or limited mission impact. This step helps determine which protections provide meaningful resilience improvements. This is where having the technique to countermeasure mapping helps as participants can analyze the presence of countermeasures already identified by SPARTA to help mitigate the appropriate techniques.

2.6. Step 6: Prioritize Risk Using SPARTA NRS

Finally, the exercise team evaluates the mission risk associated with each technique using the SPARTA Notional Risk Score (NRS) combined with the mission impact observed during the scenario. This allows organizations to prioritize defensive improvements based on adversary behaviors that present the greatest risk to mission operations.

This lifecycle ensures that tabletop exercises produce outcomes that are directly connected to real mission-relevant adversary behaviors, system architecture, and mission objectives. By linking threat modeling, operational evaluation, and risk prioritization, SPARTA enables organizations to move from theoretical threat discussions to practical security improvements for space systems. The preparation phase operationalizes the SPARTA threat modeling lifecycle described above.

3. Preparation Phase

Preparation is where most of these exercises either succeed or fall apart. If the scope isn't clear and the right people aren't in the room, you'll end up with a scenario that looks good on paper but doesn't reflect how the mission actually operates. During preparation, SPARTA is used to construct the adversary attack chain, identify indicators of behavior, and evaluate defensive countermeasures that may influence mission outcomes.

For most missions, preparation should begin two to four weeks before the exercise.

The major steps in preparation are:

1. Define exercise objectives
2. Select participants and assign roles
3. Develop scenarios using SPARTA TTPs
4. Gather supporting materials
5. Plan exercise logistics
6. Conduct a facilitator dry run

3.1. Define Exercise Objectives

Start by being clear on what you want to learn from the exercise. Keep the objectives tight and specific. If you try to evaluate everything at once, you'll end up with vague results that aren't actionable. Don't try to test everything because that's one of the fastest ways to end up with vague, unusable results.

Most teams usually focus on things like:

- Evaluate the ability to detect adversary activity using Indicators of Behavior (IOBs)
- Validate incident response procedures for space mission cyber events
- Test coordination between mission operations and cybersecurity teams
- Assess effectiveness of existing countermeasures/controls
- Identify monitoring gaps within spacecraft or ground systems
- Evaluate decision making during mission impacting cyber events

Write them in plain language. If participants have to interpret what the objective means, you've already lost clarity. Also be realistic because you're not going to test every layer of the system in one exercise. Pick a few things that matter to the mission.

Example objective statements:

Objective	Description
Detection evaluation	Determine whether teams recognize SPARTA technique indicators
Response validation	Evaluate how operators respond to spacecraft cyber anomalies
Communication coordination	Assess coordination between operations, engineering, and cyber teams
Mission protection	Evaluate ability to maintain mission functionality during cyber events

Keep it to three to five objectives max otherwise scope creep will take over the exercise. As you define objectives, it helps to anchor them to realistic attacker goals, so the analysis doesn't drift. These represent the kinds of outcomes a real adversary would actually be trying to achieve.

3.1.1. Sample Attacker Objectives

To characterize the attacker's primary goals against a space system, we define the following seven attacker objectives.

OBJ-1: Deny Ground Telemetry and Mission Data Processing
This objective targets the ground segment's ability to process telemetry and mission data transmitted from the space vehicle. Telemetry provides critical health and status information about spacecraft subsystems, orbital parameters, and mission payload performance. Mission data is provided by mission payloads on the spacecraft such as satellite communications, imaging, or other sensing data. By denying processing capabilities, adversaries prevent operators from monitoring spacecraft health, detecting anomalies, storing data, or making informed decisions about mission execution.

OBJ-2: Deny All Communications with the Space Vehicle
This objective seeks to deny the bidirectional communication link between ground infrastructure and the orbiting asset. This objective accomplishes a comprehensive communication blackout affecting both uplink and downlink channels for commanding, telemetry, and receipt of mission data. This objective can be viewed as the denial of the link segment capabilities that reside within the ground segment

OBJ-3: Execute Commands on the Space Vehicle
This objective represents an attacker accomplishing command authority over the spacecraft. This ability enables an adversary to directly manipulate vehicle behavior without legitimate operator consent. Attackers achieving this objective can issue commands such as altering spacecraft orientation, depleting propellant reserves, modifying payload configurations, or triggering destructive sequences.

OBJ-4: Deny Ground Commanding
This denial objective against the satellite uplink prevents legitimate operators from transmitting commands to the spacecraft but downlink telemetry is still possible. This attack creates a command embargo that maintains spacecraft in its last known configuration. This objective can

be utilized when an attacker desires to falsify the legitimate operator's commanding of the spacecraft (e.g., man-in-the-middle attack)

OBJ-5: Compromise Mission Data Integrity
This objective focuses on subtle manipulation that targets the trustworthiness of mission data. Adversaries achieving this objective can inject false telemetry, modify payload data, alter historical mission records, or corrupt data processing algorithms, all while maintaining the appearance of normal operations.

OBJ-6: Degrade Confidence in System Health Tools
This psychological and operational objective undermines operator trust in the monitoring and diagnostic systems used to assess spacecraft and ground segment health. Rather than directly attacking mission functions, the adversary introduces errors such as inconsistencies, false alarms, or contradictory status indicators that erode confidence in system health assessments.

OBJ-7: Exfiltrate Mission Data
This intelligence-focused objective seeks unauthorized access to and extraction of sensitive mission data, including spacecraft telemetry, payload products, operational plans, system designs, cryptographic materials, or proprietary mission analysis. Unlike denial or manipulation objectives, exfiltration may leave minimal operational traces while providing adversaries with strategic intelligence for future operations, competitive advantage, or public disclosure.

3.2. Select Participants and Roles

You need the people who actually run and protect the mission in the room and not just observers. Get the right mix of people in the room (e.g., operators, engineers, cyber folks, and at least one person who understands how decisions get made during operations). The group should mirror who would respond during a real incident, otherwise the discussion won't translate to operations.

Recommended participant categories include:

- Mission operations personnel
- Spacecraft subsystem engineers
- Cybersecurity analysts
- Mission leadership or program managers
- Communications or public affairs representatives if relevant

To keep things from getting chaotic, you'll need to assign a few key roles upfront.

Role	Responsibility
Facilitator	Guides the exercise, presents scenario injects, manages time
Red Team Simulator	Represents the adversary and explains attack progression
Blue Team Participants	Respond to the scenario as they would in real operations
Technical Advisor	Provides clarification on system architecture

Recorder	Documents discussion, decisions, and observations
Observers	Watch exercise execution and capture lessons learned

The facilitator should remain neutral and avoid influencing participant decisions. The red team simulator introduces adversary activity based on the scenario. Leverage SPARTA / TTP frameworks as a basis for attack. Observers are usually cyber engineers or leadership who are there to capture what worked and what didn't. Most exercises land somewhere around 10 to 20 people which is enough for coverage, not so many that it slows everything down. You don't need a huge group, but you need the **right** group.

3.3. Gather Exercise Materials

Before the exercise, pull together the materials participants will need to stay grounded in the scenario.

This usually includes:

- Mission architecture diagrams
- Spacecraft subsystem descriptions
- SPARTA technique matrix
- Response playbooks
- Incident response procedures
- Exercise scenario guide
- Inject timeline

These give participants enough context to make decisions that reflect the system. If possible, send these out a few days ahead of time so people aren't seeing everything for the first time.

3.4. Plan Logistics

Most tabletop exercises run about three to four hours which is long enough to get meaningful discussion, but short enough to keep people engaged.

Recommended schedule:

- Introduction and objectives -- 30 minutes
- Scenario execution -- 90 to 120 minutes
- After action discussion -- 60 minutes

Exercises may be conducted in person (ideal), via video conference, or using hybrid participation

Tools may include whiteboards, collaboration software (Zoom, Teams, etc.), shared document workspace, and presentation slides for injects.

The facilitator should ensure all participants can clearly see scenario materials.

3.5. Develop Scenarios Using SPARTA

Exercise scenarios should be built from realistic adversary attack chains using TTP frameworks like SPARTA, rather than generic cyber scenarios that don't map to the mission. For spacecraft-focused tabletop exercises, SPARTA tactics and techniques should be used. The intent is to walk through how an adversary would realistically move through the mission architecture based on how the system actually operates.

Scenario development is typically structured using the following steps.

Step 1. Identify the Mission and Architecture

Before mapping SPARTA techniques to system components, it is important to understand how the mission actually operates and which functions are critical to mission success. Cyber-attacks against space systems ultimately matter because of their impact on mission operations, spacecraft safety, or mission data.

This step examines the mission from both the operational workflow perspective and the system architecture that supports those workflows.

Step 1.1 Identify Mission Threads

Mission threads describe the operational workflows required to perform mission activities, not just system interactions. A mission thread represents the sequence of actions, systems, and data flows required to accomplish a mission task.

Examples of mission threads include:

Mission Thread	Description
Spacecraft Commanding	Generate commands, uplink to spacecraft, spacecraft executes commands
Telemetry Monitoring	Receive spacecraft telemetry, process health data, monitor system status
Payload Data Collection	Payload captures mission data and stores or downlinks results
Orbit Maintenance	Operators command propulsion or attitude adjustments to maintain orbit
Mission Planning	Mission tasking and scheduling of spacecraft activities

Understanding mission threads helps identify where cyber-attacks could disrupt mission operations.

Step 1.2 Identify Critical Mission Functions

Once mission threads are understood, identify the functions that are critical for mission success. These functions represent capabilities that must remain available for the mission to operate safely and effectively without degradation.

Examples of critical mission functions include:

Mission Function	Supporting Subsystem
Maintain communications link	Communications subsystem

Execute spacecraft commands	Command and Data Handling (C&DH)
Maintain spacecraft orientation	Attitude Determination and Control System (ADCS)
Process mission data	Payload and flight software
Monitor spacecraft health	Telemetry processing systems

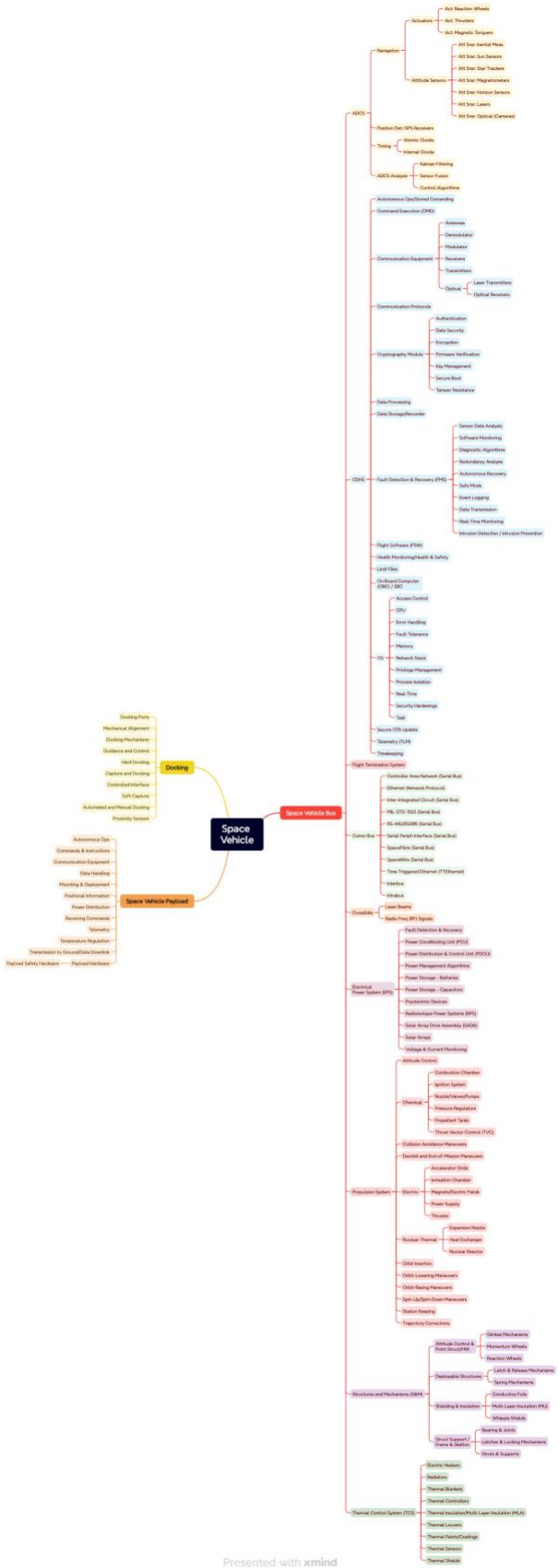
Mapping adversary techniques to critical mission functions ensures the tabletop exercise focuses on scenarios that could meaningfully affect mission operations.

Step 1.3 Identify the Mission Architecture

After identifying mission threads and critical mission functions, document the lower-level system components that supports those activities. Detailed architectural understanding improves the realism of the tabletop scenario and helps identify where adversary techniques may occur.

SPARTA provides a Spacecraft Functional Decomposition resource that can assist with this step by identifying common spacecraft subsystems and components. https://sparta.aerospace.org/sc-decomp

Here are some sample breakdowns for ground and spacecraft. The ground breakdown will be added in SPARTA 4.0.



* https://sparta.aerospace.org/sc-decomp

The output of Step 1 should include mission thread descriptions, identified critical mission functions, and system architecture diagrams with subsystem decomposition These artifacts provide the foundation for mapping SPARTA adversary techniques to the mission system.

SPARTA provides a structured method for building attack scenarios by linking adversary techniques to spacecraft subsystems, mission architecture, and observable system behaviors. The following steps demonstrate how SPARTA techniques can be mapped to a mission system to construct a realistic attack chain.

Step 2. Map SPARTA TTPs to the Spacecraft Architecture

After identifying the mission architecture/components in Step 1, the next step is to map SPARTA tactics and techniques to the specific spacecraft components that could be targeted. SPARTA tactics represent stages of the adversary attack lifecycle and are used to organize the progression of the tabletop scenario. This step ensures the tabletop scenario reflects how adversaries would interact with the actual spacecraft and mission systems.

The process should follow five sub steps.

Step 2.1 Identify High Risk SPARTA Techniques

Begin by selecting SPARTA techniques that represent high impact or high likelihood threats.

SPARTA provides a Notional Risk Score (NRS) that helps prioritize techniques most likely to affect space vehicles.

For spacecraft focused exercises, techniques are typically selected from the following tactics:

SPARTA Tactic	Reason for Focus
Execution	Techniques that directly affect spacecraft software or hardware
Persistence	Techniques that allow long term access to the vehicle
Defense Evasion	Methods that allow an attacker to remain undetected
Lateral Movement	Techniques used to pivot between subsystems
Exfiltration	Data extraction from spacecraft or payload

Techniques from Reconnaissance, Resource Development, Initial Access, and Impact may still appear in scenarios, but the highest risk technical testing typically occurs within the spacecraft execution phases.

SPARTA tactics represent stages of an adversary attack lifecycle and can be used to structure the progression of a tabletop scenario. During an exercise, injects typically follow the sequence of tactics as the adversary moves from initial access to mission impact.

Example high priority techniques commonly used in exercises:

Technique ID	Technique Name
EX-0001.01	Replay Command Packets
EX-0009.01	Flight Software Exploitation
EX-0009.03	Exploiting Known Vulnerability (COTS/FOSS)
EX-0012	Modify On Board Values
EX-0013	Command Flooding
EX-0016	Jamming

These techniques represent realistic spacecraft attack behaviors.

Step 2.2 Decompose the Space Vehicle

If not already done in step 1 during architecture definition, the spacecraft must be broken down into functional elements that support the mission.

This decomposition helps identify where adversary techniques could be executed.

Typical spacecraft decomposition includes subsystems such as:

Subsystem	Description
Command and Data Handling (C&DH)	Flight computer, command processing, memory management
Flight Software (FSW)	Software controlling spacecraft functions
Communications	Telemetry, tracking, and command systems
Attitude Determination and Control System (ADCS)	Controls spacecraft orientation
Electrical Power System (EPS)	Generates and distributes spacecraft power
Payload Systems	Mission sensors or instruments
Thermal Control System (TCS)	Maintains spacecraft temperature

The decomposition may also include specific functions or services, for example command authentication, telemetry encoding, timekeeping, reaction wheel control, and/or payload data processing.

This step helps determine which spacecraft components are most critical to mission operations.

Example mission critical function:

Mission Function	Supporting Subsystem
Maintain communications link	Communications subsystem
Maintain spacecraft orientation	ADCS
Process payload data	Payload and C&DH

Step 2.3 Enumerate the Attack Surface

Once components are identified, enumerate the attack surface for each component.

The attack surface includes all inputs, interfaces, and data flows that interact with the component.

Key questions to answer include:

- What inputs does the component receive?
- What outputs does it produce?
- What interfaces connect to the component?
- What external dependencies exist?

Example attack surface enumeration:

Component	Inputs	Interfaces
Flight Computer (C&DH)	Command packets, telemetry inputs	Spacecraft data bus
Flight Software	Command sequences, configuration parameters	Memory interface
ADCS	Torque commands, attitude data	Reaction wheel interface
Communications subsystem	Uplink commands, downlink telemetry	RF link

Understanding these data flows helps determine which SPARTA techniques are feasible.

Step 2.4 Map SPARTA Techniques to Subsystems

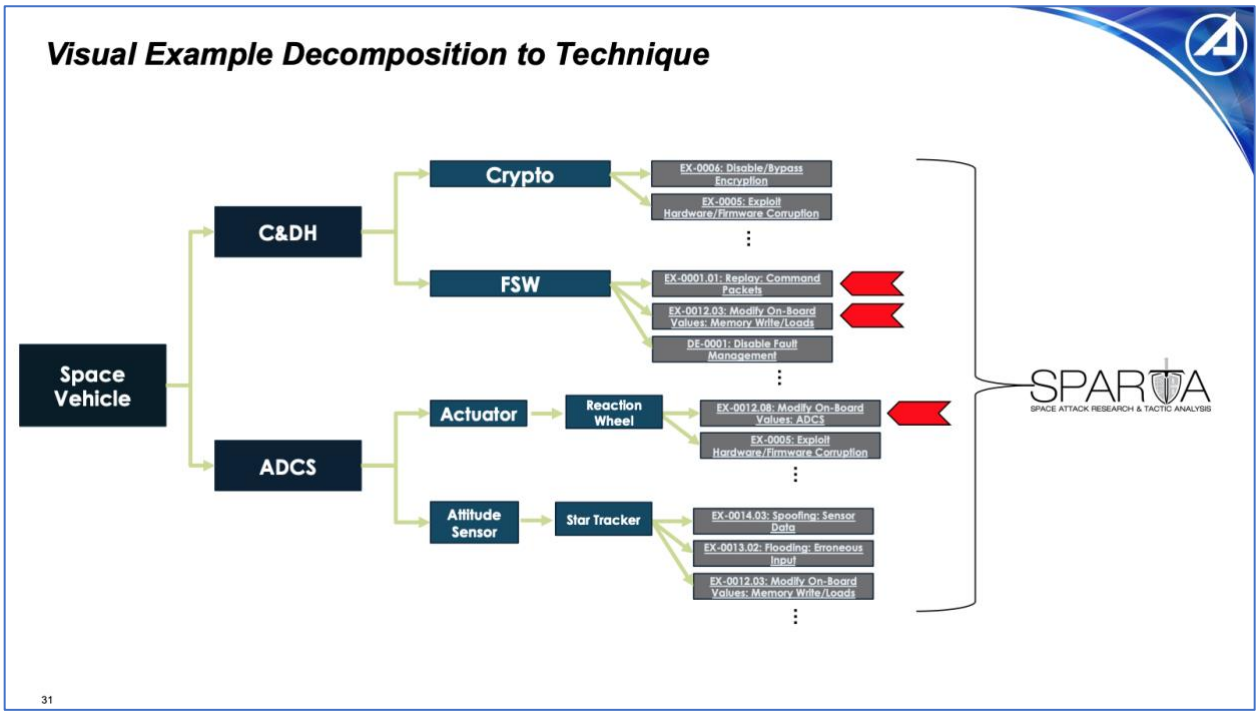
Using the decomposition and attack surface analysis, map SPARTA techniques to the subsystem where they could realistically occur.

Example mapping:

Subsystem	SPARTA Technique	Example Attack Action
Communications	EX-0001.01 Replay Command Packets	Adversary replays previously captured command packets
C&DH / FSW	EX-0012 Modify On-Board Values	Adversary alters configuration parameters in memory
ADCS	EX-0012.08 Modify On-Board Values: Attitude Determination & Control Subsystem	Reaction wheel torque commands manipulated
Communications	EX-0013 Command Flooding	Command interface overwhelmed with valid commands
Flight Software	EX-0009.03 Exploit Known Vulnerability	Exploit vulnerability in COTS flight software library

This mapping connects SPARTA techniques directly to spacecraft components.

Example from slide 31: https://sparta.aerospace.org/resources/OTR202400438_SPARTA_for_SV_Assessments.pdf



Step 2.5 Construct the Attack Chain

Finally, build a scenario that moves through these components.

Example spacecraft attack chain for a tabletop exercise:

Stage	Target Component	SPARTA Technique	Expected Effect
Initial Access	Ground command infrastructure	IA-0007	Unauthorized command access
Execution	Communications subsystem	EX-0001.01	Replayed command traffic accepted
Execution	C&DH / Flight Software	EX-0012	Configuration parameters changed
Impact	ADCS subsystem	EX-0012.08	Reaction wheel torque increased
Mission Effect	Spacecraft operations	Attitude instability	Mission degraded

Participants should evaluate how the anomaly would be detected, which telemetry or indicators reveal the attack, what operational response should occur, and which countermeasures could prevent the attack.

Step 2.6 Identify Indicators of Behavior

For each stage of the attack chain, define Indicators of Behavior that participants would observe. SPARTA contains several potential indicators, which are already mapped to techniques within the tool.

Technique	Possible Indicators	Sample SPARTA IOB
Replay Command Packets	Duplicate command timestamps	UACE-7: Duplicate Command Packet Executions UACE-20: Repeated Downlink Commands with Identical Timestamps CSNE-28:Detection of CANBus Replay Attack with Duplicate Message ID and Timing Anomalies
Modify On Board Values	Unexpected configuration changes	MIRE-7:Unexpected Memory Value Write or Modification DISE-17: Unauthorized Modification of Critical Onboard Values
Command Flooding	Command queue overflow alerts	UACE-9: Valid Command Flooding UACE-10: Logs of Processed Commands Flooding CSNE-12: Generic Flooding Attack
ADCS manipulation	Unexpected attitude adjustments	SMSR-4:ADCS Onboard Values Manipulation SMSR-5:Unexpected Spacecraft Telemetry or Movement Detected on Attitude DISE-7: Attitude Sensor Data and Actuator Behavior

These indicators become exercise injects during the tabletop.

The output of Step 2 with sub-steps should be:

- Spacecraft subsystem decomposition
- Attack surface enumeration
- SPARTA technique mapping
- Realistic attack chain
- Indicators of behavior for each stage

These artifacts form the basis of the tabletop scenario.

Step 3. Map SPARTA Countermeasures to the Attack Chain

One advantage of using SPARTA during tabletop exercises is that each technique in the framework is associated with recommended countermeasures. These countermeasures represent architectural controls, software protections, operational procedures, or monitoring capabilities that reduce the likelihood or impact of an attack. SPARTA countermeasures are used by exercise planners to understand existing protections and identify gaps. They are not intended to constrain participant responses during the tabletop.

During scenario development, planners should identify the countermeasures associated with each technique in the attack chain and determine whether those countermeasures are present in the system being exercised.

This step supports two important objectives.

- First, it helps determine whether the system currently has defenses capable of preventing or detecting the attack.
- Second, it allows the exercise to evaluate how the outcome might change if specific countermeasures were implemented.

This approach aligns with the method described in the SPARTA Space Vehicle assessment process where techniques are evaluated alongside their corresponding defensive mitigations.

Step 3.1 Identify Countermeasures for Each Technique

For each SPARTA technique selected in the attack chain, review the SPARTA countermeasure mapping and identify the recommended protections. The protections are not restricted to only countermeasures listed in SPARTA, but the default set in SPARTA provide a solid starting point.

These countermeasures may include technical, architectural, or operational safeguards and should be evaluated against the actual system architecture

Example mapping:

Technique	Spacecraft Component	Possible Countermeasure
Compromise Hosted Payload (IA-0006)	Payload	Payload isolation architecture (CM0038: Segmentation)
Modify On Board Values (EX-0012)	C&DH > CMD | FSW	Command validation (CM0055: Secure Command Mode(s))
Ground access compromise (IA-0007)	C&DH > CMD	Multi factor authentication (CM0055: Secure Command Mode(s) & CM0031: Authentication)

Step 3.2 Determine Countermeasure Presence

Exercise planners should determine whether the identified countermeasures are implemented in the system.

Three states should be captured.

Countermeasure Status	Description
Implemented	Countermeasure exists and is operational
Partially Implemented	Countermeasure exists but may be incomplete or untested
Not Implemented	Countermeasure is not present in the system

Example evaluation:

- CM0038: Segmentation - Partially Implemented
- CM0055: Secure Command Mode(s) - Not Implemented
- CM0031: Authentication – Implemented

This analysis helps identify defensive gaps before the exercise begins.

Step 3.3 Evaluate Detection Using Indicators of Behavior

Countermeasures may either prevent the attack or detect it after it occurs.

During the tabletop exercise, participants should evaluate whether the system would detect the attack using available telemetry, logs, or monitoring tools.

IOBs associated with the technique provide cues for detection.

Example:

Technique	Countermeasure	Indicator
Replay Command Packets	Command authentication	Duplicate command timestamps
Modify On Board Values	Memory integrity monitoring	Unexpected configuration changes
Command Flooding	Rate limiting	Command queue overflow alerts
ADCS Manipulation	Attitude anomaly detection	Unexpected spacecraft motion

Participants should discuss whether the indicator would be visible to operators, which monitoring systems would detect it, and how quickly it would be recognized.

Step 3.4 Evaluate Alternative Outcomes Using Countermeasures

A valuable exercise activity is evaluating how the mission outcome would change if countermeasures were present or improved. This allows participants to explore how architectural improvements could affect mission resilience.

Example scenario discussion:

Scenario Condition	Expected Outcome
No command authentication	Replay attack succeeds
Command authentication enabled	Replay attack rejected
Memory protection absent	On board configuration modified
Memory protection implemented	Unauthorized modification blocked

Participants should discuss how the presence or absence of specific protections affects mission impact. This allows organizations to assess which countermeasures provide the greatest resilience improvements.

Step 3.5 Use Countermeasure Analysis to Guide Recommendations

Insights from this analysis should feed directly into recommendations developed during the After Action Review.

Typical outcomes include:

- New system requirements.
    - Using SPARTA countermeasures accelerates this as 500+ requirements exist mapped to existing SPARTA countermeasures
- Improvements to spacecraft monitoring capabilities
- Updates to operational response procedures
- Implementation of additional SPARTA countermeasures

By linking techniques to countermeasures, the exercise helps move from threat discussion to actionable improvements in system security.

The output of Step 3 with sub-steps should be:

- Mapped set of SPARTA techniques
- Associated countermeasures for each technique
- Assessment of whether those countermeasures exist in the system
- Indicators of behavior for detection
- Potential alternative outcomes if countermeasures were implemented

These artifacts help ensure the tabletop exercise produces actionable insights about mission security.

3.6. Preparation Checklist

The following checklist can be used to verify readiness.

Task	Completed
Exercise objectives defined	☐
Participants invited	☐
Roles assigned	☐
Indicators and injects prepared	☐
Architecture diagrams gathered	☐
Exercise materials distributed	☐
Logistics confirmed	☐
SPARTA attack chain developed	☐

4. Execution Phase

The execution phase is the live portion of the tabletop exercise where participants walk through the adversary scenario and discuss detection, response, and recovery actions. During execution, the facilitator progresses through the SPARTA-based attack chain developed during preparation. Each inject corresponds to a SPARTA technique and associated indicator of behavior, allowing participants to evaluate how adversary actions manifest within spacecraft telemetry, ground systems, or operational procedures.

The objective is to evaluate detection, response, and mission resilience against the scenario:

- How adversary actions would be detected
- How operators and cybersecurity teams would respond
- Whether existing procedures support mission resilience
- Where gaps exist in monitoring, response, or architecture

A typical tabletop execution lasts 90 to 120 minutes.

4.1. Exercise Introduction and Rules

The facilitator begins the exercise by briefing participants on the scenario and the exercise format.

Topics to cover include exercise objectives, participating teams, mission architecture overview, attacker objectives, and rules of engagement.

Participants should understand that the exercise is a discussion environment**,** not a technical penetration test.

The following ground rules are recommended where participants respond based on **current** procedures and system capabilities and discussion should reflect real operational decision making. The facilitator should also remind participants that the scenario is fictional but based on known or forecasted adversary behavior.

4.2. Present the Mission Context

Before introducing adversary activity, provide participants with the operational context of the mission.

This could include mission description, spacecraft overview, ground architecture, operational tempo, current mission phase

Example:

Mission Element	Example Description
Orbit	Low Earth Orbit
Mission	Earth observation
Ground Segment	Mission operations center and backup site
Spacecraft	Bus with imaging payload
Operations	Daily tasking and scheduled downlinks

Providing this context helps participants understand how the scenario fits into mission operations.

4.3. Introduce the Initial Scenario

The facilitator introduces the first scenario inject based on the attack chain developed during preparation.

The first inject typically corresponds to the earliest observable indicator of adversary activity.

Example inject: An operator notices repeated command acknowledgements with identical timestamps during a routine telemetry review.

This inject corresponds to the SPARTA technique: Replay Command Packets (EX-0001.01)

Associated indicator: UACE-7 Duplicate Command Packet Executions

Participants should discuss:

- Whether the activity would be detected
- Which monitoring systems would alert operators
- Which team would investigate the anomaly
- What initial response actions would occur

4.4. Progress Through the Attack Chain

The facilitator introduces additional injects that represent later stages of the adversary attack.

Each inject should correspond to a step in the attack chain defined during the preparation phase.

Example progression:

Stage	SPARTA Technique	Inject
Initial Access	IA-0007 Ground Access Compromise	Suspicious login to mission operations workstation
Execution	EX-0001.01 Replay Command Packets	Duplicate command timestamps observed
Execution	EX-0012 Modify on Board Values	Unexpected spacecraft configuration change
Impact	EX-0012.08 ADCS Manipulation	Spacecraft attitude anomaly

Participants discuss detection and response actions at each stage.

The facilitator should allow discussion but keep the exercise moving forward.

4.5. Evaluate Detection Using Indicators of Behavior

At each stage of the scenario, participants evaluate whether the attack would be detected.

Questions for discussion include:

- What telemetry or logs would reveal the activity
- Which monitoring systems would detect the behavior
- How quickly operators would recognize the anomaly
- Whether indicators might be mistaken for normal spacecraft behavior

This discussion helps identify monitoring gaps and potential improvements. Detection discussions should also reference the SPARTA IOB associated with the techniques used in the scenario.

4.6. Evaluate Response Actions

Participants then discuss how they would respond to the incident.

Topics should include escalation procedures, coordination between teams, spacecraft operational actions, and cyber incident investigation steps.

Example response actions may include:

- Isolating compromised ground systems
- Switching to backup command infrastructure
- Suspending spacecraft commanding
- Placing spacecraft into safe mode

Participants should reference existing procedures and playbooks when possible.

4.7. Evaluate Countermeasure Effectiveness

During the exercise, participants should first determine response actions based on existing procedures, system knowledge, and operational judgment. The objective is not to select predefined solutions, but to assess how the organization would realistically respond under current conditions.

SPARTA countermeasures should not be introduced as prescriptive answers during scenario execution. Instead, they serve as a structured reference to support post-discussion evaluation of whether appropriate protections, mitigations, and detection capabilities exist.

After participant discussion, the facilitator should use SPARTA countermeasures to:

- Identify gaps between proposed responses and expected protections
- Highlight missing preventive, detective, or corrective controls
- Assess whether system design aligns with threat-informed defenses

Key questions include:

- Does the system currently implement this countermeasure?
- Would the countermeasure prevent the attack?
- Would the countermeasure enable earlier detection?

Example discussion: If command authentication were enabled, replayed command packets would be rejected by the spacecraft command interface.

This discussion helps identify which architectural protections provide the greatest mission resilience.

4.8. Introduce Branching Scenarios

If time allows, the facilitator may introduce alternative conditions.

Examples include:

- Adversary achieves persistence
- Telemetry data is falsified
- Command uplink capability is degraded

These variations help explore how different attack paths affect mission outcomes.

4.9. Conclude the Scenario

Once the attack chain has reached its final stage, the facilitator concludes the scenario.

Participants should briefly summarize how the attack unfolded, what actions were taken, and whether the mission was affected.

This prepares participants for the After-Action Review.

At the end of the execution phase, the team should have identified:

- Detection capabilities for each stage of the attack
- Response procedures used by the mission team
- Effectiveness of implemented countermeasures
- Potential improvements to mission security

These findings are documented during the After-Action Review.

5. After Action Review (AAR)

The AAR is conducted immediately following the exercise. Its purpose is to capture observations, lessons learned, and recommended improvements identified during the tabletop exercise.

The review should focus on evaluating mission detection, response, and recovery capabilities against the adversary scenario that was exercised.

The AAR should produce clear, actionable outcomes that can improve system architecture, operational procedures, or monitoring capabilities. Because the exercise scenario was constructed using SPARTA techniques, the AAR can directly map observations back to the adversary behaviors and countermeasures represented in the framework.

A typical AAR lasts 60 to 90 minutes.

5.1. Conduct Structured Discussion Using the NIST Cybersecurity Framework

During the AAR, discussion should be organized using the NIST Cybersecurity Framework (CSF) functions. This structure provides a simple and widely understood structure for organizing discussion across key phases of a cyber incident lifecycle.

Participants should evaluate how well the mission performed across the following functional areas.

Using the NIST CSF functions helps organize the discussion into a set of logical security principles that are familiar to most cybersecurity and operations teams. Concepts such as detect, respond, and recover are widely understood and provide a simple structure for evaluating how the mission would handle a cyber incident. Organizations may optionally map findings to CSF categories and subcategories after the exercise if integration with formal cybersecurity programs is required.

For space missions, an additional factor should be considered: **Mission Impact**. Cyber incidents ultimately matter because of their effect on spacecraft safety, mission performance, or mission

data. Including mission impact ensures the discussion remains focused on operational consequences rather than only technical cybersecurity concerns.

Participants should therefore evaluate the scenario across the following core functions + mission impact:

- Identify
- Protect
- Detect
- Respond
- Recover
- Mission Impact

This structure helps ensure the exercise examines the full lifecycle of a cyber incident while also evaluating how adversary actions affect mission success.

This approach helps participants systematically consider:

- How the attack would be identified
- What protections exist to prevent it
- How teams would respond operationally
- How the mission is impacted and would recover

5.1.1. Identify

This discussion area focuses on understanding mission assets, system dependencies, and mission impact.

- Which mission systems or subsystems were affected by the attack?
- Were the affected components identified as mission critical during system design?
- Did participants understand the dependencies between ground, link, and space systems?
- At what stage of the attack would mission functionality begin to degrade?

This discussion helps determine whether the mission architecture and risk assessments accurately capture cyber dependencies.

5.1.2. Protect

This discussion focuses on the protections and safeguards that exist to prevent adversary actions.

- Which SPARTA countermeasures were implemented in the system?
- Which protections successfully prevented or limited adversary actions?
- Which system protections were missing or insufficient?

- Are configuration management protections in place to prevent unauthorized modifications?

This discussion helps identify architectural improvements that may reduce risk.

5.1.3. Detect

This discussion area evaluates the mission's ability to detect adversary activity.

- Were the indicators of behavior noticeable to operators?
- Would monitoring tools, logs, or telemetry reveal the activity? (e.g., do we have the right telemetry to even detect?)
- Which monitoring systems (e.g., on-board IDS) would detect the anomaly?
- How quickly could the anomaly be identified?
- Would the indicators be recognized as a cyber incident or mistaken for a system malfunction?

This discussion helps identify gaps in monitoring, telemetry visibility, or anomaly detection.

5.1.4. Respond

This discussion area focuses on the mission team's operational response to the incident.

- Did existing procedures provide clear guidance for responding to the anomaly?
- Were responsibilities between mission operations, cybersecurity, and engineering teams clearly understood?
- Were escalation paths effective?
- Were response actions clearly defined for spacecraft cyber incidents?
- Could the team or the system autonomously isolate compromised systems or restrict adversary access?
- What's the turnaround time from detection to response? Human intervention required? Autonomy?

This discussion helps determine whether operational procedures and playbooks are sufficient.

5.1.5. Recover

This discussion area evaluates how the mission would recover after the incident.

Participants should discuss:

- What actions would be required to restore mission functionality?
- Could the spacecraft be returned to a safe operational state? Does a cyber-safe mode exist?

- How long would recovery take?
- Would system updates, configuration resets, or software patches be required?
- Are contingency procedures available if the primary command system is compromised?

This discussion helps evaluate the mission's resilience and ability to return to normal operations.

5.1.6. Mission Impact

This discussion area focuses on how the cyber incident effects mission operations and spacecraft safety.

- At what stage of the attack would the mission begin to experience operational impact? (e.g., what mission thread does it disrupt)
- Could mission operations continue during the incident?
- Would spacecraft safety be at risk?
- Would payload performance or mission data quality be affected?
- Would operators lose the ability to command or monitor the spacecraft?

This discussion helps determine whether the mission architecture and operational procedures provide sufficient resilience to maintain mission objectives during cyber incidents.

5.2. Consolidate Key Findings

Now convert the discussion into a small number of clear findings. Findings should describe what did not work or what worked well. If possible, limit to roughly 5 to10 major findings. The goal is clarity, not documenting every comment. An example format is below.

Finding ID	Category	Description
F-1	Detect	Replay command packets would likely not be detected by current telemetry monitoring
F-2	Respond	Operators were unclear on escalation procedures for cyber anomalies
F-3	Mission Impact	ADCS manipulation could lead to mission data loss within one orbit

5.3. Perform Mission Risk Assessment

After consolidating findings from the discussion, the exercise team should evaluate the mission risk associated with each adversary technique observed during the scenario.

SPARTA provides a Notional Risk Score (NRS) for each technique. NRS represents an estimated relative risk based on two factors: likelihood that an adversary could execute the technique and the potential impact to spacecraft systems if the technique succeeds. Using SPARTA NRS allows the exercise team to prioritize findings using a risk metric derived from space system adversary techniques rather than general cybersecurity threat models.

However, the NRS represents a generalized estimate across spacecraft systems. The tabletop exercise allows the team to evaluate how that technique would affect the specific mission architecture.

Participants should therefore evaluate both the SPARTA NRS for the technique and the mission impact observed during the exercise.

Example evaluation:

Technique	SPARTA NRS	Mission Impact	Exercise Observation	Resulting Risk
EX-0001.01 Replay Command Packets	25 (High)	Medium	Duplicate commands would likely not be detected	Medium
EX-0012 Modify on Board Values	19 (Medium)	High	Parameter changes could affect spacecraft configuration	High
EX-0012.08 ADCS Manipulation	24 (High)	High	Spacecraft attitude instability could degrade mission data	High

This process helps determine which attack techniques represent the most significant mission risk.

Techniques with both high NRS and high mission impact should be prioritized for mitigation. This allows organizations to prioritize defensive improvements based on adversary techniques that pose the greatest risk to spacecraft operations

High priority techniques should be addressed through improvements such as implementing additional SPARTA countermeasures, improving telemetry monitoring and anomaly detection, and updating operational response procedures.

5.4. Develop Recommended Actions

Each high or medium risk finding should produce an action. SPARTA countermeasures already provide a structured set of defensive controls that can be directly used to develop engineering, monitoring, or operational improvements.

Recommended actions may fall into several different categories:

Category	Examples
Architecture improvements	Add command authentication (CM0031: Authentication, CM0055: Secure Command Mode(s), isolate payload interfaces (CM0038: Segmentation)
Monitoring improvements	Telemetry anomaly detection, command monitoring (CM0032: IDPS)
Operational procedures	Incident escalation procedures (CM0088: Policy Update)
Training	Operator cyber response training (CM0041: User Training)

5.5. Assign Ownership and Priority

Recommendations are useless if nobody owns them.

Assign each action to an organization.

Example:

Action	Owner	Priority
Implement command monitoring	Flight software team	High
Develop cyber anomaly playbook	Mission operations	Medium

5.6. Document the Exercise Report

The final step of the AAR is producing a short exercise report.

The report should summarize the exercise scenario, findings, and recommended improvements.

A typical report includes the following sections.

Exercise Summary

- Mission context
- Scenario description
- Participants

Findings

- Detection observations
- Response observations
- Architectural observations

Recommendations

- System improvements
- Procedural updates
- Future training needs

Appendices

- SPARTA technique mapping
- Attack chain used in the scenario
- Identified indicators of behavior

The report should be distributed to mission leadership, cybersecurity teams, and system engineering organizations.

Output of the AAR

The AAR should produce the following artifacts.

- Documented observations from the exercise
- SPARTA technique to observation mapping
- Validated SPARTA attack chain for the mission architecture that can be reused for future exercises or threat modeling activities
- Countermeasure effectiveness evaluation
- Prioritized security improvements
- Exercise report summarizing findings

These outputs ensure the tabletop exercise leads to measurable improvements in mission cybersecurity and operational resilience.