# DEF CON 34
# SpaceCOP: Houston, We Have an Intrusion

Space Cyber Orbital Protection
Threat-informed onboard intrusion detection for spacecraft

**Brandon Bailey, Randi Tinney**
***Cybersecurity and Advanced Platforms Subdivision (CAPS)***
***The Aerospace Corporation***

* Research was funded by the Department of Homeland Security (DHS) Science and Technology Directorate to advance proactive detection capabilities tailored to the unique needs of the space domain.

**brandon.bailey@aero.org,**
**randi.j.tinney@aero.org**



**Approved for Public Release via OTR 2026-00836**

# MODERN SPACECRAFT REALITY

From isolated boxes with radios to networked, software-defined mission systems

## THEN | LEGACY ASSUMPTION

Isolated box with a radio

Ground Station
TT&C Only

## NOW | MODERN REALITY

- **FLIGHT SOFTWARE / APPS** — Mission functions, autonomy, payload apps
- **SOFTWARE BUS / MIDDLEWARE** — DDS, MQTT, ROS 2, etc.
- **ONBOARD COMPUTER** — Processing, health mgmt, fault protection
- **COTS / THIRD-PARTY** — COTS hardware, firmware, sensors, components
- **OS / RTOS / LINUX LAYER** — QNX, VxWorks, RTEMS, Linux
- **DATA STORAGE / FILE SYSTEM** — Flash, SSD, file systems, logs, telemetry

- **PAYLOAD** — Imaging, sensing, science, comms
- **ADCS** — Attitude determination & control
- **POWER SUBSYSTEM** — EPS, power distribution, batteries, solar
- **PROPULSION** — Thrusters, valves, propellant mgmt
- **THERMAL CONTROL** — Heaters, radiators, temp sensors
- **TT&C / COMMUNICATIONS** — Radios, modems, antennas, routers, encryption

- **CROSSLINK / EXTERNAL CONNECTIVITY** — Inter-satellite links, relay, commercial nets
- **UPLINK / DOWNLINK TO GROUND SEGMENT** — Command, telemetry, data downlink
- **GROUND SYSTEMS / MISSION OPS** — Monitoring, planning, analytics, commanding
- **CLOUD / ENTERPRISE / DEV PIPELINE** — Data processing, storage, ML/AI, dashboards
- **SUPPLY CHAIN / THIRD-PARTY ECOSYSTEM** — Vendors, components, updates, services

**INTERNAL NETWORKED SUBSYSTEMS / BUSES**
SpaceWire | CAN / CAN FD | MIL-STD-1553 | Ethernet (TSN) | PCIe | I2C | SPI

### 1 NETWORKED SUBSYSTEMS

Multiple connected components, buses, and external links.

### 2 SOFTWARE-DEFINED FUNCTIONALITY

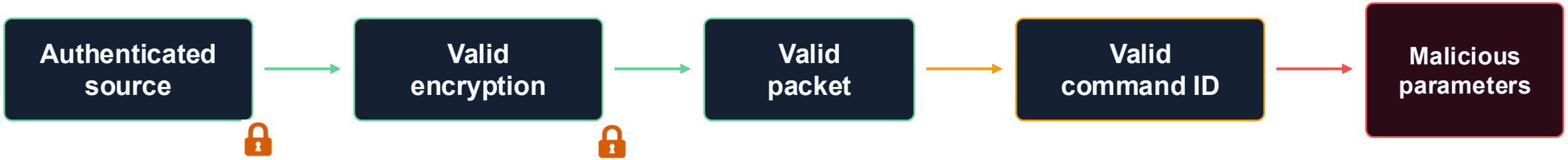
Capabilities delivered through software, middleware, and configurable platforms.

### 3 EXPANDED CYBER ATTACK SURFACE

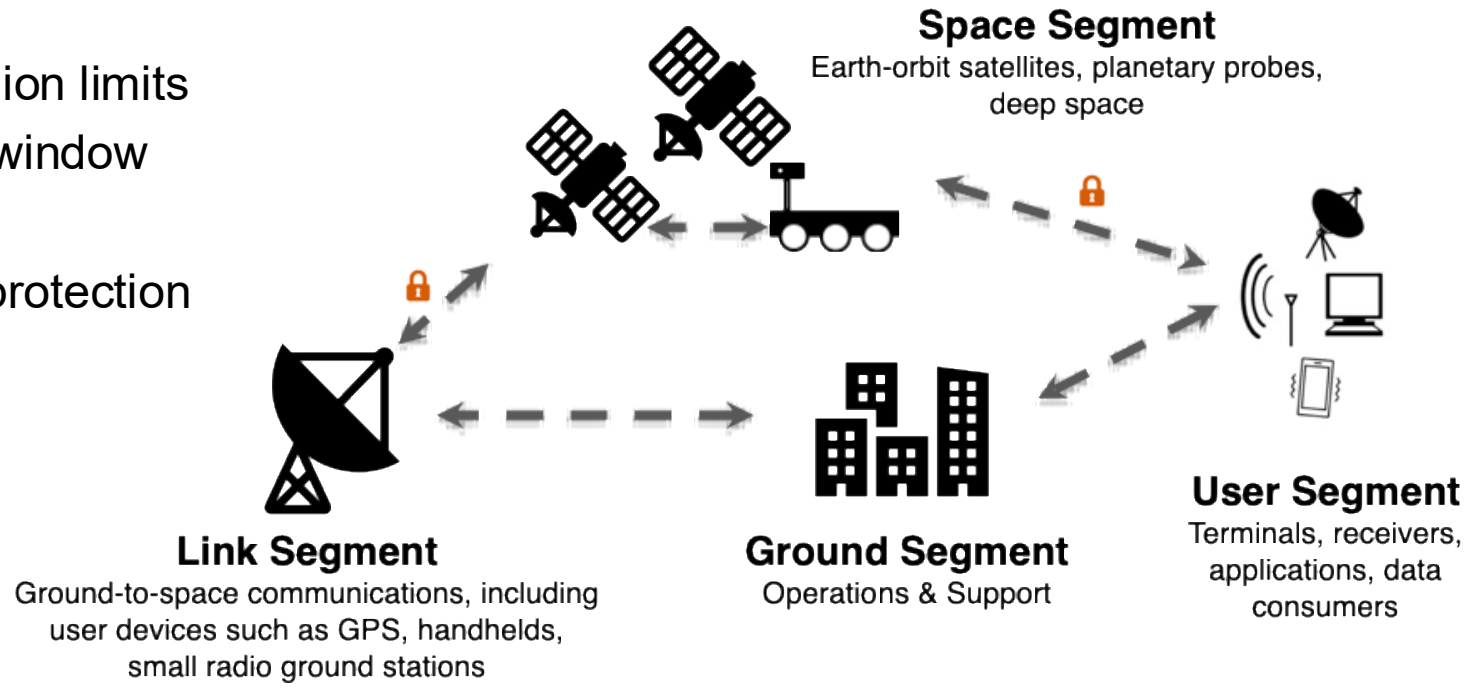
More connections, more components, more ways to be targeted.

# The Command Was Valid

A valid command can still be malicious.

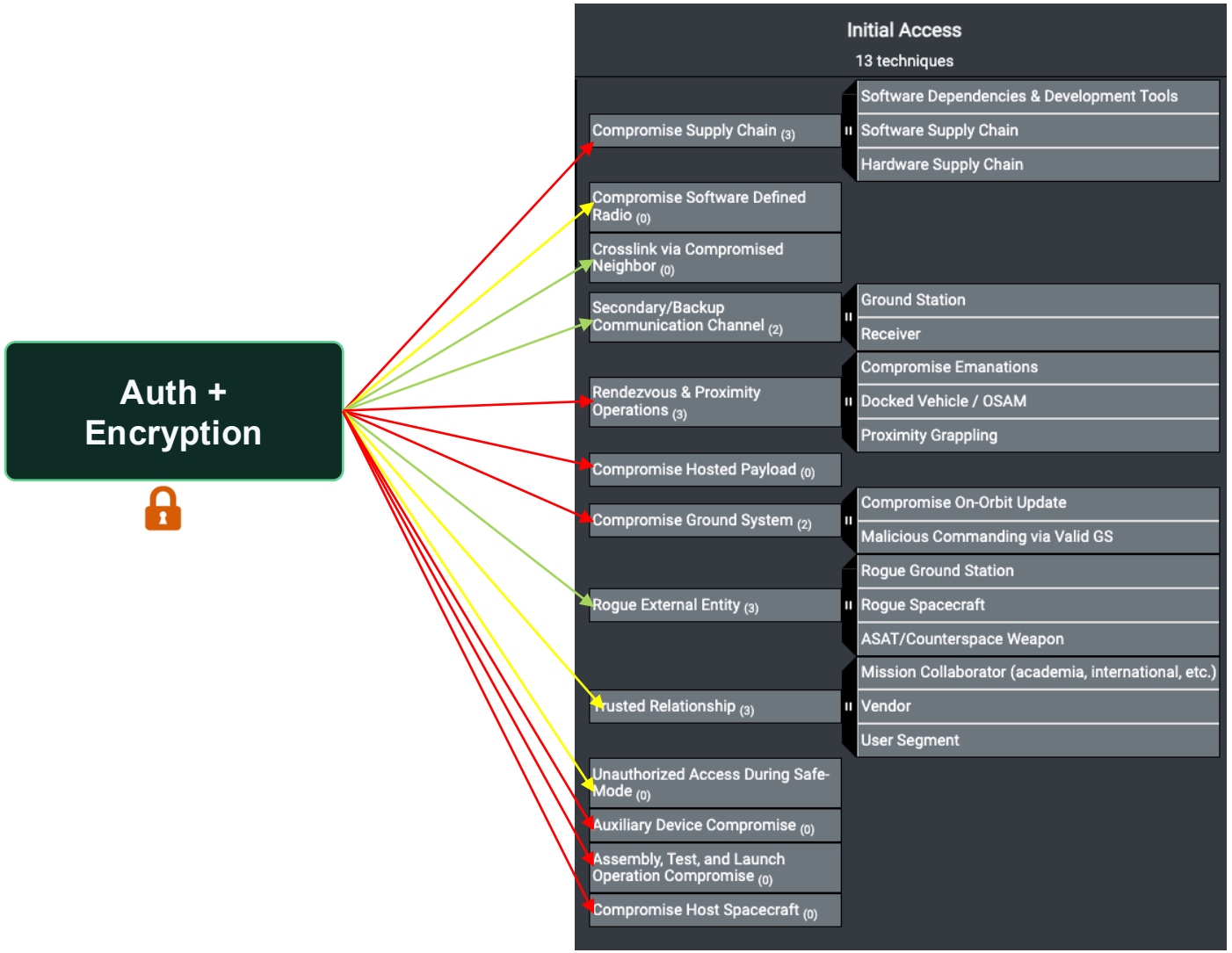



- Authentication tells the spacecraft who sent the command. It does not prove the command is safe, expected, or mission-appropriate.
  - Reaction-wheel torque outside safe mission limits
  - Thruster command outside a maneuver window
  - Safe-mode exit at an unexpected time
  - Authorized command used to disable a protection



# Trusting the Link Is Not Enough

The link can be protected while the vehicle is compromised.




- Authentication and encryption are highly effective against unauthorized communication but provide limited protection when the attacker compromises a trusted endpoint or uses valid credentials.

**Encryption protects the pathway. Onboard IDS watches what happens after access is granted.**

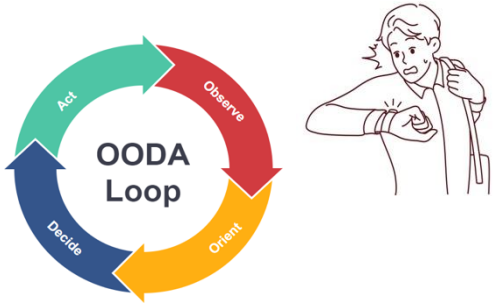
# Why the Ground May Never See It

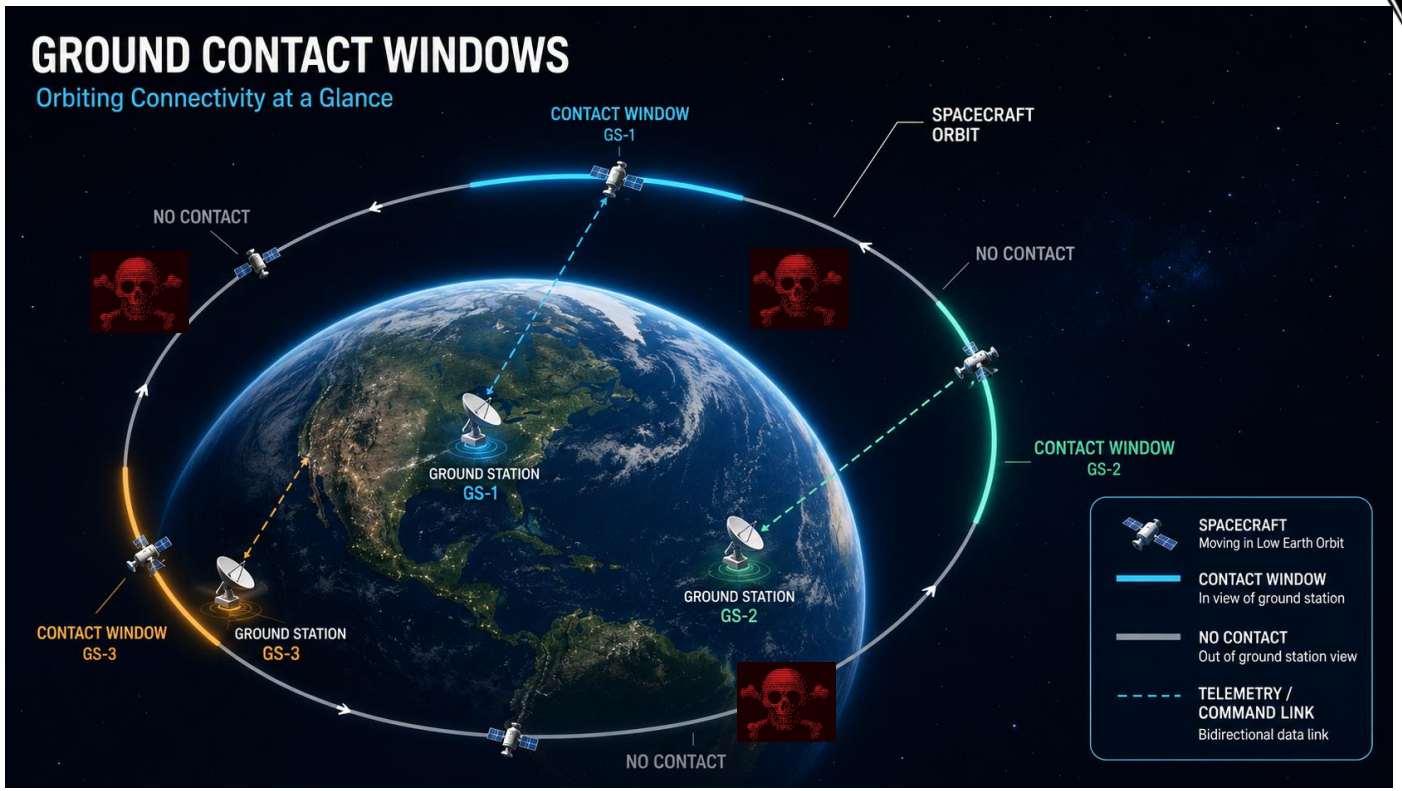
## Don’t Fall into the Trap

- Let’s only monitor the telemetry on the ground

What If ????

- The attack occurs outside a ground contact window.
- The relevant operating-system or process telemetry is not downlinked.
  - Cyber relevant telemetry is misunderstood and underutilized
- The ground system itself may be the source of the compromise.
- The effect occurs faster than an operator can respond.
  - Latency during contact windows provides advantage to attackers




**Ground-only monitoring creates blind spots during the exact window that matters.**

# Monitoring a Space Mission for Cyber Intrusion

*Be Accountable, Don't Be Satisfied with Basic Protection / Level 1 Maturity*

Where many are…

**Maturity Level 0**

No sensor placement anywhere and not monitoring hosts/endpoints

**Maturity Level 1**

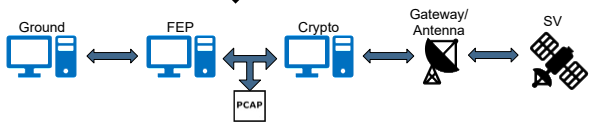
Only dependent on border sensors (egress only) only traditional IT traffic with some, rarely all, machines reporting to SIEM (i.e., Splunk/ELK). Ground only.

**Maturity Level 2**

Have sensors only at border feeding SOC capability but focus is only traditional IT traffic. All machines reporting to SIEM (i.e., Splunk/ELK). Sensors placed in-line of critical data flows feeding local SOC and analyzes TT&C traffic for bad commands going to SV.

**Maturity Level 3**

Sensors placed in-line of critical data flows feeding local SOC and analyzes TT&C traffic for bad **commands** & analyzing downlinked **telemetry** for intrusion. IDS deployed on SV.



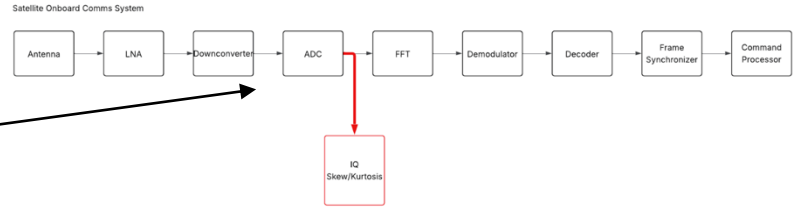

Often overlooked area for CND/IR monitoring

Given malicious commanding from ground is considered high risk, we need to monitor this link for malicious commanding

Aerospace has operational SW called Eirene Sceptre to help with monitoring TT&C using AI/ML and signatures

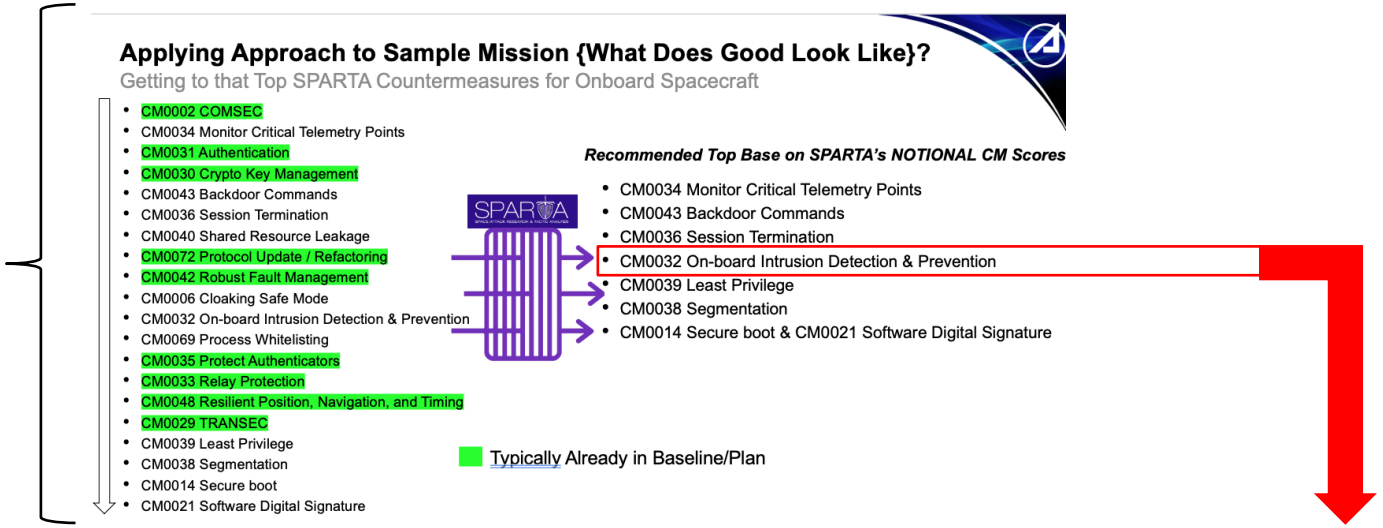
The future MUST include on-board monitoring to include:

- Commands/Telemetry
- Operating System
- SW uploads/downloads
- Critical memory regions
- FSW
- Payloads
- RF Signals
- …

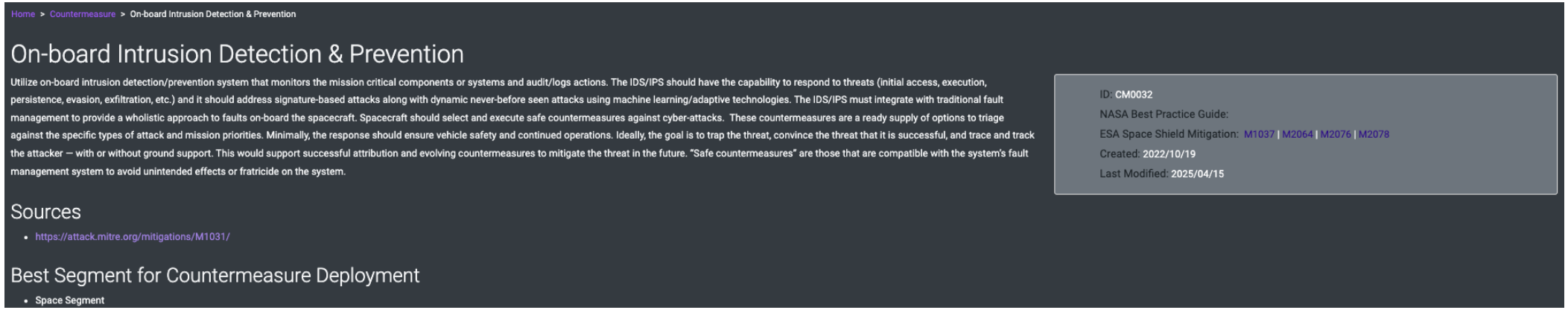


## SPARTA Countermeasure Prioritization

https://sparta.aerospace.org/resources/Prioritization_SPARTA_CMs_pub.pdf




*One of the key countermeasures that would have helped detect many previous mentioned attacks  onboard IDS*



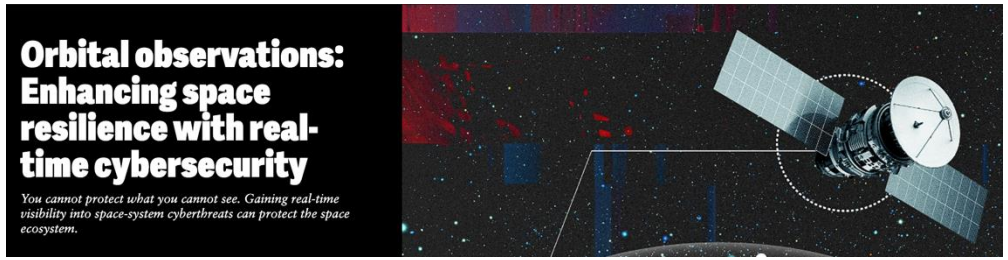
# Pointers are Everywhere

## USG Policy, Commercial, NIST, etc.

- Government experiments recommending detection for future missions
- Leads to NSS Policy identified importance of detection

**The First Defense: RF Signal Analysis for Early Cyber Threat Detection in Space**

Detecting cyber threats before they reach software is key to protecting space systems. This session focuses on Radio Frequency Signal Anomaly Detection (RFAD) as a method for early warning. Using case studies and competition results from MIT's Anomaly Detection Challenge, we'll examine how RFAD algorithms operate under tight SWaP constraints and what industries like GNSS and IoT can teach us. Learn best practices for integrating RFAD into space architecture as a first layer of defense.




https://www.deloitte.com/us/en/insights/industry/government-public-sector-services/critical-need-for-cybersecurity-in-space-systems.html

**CNSS Instruction 1200**

Updates

- Expansion on the EO-directed areas focused on spacecraft capabilities:
  - **On-board Intrusion Detection and Prevention** [§13.c]
    - Real-time out-of-band monitoring, detecting anomalous activities, and responding to threats
    - Continuous event logging, secure transmission of events, automated alerting
  - **Space Platform Hardware Roots of Trust for Secure Boot** [§13.e]
    - Cryptographic trust for boot; ability to restore or promote to a known trusted state
    - Hardware designs need to consider hardware-based comp
  - **Secure Software Patch Development and Deployment** [§
    - Identify, prioritize, and remediate vulnerabilities across spa
    - Secure deployment and verification of software/firmware p

| ID | Name |
|---|---|
| AU-2 | Event Logging |
| AU-3 | Content of Audit Records |
| AU-3(1) | Content of Audit Records \| Additional Audit Information |
| AU-4 | Audit Log Storage Capacity |
| AU-4(1) | Audit Log Storage Capacity \| Transfer to Alternate Storage |
| AU-5 | Response to Audit Logging Process Failures |
| AU-6(1) | Audit Record Review, Analysis, and Reporting \| Automated Process Integration |
| AU-6(4) | Audit Record Review, Analysis, and Reporting \| Central Review and Analysis |
| AU-8 | Time Stamps |
| AU-9 | Protection of Audit Information |
| AU-9(2) | Protection of Audit Information \| Store on Separate Physical Systems or Components |
| AU-14 | Session Audit |
| CA-7(6) | Continuous Monitoring \| Automation Support for Monitoring |
| CP-10 | System Recovery and Reconstitution |
| IR-4 | Incident Handling |
| IR-4(11) | Incident Handling \| Integrated Incident Response Team |
| IR-4(12) | Incident Handling \| Malicious Code and Forensic Analysis |
| IR-4(14) | Incident Handling \| Security Operations Center |
| IR-5 | Incident Monitoring |
| IR-5(1) | Incident Monitoring \| Automated Tracking, Data Collection, and Analysis |
| PL-8 | Security and Privacy Architectures |
| PL-8(1) | Security and Privacy Architectures \| Defense in Depth |
| RA-10 | Threat Hunting |
| SA-8(21) | Security and Privacy Engineering Principles \| Self-analysis |
| SA-8(22) | Security and Privacy Engineering Principles \| Accountability and Traceability |
| SA-8(23) | Security and Privacy Engineering Principles \| Secure Defaults |
| SC-5 | Denial-of-service Protection |
| SC-5(3) | Denial-of-service Protection \| Detection and Monitoring |
| SC-7(9) | Boundary Protection \| Restrict Threatening Outgoing Communications Traffic |
| SC-7(10) | Boundary Protection \| Prevent Exfiltration |
| SC-16(2) | Transmission of Security and Privacy Attributes \| Anti-spoofing Mechanisms |
| SI-3 | Malicious Code Protection |
| SI-3(10) | Malicious Code Protection \| Malicious Code Analysis |
| SI-4 | System Monitoring |
| SI-4(1) | System Monitoring \| System-wide Intrusion Detection System |
| SI-4(2) | System Monitoring \| Automated Tools and Mechanisms for Real-time Analysis |
| SI-4(4) | System Monitoring \| Inbound and Outbound Communications Traffic |
| SI-4(5) | System Monitoring \| System-generated Alerts |
| SI-4(10) | System Monitoring \| Visibility of Encrypted Communications |
| SI-4(11) | System Monitoring \| Analyze Communications Traffic Anomalies |
| SI-4(16) | System Monitoring \| Correlate Monitoring Information |
| SI-4(23) | System Monitoring \| Host-based Devices |
| SI-4(24) | System Monitoring \| Indicators of Compromise |
| SI-4(25) | System Monitoring \| Optimize Network Traffic Analysis |
| SI-7(8) | Software, Firmware, and Information Integrity \| Auditing Capability for Significant Events |
| SI-10(6) | Information Input Validation \| Injection Prevention |
| SI-16 | Memory Protection |

***Using/combining only these 40+ NIST 800-53 controls "should" result in an IDS/IPS being deployed on the spacecraft***

**Detect is what connects the dots because it is the sensing layer that enables every other resiliency function to matter.**

# A Spacecraft Is Not a Data Center

The constraints are harder, but the behavior is more deterministic.


**Defender advantage: spacecraft operations are often predictable enough to baseline command timing, process behavior, and subsystem state.**

# Fault, Attack, or Both?

Seeing the Unseen…

| Observable | Benign or Fault Cause | Potential Attack Cause |
|---|---|---|
| Unexpected time delta | Oscillator drift, GNSS fault, bad time update | Time spoofing or clock manipulation |
| Abnormal sensor value | Radiation upset, sensor failure, calibration error | Sensor-data or onboard-value manipulation |
| New process or syscall | Application restart, software defect, maintenance activity | Rogue application, malware, privilege abuse |
| CPU or memory spike | Fault recovery, software loop, workload increase | Resource exhaustion or ransomware-like behavior |
| Packet loss or latency | RF conditions, handover, congestion | Jamming, flooding, protocol manipulation |

**Traditional FDIR ≠ Enough**

- Fault Detection, Isolation, and Recovery (FDIR) systems are a standard part of space vehicle design, aimed at handling random hardware or software failures due to environmental factors (radiation, thermal drift, aging components)
- These faults are often well-characterized behaviors and repeatable recovery procedures

**Cyber Events Are Different but Similar in Effect**

- Cyberattacks are intentional and can be considered dynamic faults injected with adversarial intent.

**Same Outcomes, Different Origins**

- The result of cyber faults is often **indistinguishable** from traditional system faults without contextual **awareness**

**An anomaly is evidence, not attribution. Context and correlation determine whether it is a fault, misuse, or attack.**

# Introducing SpaceCOP

## Space Cyber Orbital Protection

Onboard

Threat-informed

Behavior-based


- An onboard, threat-informed intrusion detection capability that monitors spacecraft behavior for evidence of malicious or unauthorized activity.

### Open-source cFS SpaceCOP

NASA Core Flight System application
Releases at DEF CON 34
Uses SPARTA IOBs
Integrates with cFS messages, events, telemetry, and Linux monitoring

### Side-loaded SpaceCOP

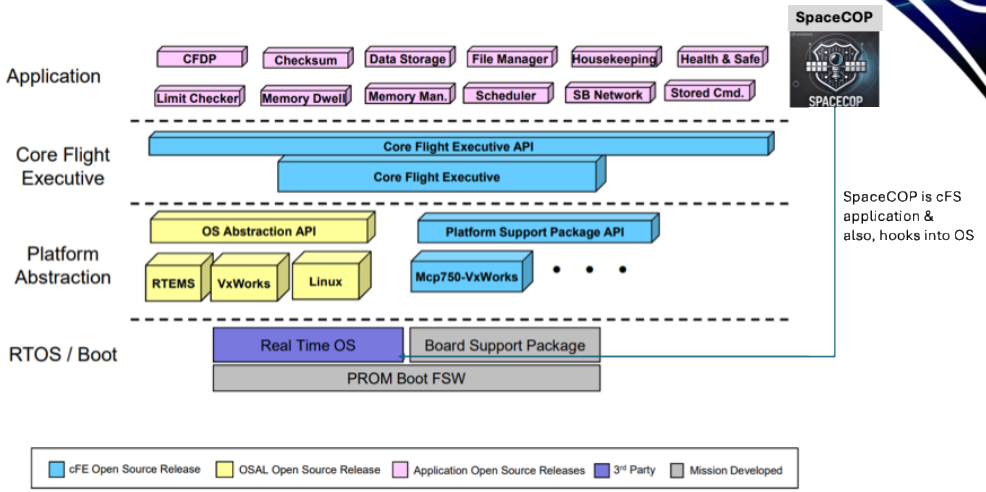
Closed-source implementation
Broader compatibility beyond cFS
Separate monitoring environment
Designed to preserve visibility through stronger independence

- SpaceCOP is a custom cFS flight application that utilizes a custom kernel module to help with the detection of SPARTA IoBs
  - Created custom kernel module to monitor for SYSCALLs
  - Monitors file integrity
  - Monitors disk/memory/CPU utilization
  - Hooks into the software bus to monitors sensor states and commands
  - Monitors for time delta
- The added IDS currently does not overwhelm system resources, showing that it is an effective way to monitor for IOBs

**Both versions try to answer the same question: what is the spacecraft doing right now, and does that behavior look adversarial?**

# Multi-Threaded Flight Application

## Ensures continuous monitoring without delay




- Command Monitoring
  - Monitors all potential APIDs and looks for unknown messages
  - Detects potential flooding attacks
  - Monitors commands for sensors for state changes and parameters
  - Monitors packet time
- Disk Monitoring
  - Checks that available storage space remains above a defined threshold
- File Monitoring
  - Compares contents and hashes of files within critical directories. Alerts if a file is added, changed, or deleted
- Memory Monitoring
  - Monitors memory usage to detect processes that exceed the threshold amount
- Syscall Monitoring
  - Utilizes custom kernel module to monitor any suspicious syscall activity
- System Monitoring
  - Checks CPU usage for all programs to ensure they are below a defined threshold
  - Detects any new processes being spawned

# Using SPARTA IOB for IDS Creation

## Unexpected Time Delta Detected Example

- Review 2025's DEF CON presentation for overview of IOBs

https://www.youtube.com/watch?v=UL_c4K5dTuc


https://sparta.aerospace.org/resources/OTR202500882-DefCon2025-Hacking_Space_to_Defend_It_1.pdf

**An IOC tells us what artifact may be present**

**An IOB tells us what the spacecraft is doing**

### Indicator of Compromise

Known malware hash
Known rogue process name
Known malicious IP address
Known command signature

### Indicator of Behavior

Valid command outside approved window
Unexpected write to boot memory
Crypto key accessed by unexpected process
Unexpected subsystem communication

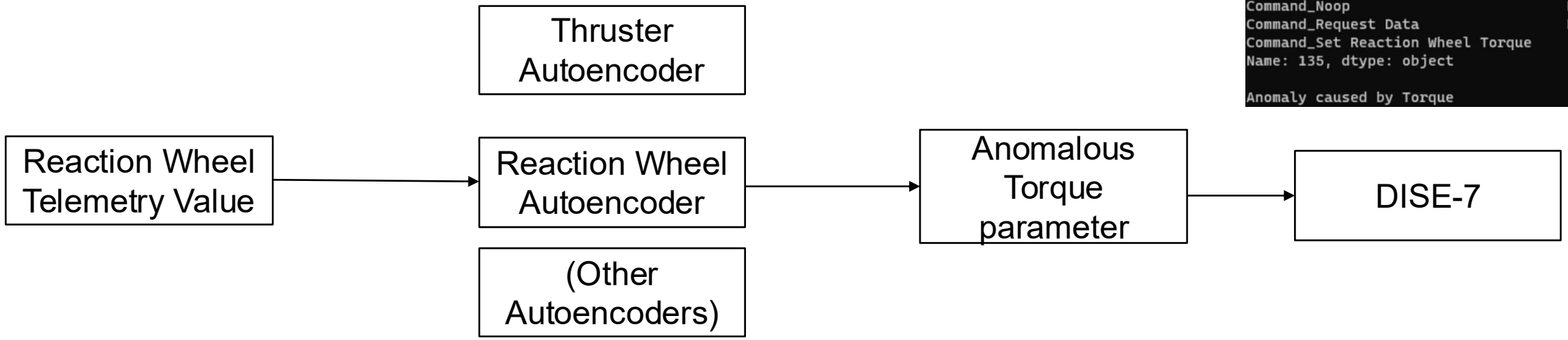
**SPARTA Technique** → **Adversary Behavior** → **Indicator of Behavior** → **Required Data** → **SpaceCOP Rule** → **Alert + Context**

# Machine Learning Overview

## Detecting Anomalies in cFS Messages

- Autoencoders used for detecting anomalies in provided data
- Training an autoencoder for each type of command and telemetry increases the accuracy and simplicity of the ML models
  - Training materials in the lab are based off a “normal behavior script”
  - Models can be easily updated via CFDP if changes are needed
- Can be used for Explainable AI, showing which features of the data caused an input to be considered anomalous

```
[SPACECOP] Anomaly detected in command 4:
Anomaly                                True
Wheel                                   1.0
Torque                                -15.0
Command_Noop                          False
Command_Request Data                  False
Command_Set Reaction Wheel Torque      True
Name: 4, dtype: object

Anomaly caused by Torque

[0]
[SPACECOP] Anomaly detected in command 12:
Anomaly                                True
Wheel                                   3.0
Torque                                  5.0
Command_Noop                          False
Command_Request Data                  False
Command_Set Reaction Wheel Torque      True
Name: 12, dtype: object

Anomaly caused by Wheel

[1]
[SPACECOP] Anomaly detected in command 135:
Anomaly                                True
Wheel                                   1.0
Torque                                 50.0
Command_Noop                          False
Command_Request Data                  False
Command_Set Reaction Wheel Torque      True
Name: 135, dtype: object

Anomaly caused by Torque
```




**Alpha version is implemented into initial release of SpaceCOP**

# Key Considerations for Spaceborne Intrusion Detection

Detection must understand spacecraft behavior, mission context, and space-specific data

- **Define What "Bad" Looks Like**
  - Establish normal command and telemetry behavior early (pre-flight, simulation, digital twin).
  - Treat anomaly detection as mission-specific where each spacecraft has its own operational "signature."
  - Incorporate real mission data into behavioral baselines for both nominal and off-nominal ops.
- **Adopt a Common Threat Taxonomy**
  - Use frameworks like **SPARTA** to derive detections via Indicators of Behavior (IOBs) which map to adversary TTPs.
  - Standardize how detections are expressed and shared (e.g., STIX 2.1).
- **Enable Tools to Understand Space Telemetry**
  - Extend DCO tools (Splunk, ELK, Zeek) to parse CCSDS, CFDP, and mission-specific protocols.
  - Implement telemetry decoders that normalize data before analytics or SIEM ingestion.
    - Market may be too small for vendors to invest in space-protocol capabilities – industry growing quickly.
  - Build reusable parsing layers to make detections portable across missions.

**The space-cyber domain today mirrors early ICS/OT security with limited vendor ecosystem, bespoke protocols, and few tailored detection tools. Now many COTS and FOSS products support ICS/OT protocols, detections, etc.**

# Key Considerations for Spaceborne Intrusion Detection

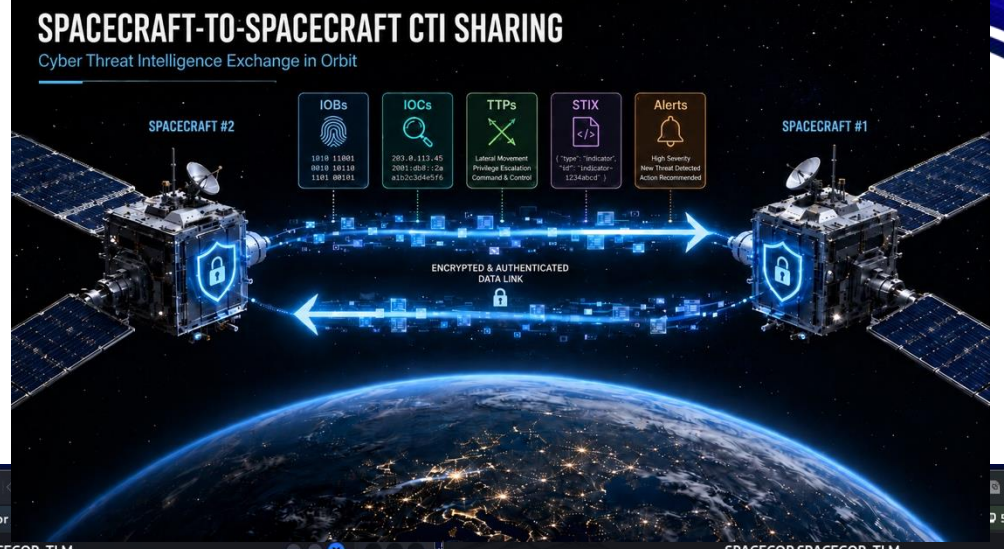
Detection must understand spacecraft behavior, mission context, and space-specific data

- **Account for Mission-Specific Architectures**
  - Plan for unique command/telemetry databases and automate schema ingestion where possible.
    - Aerospace built tool for mission that mapped all SPARTA IOBs to existing telemetry pneumonic
      - Also, leveraged design documents and RAG to derive “normal” limit values to drive anomalous detection
  - Leverage spacecraft modes and transitions to guide mode-aware detection logic.
  - Use modular detection logic that ties to subsystem roles (ADCS, EPS, FSW, etc.).
- **Design for Context Awareness**
  - Identical commands or telemetry patterns can be safe in one mode and catastrophic in another.
  - Detection must be mode-, phase-, and subsystem-aware (e.g., safe mode, orbit insertion, payload ops).
  - Avoid static thresholds and use adaptive baselines that evolve over mission life.
  - Incorporate physics- or control-model–based validation to confirm abnormal telemetry.

**Detection isn’t a bolt-on capability & it must be engineered into the mission architecture from the start.**

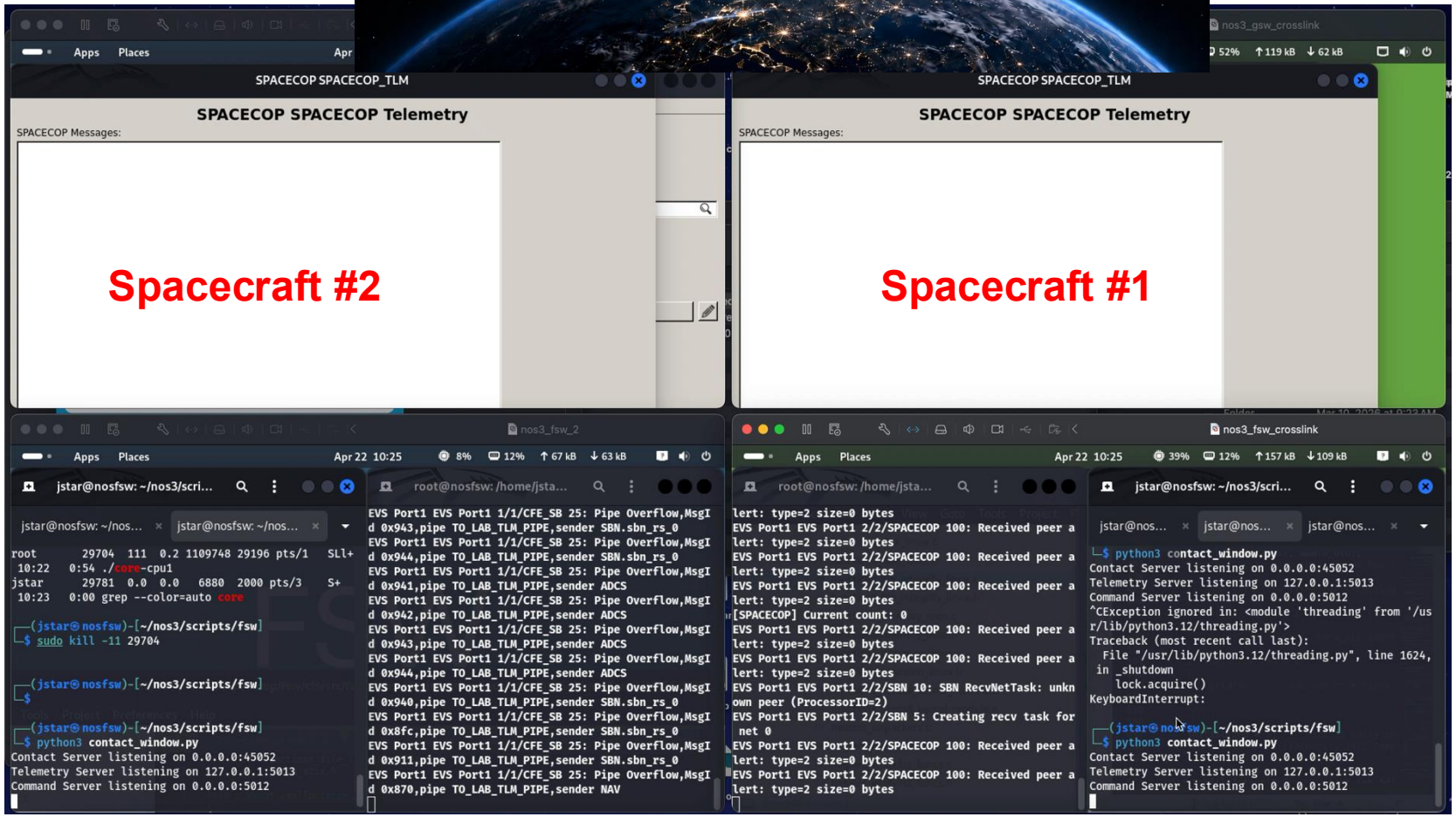
# Cyber Threat Intelligence

- Indicators within SpaceCOP generate CTI based on triggers
- Formatted in the STIX 2.1 language for computer ingestion
- Includes
  - Timestamps
  - Geographical locations
  - Any relevant artifacts **<big TBD>**
    - Current standard lacking
- Stored onboard until download is requested





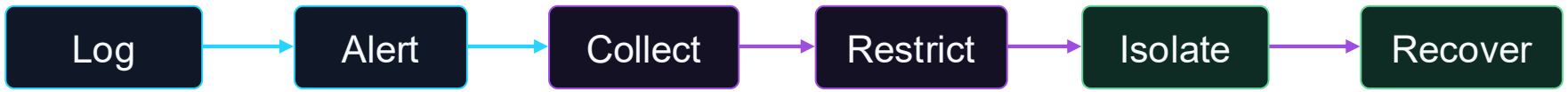
```
{
  "type": "bundle",
  "id": "bundle--b9283eab-1a8b-4d45-9bfe-0a7891234bcd",
  "objects": [
    {
      "type": "indicator",
      "spec_version": "2.1",
      "id": "indicator--fdc708a9-1cd4-496d-bbdb-eafe361c0f0c",
      "created": "2026-01-06T12:00:00.000Z",
      "modified": "2026-01-06T12:00:00.000Z",
      "name": "File or Data Integrity Check Failure",
      "pattern": "[file:hashes != 'expected_hash_value' AND file:name = 'data_file']",
      "pattern_type": "stix",
      "confidence": 85,
      "x_sparta_iob_id": "DISE-1",
      "x_latitude": 37.7749,
      "x_longitude": -122.4194
    },
    {
      "type": "file",
      "spec_version": "2.1",
      "id": "file--12345678-90ab-cdef-1234-567890abcdef",
      "name": "data_file",
      "hashes": {
        "MD5": "ae7f3c6b1234567890abcdef12345678",
        "SHA-1": "0123456789abcdef0123456789abcdef01234567",
        "SHA-256": "3a7bd3e2360a0a1f3c8c5e1d4f0f3a2b4c5d6e7f8a9b0c1d2e3f4a5b6c7d8e9f"
      },
      "size": 435200,
      "mime_type": "application/octet-stream"
    }
  ]
}
```

# Upgrading to Intrusion Prevention

## The next big step

- With the ability to detect threats, the next step is to prevent them
  - Sophisticated threat actors will perform attacks when telemetry is not easily available
- With CTI reports being generated, the next logical step is to take the reports, actively search for the issues, and prevent them before the attack is performed
  - Easier said than done!
- Things to do before this ability can be performed:
  - Identify acceptable risk to mission before automatically blocking commands, removing/quarantining files, etc.
    - Will need an ability to custom tailor for all missions using SpaceCOP
  - Implement AI/ML to determine the correct course of action for each indicator

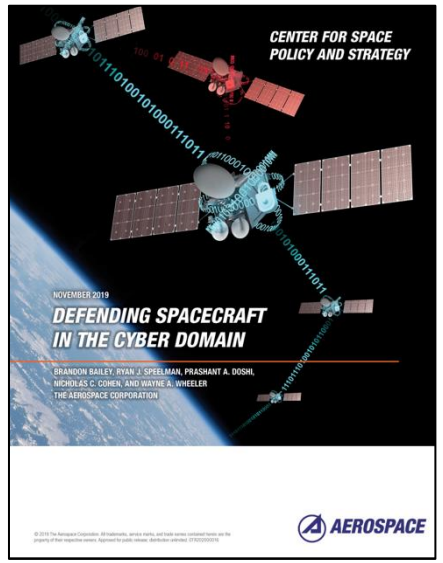


# Automated Cyber Response Is No Longer Optional

## Spacecraft must respond in real time, integrating with FMS is key to mission continuity

- Automated response **is essential**
  - Spacecraft must be equipped to isolate faults, block malicious commands, or switch to safe configurations without waiting for ground input
- Integrating with Fault Management System (FMS) is critical
  - Cyber response must tie into existing fault management systems to avoid conflicts, reduce risk of false positives, and ensure mission continuity.
- Cyber must become a first-class spacecraft function
  - Like attitude control or thermal regulation, cyber defense must operate as a real-time onboard control system, not just a monitoring tool
- Call to action paper in November 2019 identified this

https://csps.aerospace.org/sites/default/files/2021-08/Bailey_DefendingSpacecraft_11052019.pdf




***Intrusion Detection and Prevention Systems***

The backbone of a cyber-resilient spacecraft should be a robust intrusion detection system (IDS). The IDS should consist of continuous monitoring of telemetry, command sequences, command receiver status, shared bus traffic, and flight software configuration and operating states. From a telemetry

recommended course of action. If a severe rules violation occurs or a higher threshold is crossed, the spacecraft's intrusion prevention system (IPS) will take automated actions, which may include swapping to a redundant side, quarantining command sequences, reloading flight software, and/or halting suspect units. An example of the first

The IPS system should be integrated into the existing onboard spacecraft fault management system (FMS) because the FMS has its own fault detection and response system built in. Typically,

seen in operations. The reason that both the IPS and FMS systems should be integrated is that they are essentially performing the same functions but are looking for different anomaly signatures. In fact, there may be scenarios where each of them detects an anomalous condition and attempts to take an action. Having them integrated ensures they do not take conflicting actions.

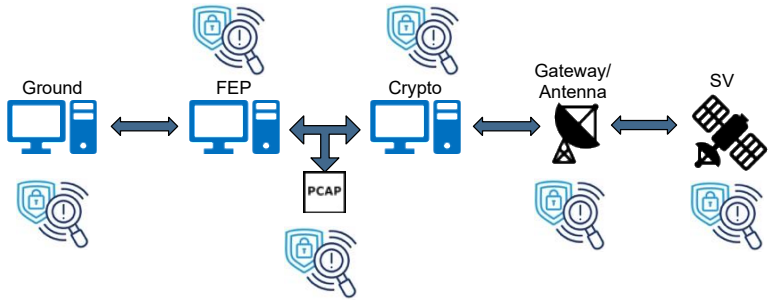
The spacecraft IPS and the ground should retain the ability to return critical systems on the spacecraft to known cyber-safe mode. Cyber-safe mode is an operating mode of a spacecraft during which all

- SPARTA techniques and now IOBs support this position that attacks can occur and be detected but response is needed to assure mission continuity

**Detection and Response are Necessary, Cyber can be Mission Killer**

# Let's Get To >>>> Maturity Level 3

## Should Be Monitoring …




Sensors placed in-line of critical data flows feeding local SOC and analyzes TT&C traffic for bad **commands** & analyzing downlinked **telemetry** for intrusion. IDS deployed on SV.

**Command & Telemetry Layer**

- **Command Validation:** Detect unauthorized, unexpected, or out-of-sequence commands.
- **Command Rate / Frequency Monitoring:** Identify command floods, replays, or deviations from scheduled operations.
- **Telemetry Integrity:** Verify telemetry sources, value ranges, etc.
- **Cross-Correlation:** Compare ground command logs with received commands to detect insertion or modification.
- **Telemetry Delay / Drop Patterns:** Detect link manipulation or time-delay attacks.
- **IOB-based Behavioral Monitoring:** Watch for irregular command dependencies or anomalous mode transitions.

**RF and Communications / Link Layer**

- **Signal Authentication:** Validate uplink identity and signal fingerprint against authorized ground stations.
- **Anomaly Detection in RF Spectrum:** Identify spoofed signals, replay bursts, or jamming attempts.
- **Protocol Parsing:** Decode and inspect space protocols (e.g., CCSDS/AOS/CFDP) for malformed or malicious payloads.
- **Timing / Doppler Anomalies:** Detect shifts inconsistent with known orbital dynamics (indicating spoofing).
- **Crosslink Integrity:** Monitor for unauthorized cross-satellite data exchanges or injection attempts.
- **Downlink Sanitization:** Inspect downlinked data for embedded malware or command channels.

**Software Uploads, Downloads, and Updates**

- **Upload Source Verification:** Authenticate and verify update origin and digital signatures.
- **Delta / Patch Validation:** Monitor differences between new and expected binary content before applying.
- **Transfer Channel Monitoring:** Inspect update data streams for embedded payloads or injected content.
- **Execution After Update:** Watch for abnormal FSW behavior following update (e.g., CPU usage, command respo
- **Rollback or Reversion Events:** Detect unexpected downgrade or version rollbacks.

**Payload Monitoring**

- **Payload Command Interface:** Detect unexpected or unscheduled payload tasking.
- **Data Exfiltration:** Watch for unauthorized data dumps or anomalous downlink sessions.
- **Payload-to-FSW Interaction:** Track abnormal access from payload software to core spacecraft buses or memory.
- **Configuration Integrity:** Verify payload configuration parameters against ground-approved settings.
- **Power / Resource Abuse:** Detect payloads consuming unexpected power or CPU cycles.

**Flight Software (FSW) & Operating System**

- **Process Monitoring:** Detect unexpected binaries, rogue processes, or abnormal resource usage.
- **Thread / Task Integrity:** Monitor RTOS task creation, timing jitter, and unauthorized priority elevation.
- **Memory Protection Violations:** Detect read/write operations into restricted or protected memory regions.
- **System Call Auditing:** Observe abnormal kernel calls or system library invocations.
- **FSW Baseline Comparison:** Continuously verify code segments, hashes, and loaded modules against flight-certified baselines.
- **Scheduler / Watchdog Tampering:** Monitor for watchdog timer disablement or altered execution cadence.

**Critical Memory Regions & Storage**

- **Firmware / Bootloader Integrity:** Verify secure boot chain and detect unauthorized firmware modifications.
- **EEPROM / Flash Monitoring:** Track changes to configuration tables, key stores, and stored software images.
- **Stack/Heap Anomalies:** Watch for buffer overflows or memory corruption patterns indicative of exploitation.
- **Cryptographic Key Store Access:** Detect unauthorized reads/writes to key vaults or crypto hardware modules.
- **Non-Volatile Memory Validation:** Periodic integrity checks and hash comparisons of static data.

**Subsystem-Specific Monitoring**

- **ADCS (Attitude Determination and Control System):** Monitor control loop commands, reaction wheel torque, and sensor fusion consistency.
- **EPS (Electrical Power System):** Detect anomalous load draws, bus voltage fluctuations, or spoofed sensor values.
- **Thermal Control:** Identify heater activation anomalies, unexpected thermal excursions, or falsified temperature telemetry.
- **Propulsion System:** Track unauthorized valve actuations, thruster firings, or inconsistent burn sequences.
- **Structures & Mechanisms:** Monitor motor control signals, actuator movement profiles, and deployment command sequences.

# Houston, We Have Visibility

## Download SpaceCOP Today

GitHub

nos3 NASA Operational Simulator for Space Systems


+


**The link is not the final trust boundary.**

**A valid command may still be malicious.**

**Detection must exist where the mission is affected.**

**SPARTA IOBs**
**sparta.aerospace.org**

**Break it. Test it. Build detections. Contribute spacecraft behaviors we have not considered.**

# Any Questions?