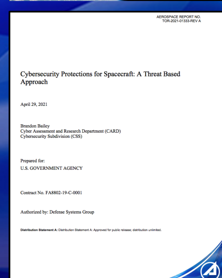
DefCon 2023: Aerospace Village
Building Space Attack Chains using SPARTA

Brandon Bailey
Cybersecurity and Advanced Platforms Subdivision (CAPS)
Cyber Assessment & Research Dept (CARD)
The Aerospace Corporation


AEROSPACE REPORT NO.
TOR-2021-01333-REV A

Cybersecurity Protections for Spacecraft: A Threat Based Approach

April 29, 2021

Brandon Bailey
Cyber Assessment and Research Department (CARD)
Cybersecurity Subdivision (CSS)

Prepared for:
U.S. GOVERNMENT AGENCY

Contract No. FA8802-19-C-0001

Authorized by: Defense Systems Group

Distribution Statement A: Distribution Statement A: Approved for public release; distribution unlimited.

Papers:

[Defending Spacecraft in the Cyber Domain](#)
[Establishing Space Cybersecurity Policy, Standards, & Risk Management Practices](#)
[Cybersecurity Protections for Spacecraft: A Threat Based Approach](#)
[Protecting Space Systems from Cyber Attack](#)

Presentations:

[DEF CON 2020: Exploiting Spacecraft](#)
[DEF CON 2021: Unboxing the Spacecraft Software BlackBox Hunting for Vulnerabilities](#)
[DEF CON 2022: Hunting for Spacecraft Zero Days using Digital Twins](#)

Aerospace TechBlog | DIGITAL ENGINEERING | RESEARCH | SPACE CYBER | SPACE DEBRIS | TECH & POLICY | AEROSPACE.ORG

Space Cyber
https://medium.com/the-aerospace-corporation/space-cyber/home

brandon.bailey@aero.org
240.521.4326 (c)

SPARTA
SPACE ATTACK RESEARCH & TACTIC ANALYSIS
https://sparta.aerospace.org/resources/

Approved for public release. OTR-2023-00989

Speaker Bio

Current Job: Senior Project Leader, Cybersecurity and Advanced Platforms Subdivision (CAPS), at The Aerospace Corporation
- Developing cyber labs for training, perform penetration testing & vulnerability assessments {Ethical Hacking!}
- Performing cybersecurity research on ground systems and spacecraft systems to better position the federal government with respect to protection of our critical space infrastructure.

2005

B.S. Electrical Engineering
West Virginia University

Lockheed Martin Supporting National Geospatial Intelligence Agency

2009

NASA's Independent Verification and Validation Program

Working for Small Business in West Virginia doing Spacecraft and Ground Simulation/ Emulation

2013

Transitioned to NASA Government Employee GS-13

Began "Hacking" Space Systems

2018

Left the Government as GS-15 to become Chief Technology Officer (CTO) for small business in West Virginia

- Supported NASA part-time as contractor

2019

Left Job as CTO to join Aerospace Corporation Federally Funded and Development Center

2023

Pen-tested / "Ethically Hacked" Space Systems

2013-2019

- Mars' Rovers (MER & MSL) & Deep Space Network (DSN) at JPL
- Hubble Space Telescope (HST) at GSFC
- Closed IONet (CIONet) within NASCOM at GSFC
- Space Network (SN) at the White Sands Complex (WSC)
- KSC Ground Systems Development and Operations (GSDO) Kennedy Ground Control System (KGCS) and Launch Control System (LCS)
- James Web Space Telescope (JWST) Ground System at the Space Telescope Science Institute (STScI) in Baltimore
- Huntsville Operations Support Center (HOSC) at Marshall Space Flight Center
- Near Earth Network (NEN) at Wallops Flight Facility
- ISS Mission Control Center (MCC) at Johnson Space Center
- Wind tunnels at Glenn Research Center
- Hypersonic Environment at Langley Research Center

2019-2023

- DefCON Presentations:
  - DEF CON 2020: Exploiting Spacecraft
  - DEF CON 2021: Unboxing the Spacecraft Software BlackBox Hunting for Vulnerabilities
  - DEF CON 2022: Hunting for Spacecraft Zero Days using Digital Twins
- Papers/Articles:
  - 2019: Defending Spacecraft in the Cyber Domain
  - 2020: Establishing Space Cybersecurity Policy, Standards, & Risk Management Practices
  - 2021: Cybersecurity Protections for Spacecraft: A Threat Based Approach
  - 2022: Protecting Space Systems from Cyber Attack
- July 2022 Congressional Testimony:
  - Video: https://science.house.gov/hearings?ID=996438A6-A93E-4469-8618-C1B59BC5A964
  - Written Testimony: https://republicans-science.house.gov/_cache/files/2/9/29fff6d3-0176-48bd-9c04-00390b826aed/A8F54300A11D55BEA5AF2CE305C015BA.2022-07-28-bailey-testimony.pdf
- SPARTA Launched
  - https://sparta.aerospace.org

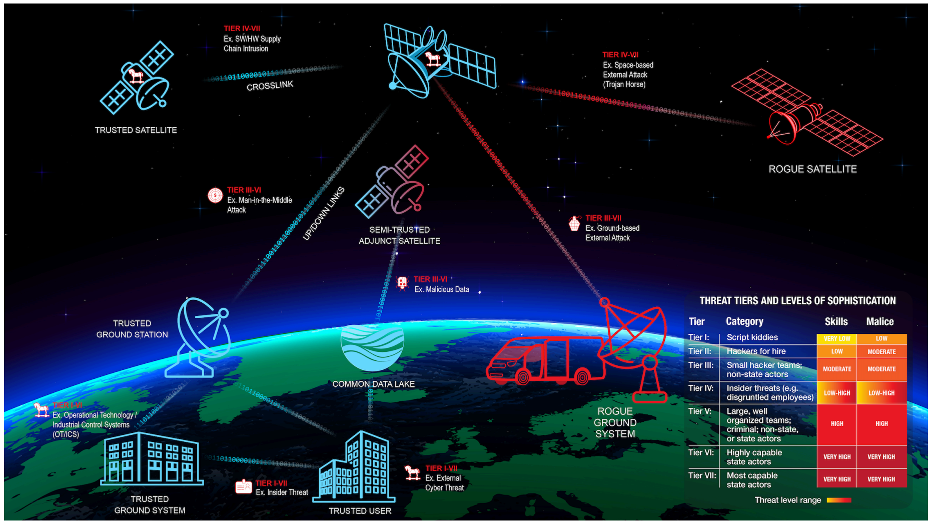
SPARTA
SPACE ATTACK RESEARCH & TACTIC ANALYSIS

NASA's Exceptional Service Medal (2019) for "groundbreaking" cyber work

The Cybersecurity in Space Problem

- Traditional spacecraft/payload architectures, sub-systems, and supply chains were developed before current cyber threats were envisioned
- Traditionally, cybersecurity for DoD, civilian and commercial space systems has concentrated on the ground segment with minimal, if any, cyber protections onboard the SV/payload
  - *Encryption/Authentication, TRANSEC, COMSEC, and TEMPEST are typically the only controls (if any)*

- Aerospace is helping lead advancement in cybersecurity for the spacecraft and ground systems
  - *Many articles/publications identify problems, but few are solutions oriented*
    - Aerospace has had concerted effort on publishing information publicly to inform commercial & gov space sector
  - *One area is helping customers define the "right" requirements*
    - Defining the requirements using threats / tactics, techniques and procedures (TTPs) vice compliance requirements (ISO/RMF baselines generated for traditional IT)
      - *TOR 2021-01333 REV A and now SPARTA provide resources to managers/developers/etc. to implement countermeasures to reduce cyber risk for space systems*



***blue lines** indicate normal expected communications/access*
***red lines** indicate communications from adversary's infrastructure directly*

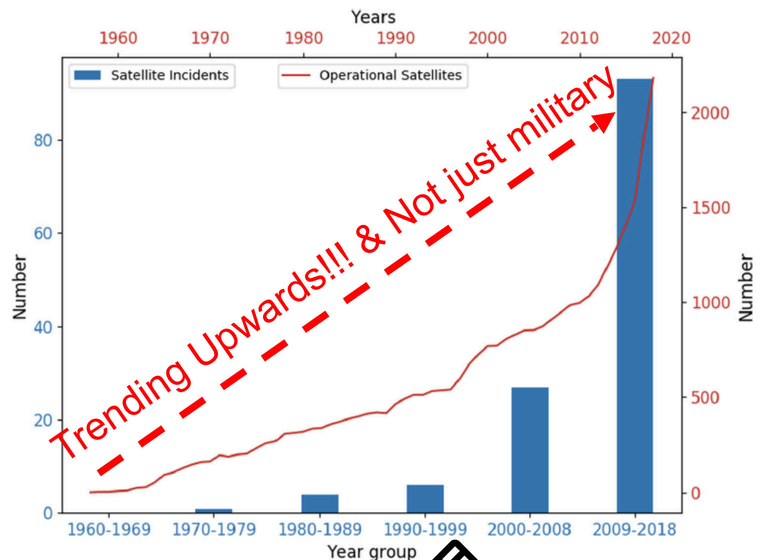
By defining the right cyber requirements/countermeasures, customers will be able reduce cyber risk for the space system

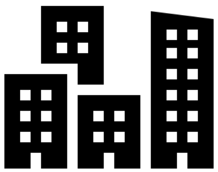
Example Cyber Incidents Against Space Systems

1. SPACE: Cybersecurity's Final Frontier, London Cybersecurity Report, *June 2015.*
2. Black Hat 2020: Satellite Comms Globally Open to $300 Eavesdropping Hack, Threatpost, Aug. 2020
3. Turla APT Group Abusing Satellite Internet Links, Threatpost, Sep. 2015
4. Network Security Breaches Plague NASA, Bloomberg, Nov 2008
5. Hackers Seized Control of Computers in NASA's Jet Propulsion Lab, WIRED, Mar. 2012
6. UT Austin Radio Radionavigation Laboratory
7. 2019 NASA OIG Report
8. Cyber security in New Space

Cyber security in New Space

Fig. 6 Number of satellites attacks per year group is plotted on the bottom and left axes, and the number of operational satellites between 1958 and 2018 is plotted on the top and right axes

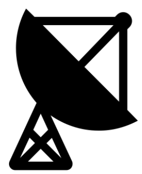




April 2005[4]: A rogue program penetrated NASA KSC networks, surreptitiously gathered data from computers in the Vehicle Assembly Building and removed that data through covert channels.

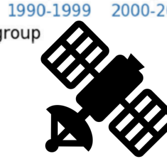
2011[5]: Cybercriminals managed to compromise the accounts of about 150 most privileged JPL users.

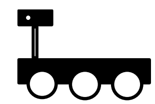
2018[7]: Weaknesses in JPL's system of security controls exploited; attacker moved undetected within multiple internal networks for about 10 months



Since 2007[3] several elite APT groups have been using — and abusing — satellite links to manage their operations — most often, their C&C infrastructure, for example, Turla.

Black Hat 2020[2]: Eavesdropping on Sat ISPs. Basically, ISP not protecting their links and it can be picked up easily.



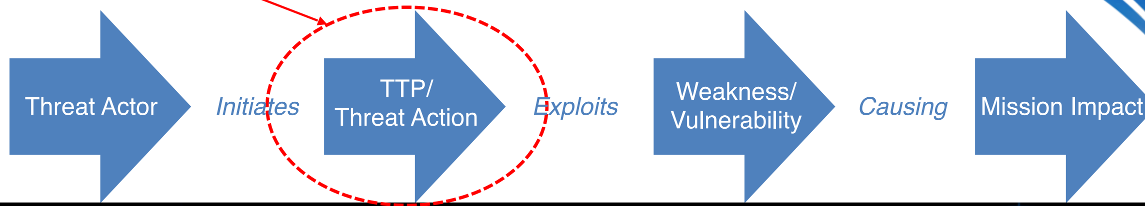


June/July 2008[1]: *Terra EOS AM-1/Landsat-7, attempted satellite hijacking,* hackers achieved all steps for remote command of satellite.

2013-2014:[6] UT Austin Radio-Navigation Lab conducts GPS spoofing for UAV control and navigation interruption.

Attacks/TTPs

Problem Statement: Where are these documented for space and how do you mitigate?

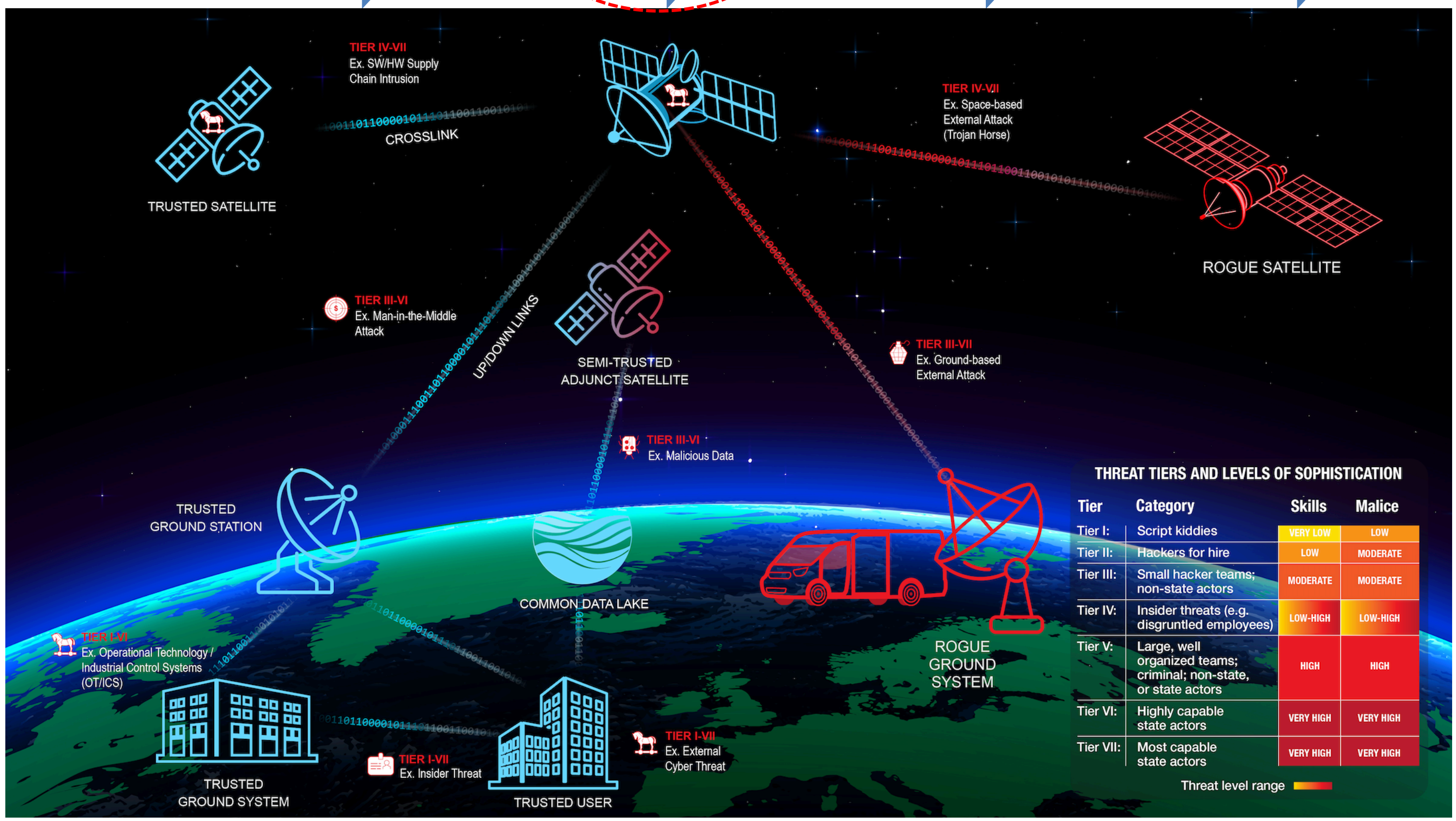


SPD-5[1] defines "Space System" as *"a combination of systems, to include ground systems, sensor networks, and one or more space vehicles, that provides a space-based service."*

SPD-5[1] states *Protection against unauthorized access to critical space vehicle functions. This should include safeguarding command, control, and telemetry links using effective and validated authentication or encryption measures designed to remain secure against **existing** and **anticipated** threats during the entire mission lifetime*

Attacks / TTPs can occur across all segments within a space system {i.e., ground, link, and space} to achieve the desired impact for the threat actor

TTP= Tactics, Techniques, & Procedures



Tier	Category	Skills	Malice
Tier I:	Script kiddies	VERY LOW	LOW
Tier II:	Hackers for hire	LOW	MODERATE
Tier III:	Small hacker teams; non-state actors	MODERATE	MODERATE
Tier IV:	Insider threats (e.g. disgruntled employees)	LOW-HIGH	LOW-HIGH
Tier V:	Large, well organized teams; criminal; non-state, or state actors	HIGH	HIGH
Tier VI:	Highly capable state actors	VERY HIGH	VERY HIGH
Tier VII:	Most capable state actors	VERY HIGH	VERY HIGH

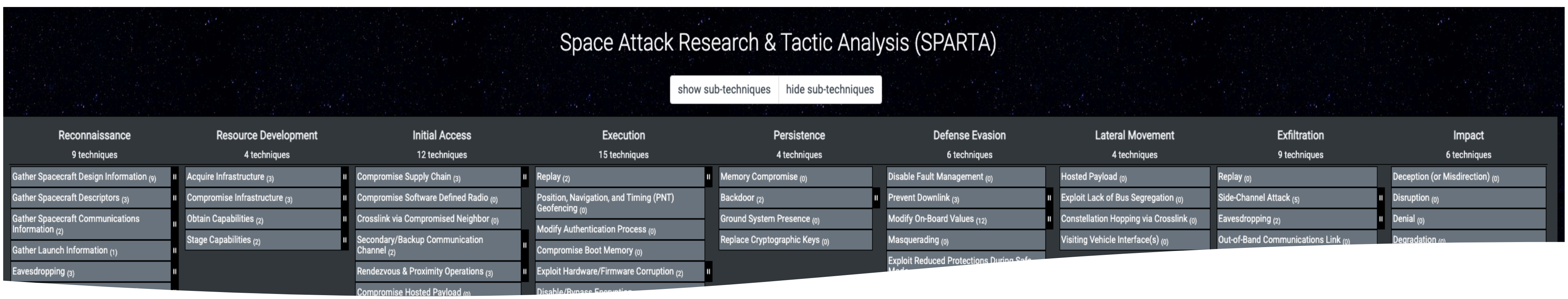
1 Memorandum on Space Policy Directive – 5 Cybersecurity Principles for Space Systems, Sep 2020

Space Attack Research & Tactic Analysis (SPARTA) – Launched Oct 2022

Filling the TTP Gap for Space

- Cybersecurity matrices are industry-standard tools and approaches for commercial and government users to navigate rapidly evolving cyber threats and vulnerabilities and outpace cyber threats
  - *They provide a critical knowledge base of adversary behaviors*
  - *Framework for adversarial actions across the attack lifecycle with applicable countermeasures*
- Current cybersecurity matrices (including MITRE ATT&CK) are limited to ground systems which lead to a gap!

- **Aerospace's SPARTA is the first-of-its-kind body of knowledge on cybersecurity protections for spacecraft and space systems, filling a critical vulnerability gap exists for the U.S. space enterprise**



SPARTA provides unclassified information to space professionals about how spacecraft may be compromised

International Collaboration

CyberInflight

- Expanding the reference section with CyberInflight's space security attacks database
  - *Working with them to map TTPs to increase the real-world examples of the TTPs in use by threat actors*
- Inclusion of their database deployed in July 2023 – v1.3.2
  - *https://sparta.aerospace.org/resources/updates-current*
- Since Oct 2022, received input from SPARTA from many government and commercial entities
  - *Including inputs from several international partners*

External Contributors

Special thanks to the following non-Aerospace Corporation individuals or organizations who have contributed to SPARTA content development and peer reviews:

- Gregory Falco
- Nick Tsamis
- Mario Zuniga
- Francesco Traini, Università Politecnica delle Marche
- Antonios Atlasis
- Ignacio Aguilar Sanchez
- Tim Dafoe
- Wayne Henry
- Andres Coronado
- Timothy O'Neill
- Florent Rizzo, CyberInflight's Market Intelligence Team
- Matthias Popoff, CyberInflight's Market Intelligence Team
- Héloise Do Nascimento Cardoso, CyberInflight's Market Intelligence Team

https://sparta.aerospace.org/contribute

Website Updates

- Updated TTP references using CyberInflight's Market Intelligence Team's space attack database
- Created Tools link to house Navigator and CM Mapper
- Fixed Navigator to work with other versions of SPARTA, but now all previously created JSON files are now obsolete
- Added 'Needed Countermeasures' to Navigator
- Updated Contribtors list

Techniques

New Techniques

Modified Techniques

- REC-0001: Gather Spacecraft Design Information
- REC-0002: Gather Spacecraft Descriptors
- REC-0003: Gather Spacecraft Communications Information
- REC-0004: Gather Launch Information
- REC-0008: Gather Supply Chain Information
- REC-0009: Gather Mission Information
- RD-0002: Compromise Infrastructure
- EX-0005: Exploit Hardware/Firmware Corruption
- EX-0013: Flooding
- EX-0014: Spoofing
- EXF-0007: Compromised Ground System
- EXF-0010: Payload Communication Channel
- IMP-0002: Disruption
- IMP-0003: Denial
- IMP-0004: Degradation
- IMP-0005: Destruction
- IMP-0006: Theft

Sub-Techniques

New Sub-Techniques

Modified Sub-Techniques

- REC-0003.01: Communications Equipment
- REC-0003.03: Mission-Specific Channel Scanning
- REC-0005.04: Active Scanning (RF/Optical)
- REC-0008.04: Business Relationships
- RD-0001.02: Commercial Ground Station Services
- EX-0013.02: Erroneous Input
- EX-0016.02: Downlink Jamming
- EXF-0003.02: Downlink Intercept

SPARTA Use Cases

Deep Dive on Some Use Cases

https://sparta.aerospace.org/resources/SPARTA_Overview_InDepth_Nov22.pdf

- Space system developers
  - *Engineers now have a resource that contains TTPs, threats, and countermeasures to enable the engineering of protections early in the lifecycle -- establishing countermeasures to disrupt the attack chains*
- Defensive Cyber Operations
  - *Enables the building of monitoring solutions, analytics, automation, etc. for DCO Operators/Blue Team members*
    - Measure how effective systems/operators are at detecting TTPs for their specific space system
      - *Ex: These commands/telemetry possibly indicate TTP attacking the software watchdog timer {EX-0012.11}*
- Threat intelligence reporting / tracking of TTPs
  - *Report data to the community tying threat actor's TTPs against space systems using a common taxonomy*
    - Leverage the unique identifiers and aggregate reporting using a similar approach as the current industry standard for Enterprise IT systems
- Assessments / Table-Tops
  - *Provides a framework for assessment engineers / red teamers to leverage for designing attack chains against the space segment*
- Education / Training / Research
  - *Expands the footprint of knowledge to a wider audience – raises the bar on what is considered common knowledge*

SPARTA will crowdsource info from space enterprise researchers and threat intel via sparta@aero.org

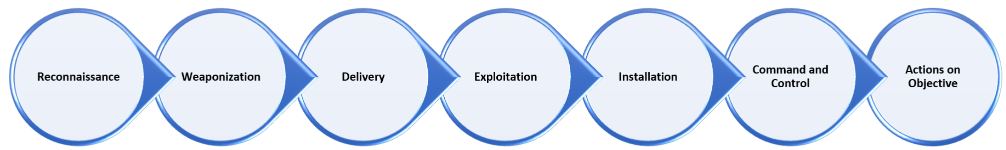
Attack Chain Development Can Support All Use Cases

Building Spacecraft Attack Chains using SPARTA

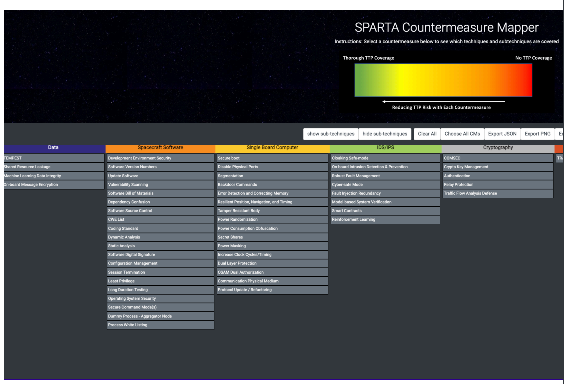
SPACE ATTACK RESEARCH & TACTIC ANALYSIS

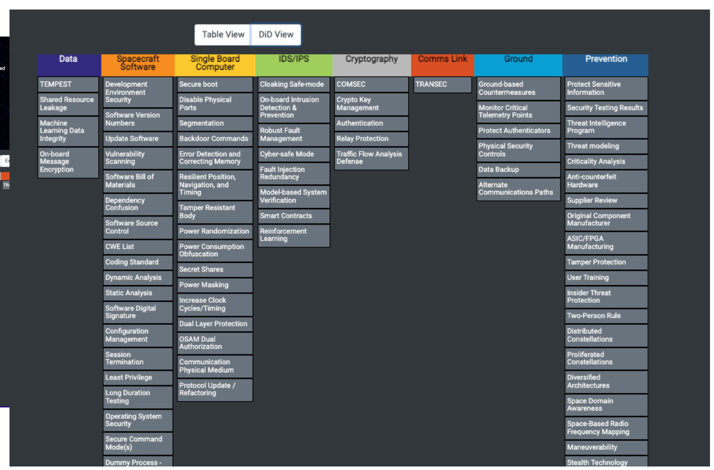
Attack Chains / Attack Flow != Cyber Kill Chain

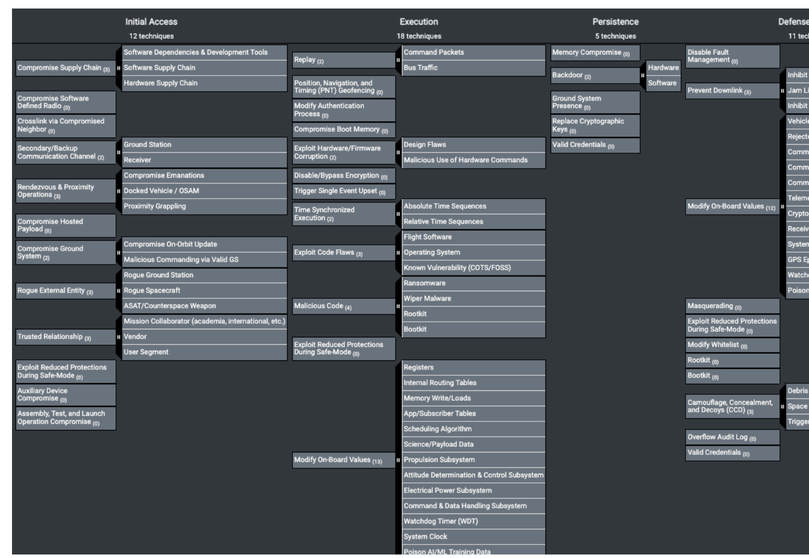
- Attack Chains help demonstrate exactly what an attacker is doing at every step of the way - in a simple and easy to understand visual story
  - *This is not Cyb*er Kill Chain which are stages comprising a cyberattack, geared towards "breaking" any phase of the "kill chain" which stop an attacker



- Attack Chains using ATT&CK and or SPARTA are **more than a sequence** of attack tactics
  - Knowledge base that correlates environment-specific (IT, OT/ICS, Cloud, Space) cybersecurity information along a hierarchy of TTP, and other knowledge (detections, mitigations, countermeasures, etc.)
- Ex: building the attack chains, especially in SPARTA, helps derive countermeasures | mapper







Building Spacecraft Attack Chains

SPARTA
SPACE ATTACK RESEARCH & TACTIC ANALYSIS

Blast from the Past
- Replay Attack from DefCon 2020
- Memory Injection Attack DefCon 2022

New Attacks
- Supply Chain Attack – Time bomb that executes command sequence 30 secs after boot
- Reaction Wheel Attack – Sending commands from rogue ground station due to no auth/encryption

CySat 2023
- ESA OPS-SAT Attack

Theoretical Attack Chain in Backup
- PCspooF

Resources to Help

- ATT&CK - https://attack.mitre.org/ -- if doing attack chains for IT/Enterprise/Ground Systems
  - *https://attack.mitre.org/docs/ATTACK_Design_and_Philosophy_March_2020.pdf*
  - *https://www.cisa.gov/sites/default/files/2023-01/Best%20Practices%20for%20MITRE%20ATTCK%20Mapping.pdf*
  - *https://github.com/cisagov/decider*
  - *https://center-for-threat-informed-defense.github.io/attack-flow/ui/*

- SPARTA - https://sparta.aerospace.org/resources/
  - *https://aerospacecorp.medium.com/sparta-cyber-security-for-space-missions-4876f789e41c*
  - *https://aerospace.org/article/leveraging-sparta-matrix*
  - *SPARTA can help educate on the types of space TTPs*
    - SPARTA tools like navigator can help visualize the attack chains - https://sparta.aerospace.org/navigator
    - SPARTA's countermeasure mapper helps understand how countermeasure impact TTPs https://sparta.aerospace.org/countermeasures/mapper

The steps below describe how to successfully map CTI reports to ATT&CK. Analysts may choose their own starting point (e.g., identification of tactics versus techniques) based on the information available and their knowledge of ATT&CK. Appendix B provides an annotated example of a cybersecurity advisory that incorporates ATT&CK.

ATT&CK Mapping for Finished Reports
Some Helpful Tips

1. Closely review images, graphics, and command line examples—these may depict additional techniques not explicitly called out in the report.
2. Use the [ATT&CK Navigator] tool to highlight the specific tactics and techniques. See MITRE's [Introduction to ATT&CK Navigator] video. **Note:** Navigator was defined for a number of use cases (from identifying defensive coverage gaps, to red/blue team planning, to highlighting the frequency of detected techniques.)
3. Double-check to determine if you accurately captured all ATT&CK mappings. Additional mappings are often missed on the first pass, even by the most experienced analysts.
4. Only limit mapping to the tactic level when there is insufficient detail to identify an applicable technique or sub-technique.

SPARTA Navigator

1. **Find the behavior.** Searching for signs of adversary behavior is a paradigm shift from looking for Indicators of Compromise (IOCs), hashes of malware files, URLs, domain names, and other artifacts of previous compromise. Look for signs of how the adversary interacted with specific platforms and applications to find a chain of anomalous or suspicious behavior. Try to identify how the initial compromise was achieved as well as how the post-compromise activity was performed. Did the adversary leverage legitimate system functions for malicious purposes, i.e., living off the land techniques?
2. **Research the Behavior.** Additional research may be needed in order to gain the required context to understand suspicious adversary or software behaviors.
   a. Look at the original source reporting to understand how the behavior was manifest in those reports. Additional resources may include reports from security vendors, U.S. government cyber organizations, international CERTS, Wikipedia, and Google.
   b. While not all of the behaviors may translate into techniques and sub-techniques, technical details can build on each other to inform an understanding of the overall adversary behavior and associated objectives.
   c. Search for key terms on the ATT&CK website to help identify the behaviors. One popular approach is to search for key verbs used in a report describing adversary behavior, such as "issuing a command," "creating persistence," "creating a scheduled task," "establishing a connection," or "sending a connection request."

SPARTA has search feature, but you can export all of SPARTA in JSON or Excel and that might be better option for searching.

3. **Identify the Tactics.** Comb through the report to identify the adversary tactics and the flow of the attack. To identify the tactics (the adversary's goals), focus on ***what*** the adversary was trying to accomplish and ***why***. Was the goal to steal the data? Was it to destroy the data? Was it to escalate privileges?
   a. Review the tactic definitions to determine how the identified behaviors might translate into a specific tactic. Examples might include:

      iii. Creates persistence by creating the following scheduled task.
      Tactic: *Persistence* [TA0003]
   b. Identify all of the tactics in the report. Each tactic includes a finite number of actions an adversary can take to implement their goal. Understanding the flow of the attack can help identify the techniques or sub-techniques that an adversary may have employed.
4. **Identify the Techniques.** After identifying the tactics, review the technical details associated with ***how*** the adversary tried to achieve their goals. For example, how did the adversary gain the *Initial Access* [TA0001] foothold? Was it through spearphishing or through an external remote service? Drill down on the range of possible techniques by reviewing the observed behaviors in the report. **Note:** if you have insufficient detail to identify an applicable technique, you will be limited to mapping to the tactic level, which alone is not actionable information for detection purposes.
   a. Compare the behavior in the report with the description of the ATT&CK techniques listed under the identified tactic. Does one of them align? If so, this is probably the appropriate technique.
   b. Be aware that multiple techniques may apply concurrently to the same behavior. For example, "HTTP-based Command and Control (C2) traffic over port 8088" would fall under both the *Non-Standard Port* [T1571] technique and *Web Protocols* [T1071.001] sub-techniques of *Application Layer Protocol* [T1071]. Mapping multiple techniques to a behavior concurrently allows the analyst to capture different technical aspects of behaviors, relate behaviors to their uses, and align behaviors to data sources and countermeasures that can be used by defenders.
   c. Do not assume or infer that a technique was used unless the technique is explicitly stated or there is no other technical way that a behavior could have occurred. In the "HTTP-based Command and Control (C2) traffic over port 8088" example, if the C2 traffic is over HTTP, an analyst should not assume the traffic is over port 80 because adversaries may use non-standard ports.
   d. Use the Search bar on the top left of the [ATT&CK website]—or CTRL+F on the [ATT&CK Enterprise Techniques web page]—to search for technical details, terms, or command lines to identify possible techniques that match the described behavior. For example, searching for a particular protocol might give insight into a possible technique or sub-technique.
   e. Ensure that the techniques align with the appropriate tactics. For example, there are two techniques that involve scanning. The *Active Scanning* [T1595] technique under the Reconnaissance tactic occurs ***before*** compromise of the victim. The technique describes active reconnaissance scans that probe victim infrastructure via network traffic

https://www.cisa.gov/sites/default/files/2023-01/Best%20Practices%20for%20MITRE%20ATTCK%20Mapping.pdf attack cycle. Because of this, techniques are often linked in the attack chain.

5. **Identify the Sub-techniques.** Review sub-technique descriptions to see if they match the information in the report. Does one of them align? If so, this is probably the right sub-technique. Depending upon the level of detail in the reporting, it may not be possible to identify the sub-technique in all cases. **Note:** map solely to the parent technique only if there is not enough context to identify a sub-technique.
   a. Read the sub-technique descriptions carefully to understand the differences between them. For example, *Brute Force* [T1110] includes four sub-techniques: *Password Guessing* [T1110.001], *Password Cracking* [T1110.002], *Password Spraying* [T1110.003], and *Credential Stuffing* [T1110.004]. If, for example, the report provides no additional context to identify the sub-technique that the adversary used, simply identify *Brute Force* [T1110]—which covers all methods for obtaining credentials—as the parent technique.
   b. In cases where the parent of a sub-technique aligns to multiple tactics, make sure to choose the appropriate tactic. For example, the *Process Injection: Dynamic-link Library Injection* [T1055.001] sub-technique appears in both *Defense Evasion* [TA0005] and *Privilege Escalation* [TA0004] tactics.
   c. If the sub-technique is not easily identifiable—there may not be one in every case—it can be helpful to review the procedure examples. The examples provide links to the source CTI reports that support the original technique mapping. The additional context may help affirm a mapping or suggest that an alternative mapping should be investigated. There is always a possibility that a behavior may be a new technique not yet covered in ATT&CK. For example, new techniques related to the SolarWinds supply chain compromise led to an out-of-cycle version modification to the ATT&CK framework. The ATT&CK team strives to include new techniques or sub-techniques as they become prevalent. Contributions from the community of security researchers and analysts help make this possible. Please notify the ATT&CK team if you are observing a new technique or sub-technique or new use of a technique.

Techniques and Sub-techniques
Read Descriptions Carefully

Differences in techniques and sub-techniques are often subtle. Make sure to read the detailed descriptions of these thoroughly before making a determination.

For example, *Obfuscated Files or Information: Software Packing* [T1027.002] (compressing or encrypting an executable) differs from *Data Encoding* [T1132], which involves adversaries encoding data to make the content of command and control traffic more difficult to detect. The tactics differ as well: *Software Packing* is used to achieve the *Defense Evasion* [TA0005] tactic and *Data Encoding* is aligned to the *Command and Control* [TA0011] tactic.

Another example: *Masquerading* [T1036] refers to general masquerading attempts, while *Masquerading: Masquerade Task or Service* [T1036-004] specifically refers to the impersonation of a system task or service, as opposed to files.

6. **Compare your Results to those of Other Analysts.** Improve your mappings by collaborating with other analysts. Working with other analysts on mappings lends diversity of viewpoints and helps inform additional perspectives that can raise awareness of possible analyst bias. A formal process of peer review and consultation can be an effective means to share perspectives, promote learning, and improve results. A peer review of a report annotated with the proposed tactic, techniques, and sub-techniques can result in a more accurate mapping of TTPs missed in the initial analysis. This process can also help to improve consistency of mapping throughout the team.

ATT&CK Mapping is a Team Sport
Some Helpful Tips

1. Work as a team to identify ATT&CK techniques. Input from multiple analysts with different backgrounds increases the accuracy of the mapping, reduces bias, and may lead to additional techniques being identified.
2. Perform a peer review. Even with highly experienced team members, the MITRE ATT&CK team conducts at least two reviews of new mapping content before any public release.

The following pages contain an example of a finished report that incorporates:

1. **In-line ATT&CK TTP links** as part of the narrative to flag the presence of an ATT&CK TTP. In-line ATT&CK mapping helps the reader to understand the activity as they are reading the report.[6]
2. **Summary ATT&CK tables** that identify the ATT&CK technique ID, the name, and context (i.e., details about the adversary's use of the particular technique). Analysts should provide enough information in the context section that the audience can understand the rationale for the ATT&CK mapping and, ideally, what it means for their own organization. Summary tables allow the reader to quickly scan and identify techniques or sub-techniques of concern or interest.
3. **ATT&CK Navigator Visualization** to codify the adversary tactics and techniques. Visualizations can be used to 1) summarize all of the adversary's activities, 2) highlight TTPs that are unique to an adversary, or 3) to compare and contrast multiple adversary TTPs.
4. **Permalinks**, which include the version (e.g., https://attack.mitre.org/versions/**v8**/techniques/T1105/) for all TTP links to ensure these will endure version changes of ATT&CK.
5. The corresponding **parent technique** into any reference of a **sub-technique**. **Note:** this is an especially good practice when referencing sub-techniques that have the same name.

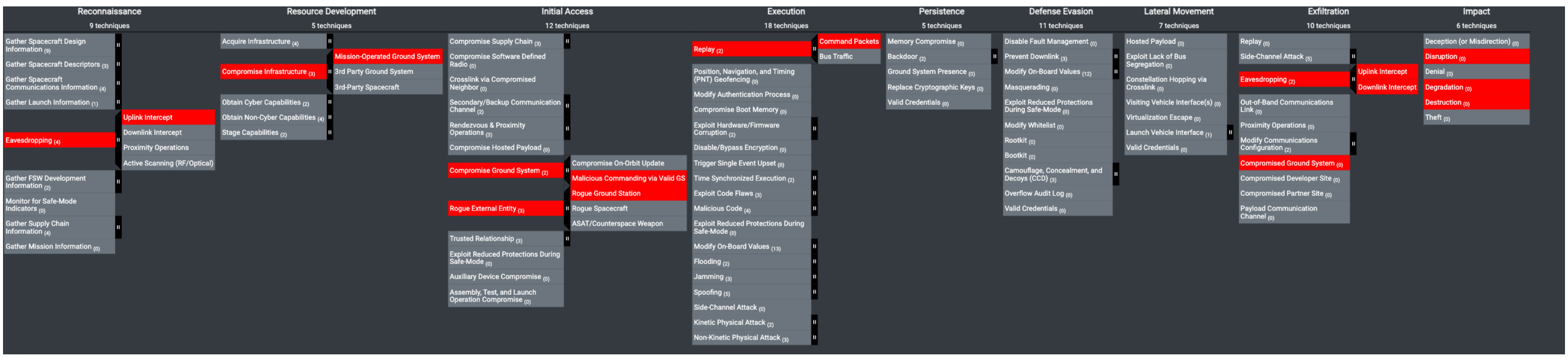
https://www.cisa.gov/sites/default/files/2023-01/Best%20Practices%20for%20MITRE%20ATTCK%20Mapping.pdf

Example Attack Chains from the Past

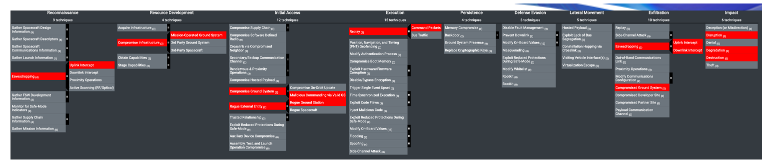
SPACE ATTACK RESEARCH & TACTIC ANALYSIS

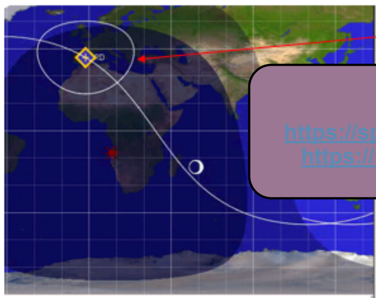
DefCon 2020 – Exploiting Spacecraft Example (https://www.youtube.com/watch?v=b8QWNiqTx1c)

Attacker performs a man-in-the-middle attack at the ground station where they record command packets in the UDP traffic [REC-0005 , RD-0005.01] for replaying to the spacecraft [EX-0001.01]. In this example UDP mimics the radio frequency link. This same attack could be applied through RF signal sniffing [REC-0005.01, IA-0008.01] vice UDP captures. From the spacecraft perspective, the flight software processes the traffic whether or not the traffic is coded to radio frequency signals and then decoded on the spacecraft. Upon receiving commands, the spacecraft flight software responds by downlinking command counter data to the ground indicating that commands were received [EXF-0003.02]. In this scenario, the attacker collected the commands at the ground station [EXF-0003.01, EXF-0007] and then promptly replay the traffic to the spacecraft [EX-0001.01] thereby causing the flight software to reprocess the commands again [EX-0001]. This would be visible in the downlinked command counters [REC-0005.02, EXF-0003.02] and unless the ground operators are monitoring specific telemetry points, this attack would likely go unnoticed. If the replayed commands were considered critical commands like firing thrusters, then more critical impact on the spacecraft could be encountered [IMP-0002, IMP-0004, IMP-0005].



Replay Attack & Command Link Intrusion



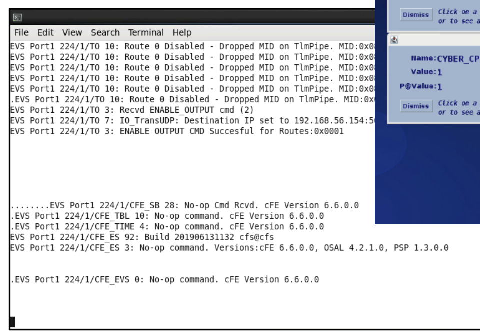


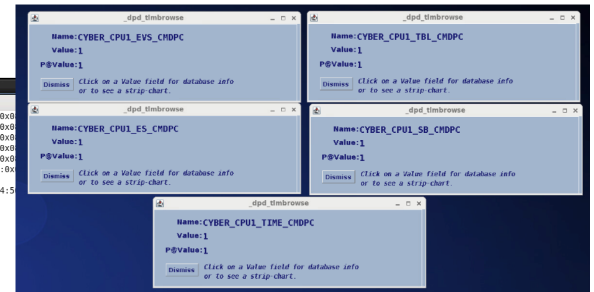
Satellite visible to ground station

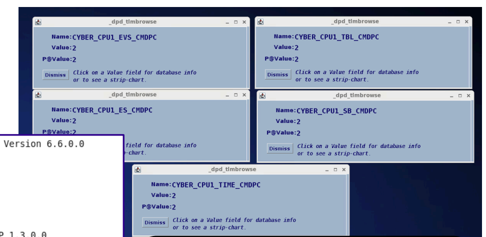
Eavesdropping
https://sparta.aerospace.org/technique/REC-0005/01/
https://sparta.aerospace.org/technique/EXF-0003/

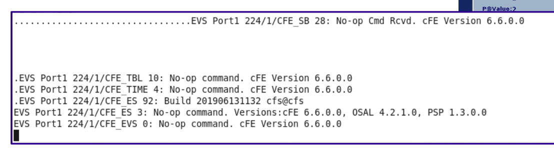
Signal Lock

Commands being transmitted

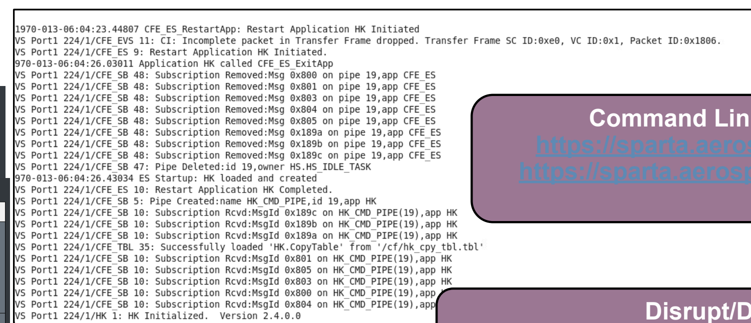








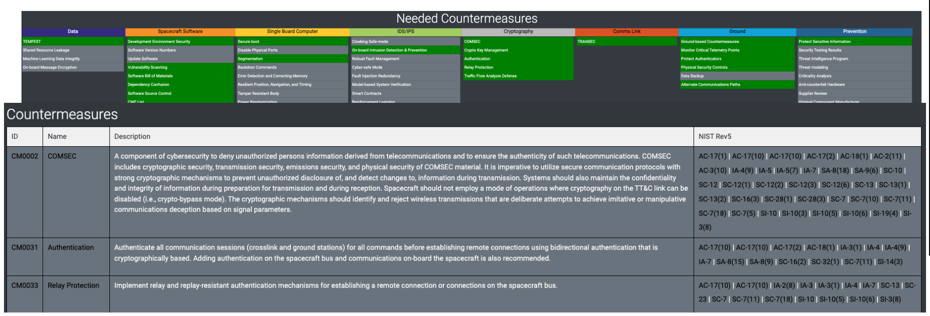
Replay
https://sparta.aerospace.org/technique/EX-0001/



Command Link Intrusion from Ground
https://sparta.aerospace.org/technique/IA-0007/
https://sparta.aerospace.org/technique/IA-0008/01/

Disrupt/Degradation
https://sparta.aerospace.org/technique/IMP-0002/
https://sparta.aerospace.org/technique/IMP-0004/

Example SPARTA Countermeasures



ID	Name	Description	NIST Rev5
CM0002	COMSEC	A component of cybersecurity to deny unauthorized persons information derived from telecommunications and to ensure the authenticity of such telecommunications. COMSEC includes cryptographic security, transmission security, emissions security, and physical security of COMSEC material. It is imperative to utilize secure communication protocols with strong cryptographic mechanisms to prevent unauthorized disclosure of, and detect changes to, information during transmission. Systems should also maintain the confidentiality and integrity of information during preparation for transmission and during reception. Spacecraft should not employ a mode of operations where cryptography on the TT&C link can be disabled (i.e., crypto-bypass mode). The cryptographic mechanisms should identify and reject wireless transmissions that are deliberate attempts to achieve imitative or manipulative communications deception based on signal parameters.	AC-17(1) | AC-17(10) | AC-17(10) | AC-17(2) | AC-18(1) | AC-2(11) | AC-3(10) | IA-4(9) | IA-5 | IA-5(7) | IA-7 | SA-8(18) | SA-9(6) | SC-10 | SC-12 | SC-12(1) | SC-12(2) | SC-12(3) | SC-12(6) | SC-13 | SC-13(1) | SC-13(2) | SC-16(3) | SC-28(1) | SC-28(3) | SC-7 | SC-7(10) | SC-7(11) | SC-7(18) | SC-7(5) | SI-10 | SI-10(3) | SI-10(5) | SI-10(6) | SI-19(4) | SI-3(8)
CM0031	Authentication	Authenticate all communication sessions (crosslink and ground stations) for all commands before establishing remote connections using bidirectional authentication that is cryptographically based. Adding authentication on the spacecraft bus and communications on-board the spacecraft is also recommended.	AC-17(10) | AC-17(10) | AC-17(2) | AC-18(1) | IA-3(1) | IA-4 | IA-4(9) | IA-7 | SA-8(15) | SA-8(9) | SC-16(2) | SC-32(1) | SC-7(11) | SI-14(3)
CM0033	Relay Protection	Implement relay and replay-resistant authentication mechanisms for establishing a remote connection or connections on the spacecraft bus.	AC-17(10) | AC-17(10) | IA-2(8) | IA-3 | IA-3(1) | IA-4 | IA-7 | SC-13 | SC-23 | SC-7 | SC-7(11) | SC-7(18) | SI-10 | SI-10(5) | SI-10(6) | SI-3(8)

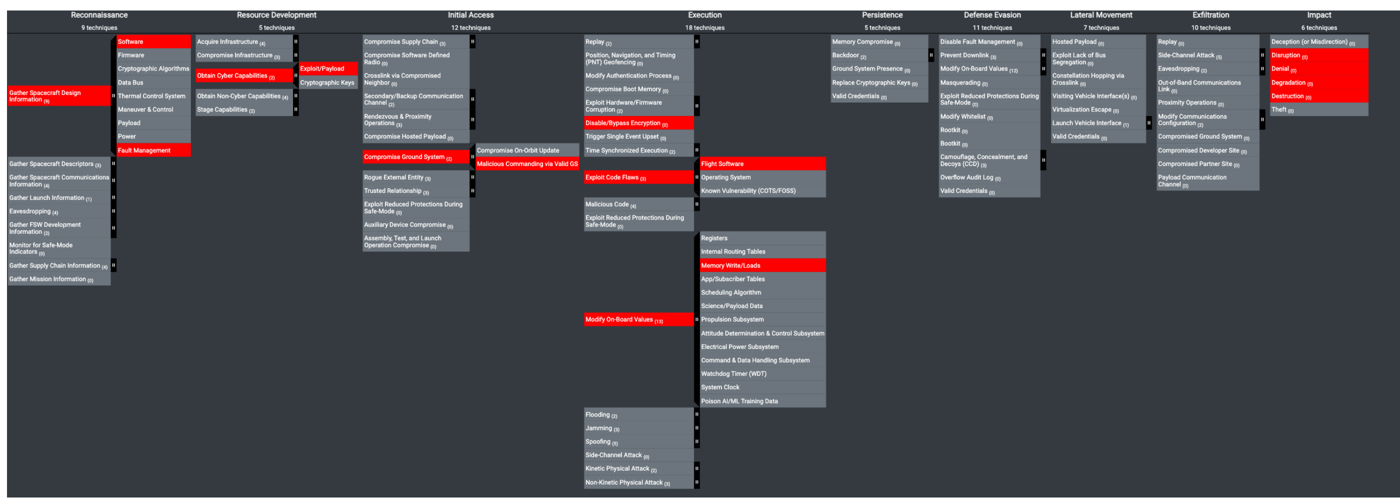
Example Attack Chains from the Past

SPARTA
SPACE ATTACK RESEARCH & TACTIC ANALYSIS

DefCon 2022 - Memory Manipulation Attack (https://www.youtube.com/watch?v=t_efCpd2PbM)

This example requires significant effort in the reconnaissance phase [REC-0001, REC-0003] to understand the specific attack vectors. However, after understanding the memory maps/locations and how the VxWorks and PowerPC interrelates, the attack can be performed to disrupt [IMP-0002] and deny [IMP-0003] the spacecraft's ability to process information. Upon performing all the necessary research, a single command packet is all that is required to affect the spacecraft. Understanding the precise memory location and overwriting it with desired values, exploits the inherit trust between the ground and the spacecraft [IA-0009].

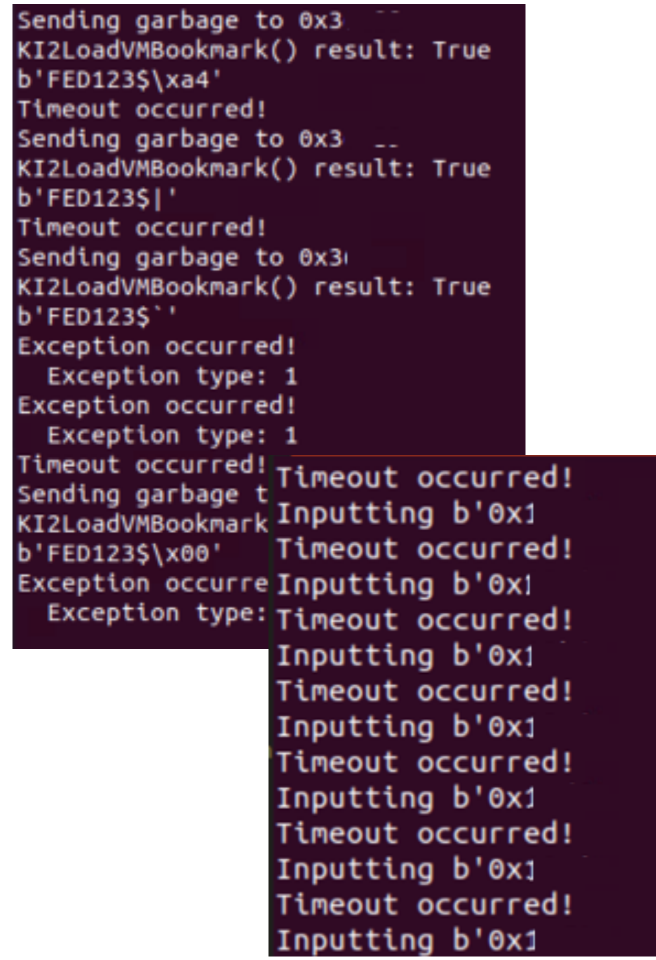
In this exploit example, the attacker leverages the authenticated/encrypted command pathway to send two commands to the spacecraft [IA-0007.02, EX-0006]. A simple NO-OP for demonstration purposes followed by a "magic packet" or "kill-pill" that corrupts the running state of the PowerPC processor thereby disabling the spacecraft's ability to process information. The below figure shows redacted information to remove the actual corrupting content, but the "vxworks!" is essentially the kernel throwing a panic and crashing. This is where having direct memory access [EX-0012.03] via the spacecraft flight software can be dangerous and must be protected [EX-0009.01]. There are many instances where the ground can issue legitimate commands to degrade/deny/destroy [IMP-0004, IMP-0003, IMP-0005] the spacecraft which puts pressure on fault management to account for this truth [REC-0001.09].



Fuzzing Memory Addresses

Lots of Trial and Error

- Hardware design documentation reveals "features" of hardware design
  - *Can these features be leveraged for nefarious purposes?*
    - Creating faults, abusing functions, etc. from design docs are common TTPs when performing aggression on spacecraft technology
- Lots of debugging and reverse engineering later
  - *Setting breakpoints, working with registers, memory regions, etc.*
    - Digital twins come in extremely handy during this research
      - *See: Hunting for Spacecraft Zero Days using Digital Twins*
  - *Triggering exceptions and understanding what they mean*



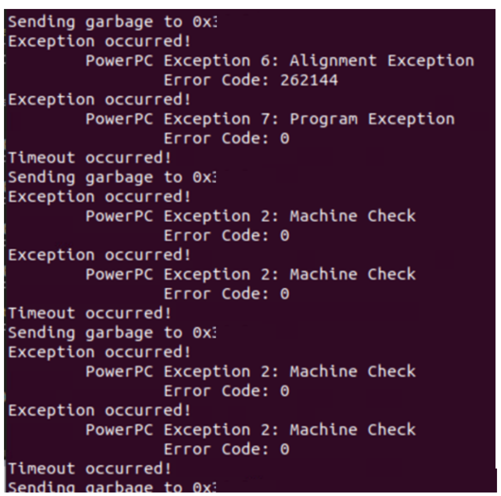


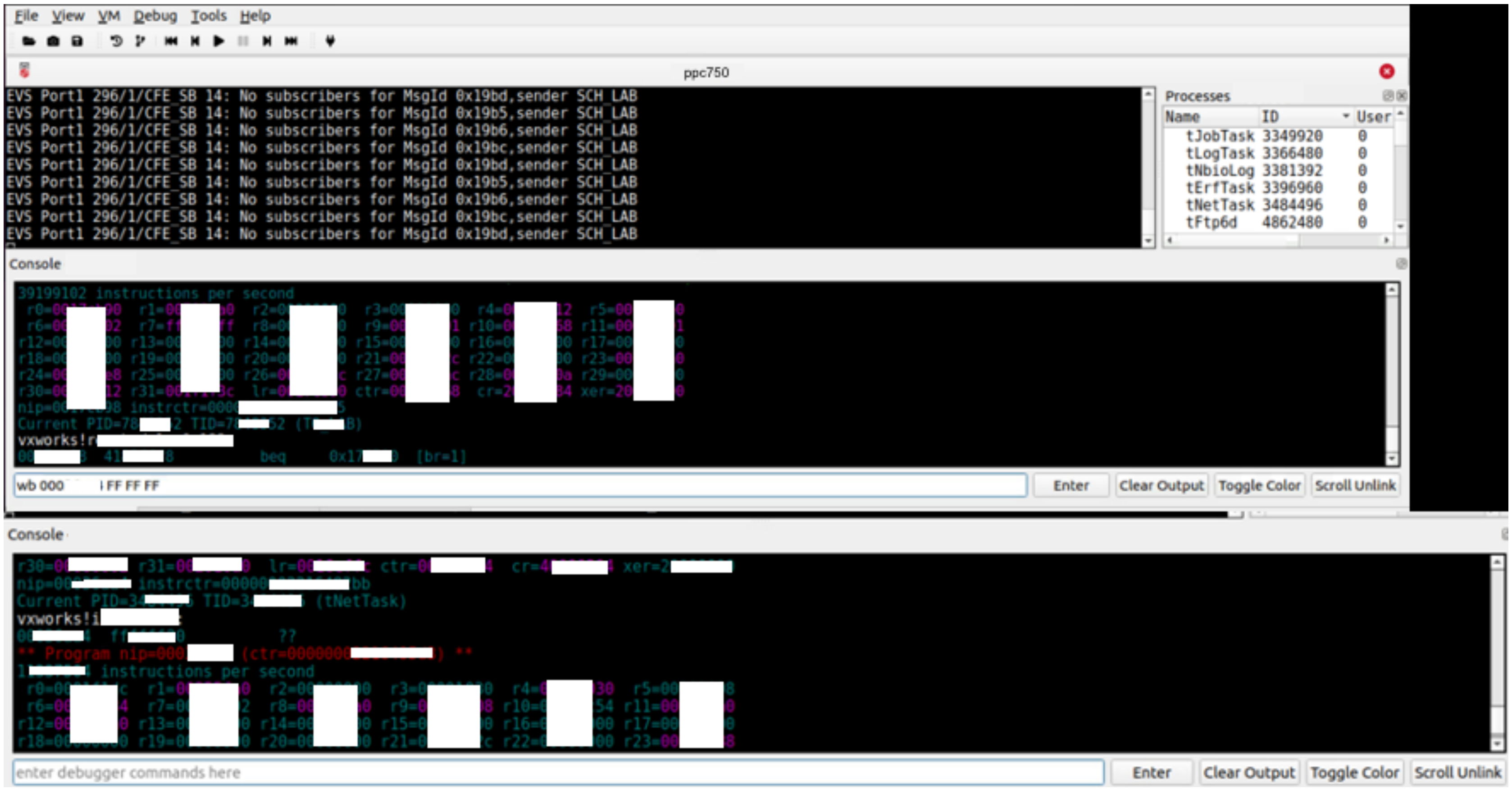
Table 6-2. Exceptions and Conditions—Overview

Exception Type	Vector Offset (hex)	Causing Conditions
Reserved	00000	—
System reset	00100	The causes of system reset exceptions are *implementation-dependent*. If the conditions that cause the exception also cause the processor state to be corrupted such that the contents of SRR0 and SRR1 are no longer valid or such that other processor resources are so corrupted that the processor cannot reliably resume execution, the copy of the RI bit copied from the MSR to SRR1 is *cleared*.
Machine check	00200	The causes for machine check exceptions are implementation-dependent, but typically these causes are related to conditions such as bus parity errors or attempting to access an invalid physical address. Typically, these exceptions are triggered by an input signal to the processor. Note that not all processors provide the same level of error checking. The machine check exception is disabled when MSR[ME] = 0. If a machine check exception condition exists and the ME bit is cleared, the processor goes into the checkstop state. If the conditions that cause the exception also cause the processor state to be corrupted such that the contents of SRR0 and SRR1 are no longer valid or such that other processor resources are so corrupted that the processor cannot reliably resume execution, the copy of the RI bit written from the MSR to SRR1 is cleared. (Note that physical address is referred to as real address in the architecture specification.)
DSI	00300	A DSI exception occurs when a data memory access cannot be performed for any of the reasons described in Section 6.4.3, "DSI Exception (0x00300)." Such accesses can be generated by load/store instructions, certain memory control instructions, and certain cache control instructions.
ISI	00400	An ISI exception occurs when an instruction fetch cannot be performed for a variety of reasons described in Section 6.4.4, "ISI Exception (0x00400)."
External interrupt	00500	An external interrupt is generated only when an external interrupt is pending (typically signalled by a signal defined by the implementation) and the interrupt is enabled (MSR[EE] = 1).
Alignment	00600	An alignment exception may occur when the processor cannot perform a memory access for reasons described in Section 6.4.6, "Alignment Exception (0x00600)." Note that an implementation is allowed to perform the operation correctly and not cause an alignment exception.

https://www.nxp.com/docs/en/user-guide/MPCFPE_AD_R1.pdf

Manually Invoking Crash – Post Fuzzing

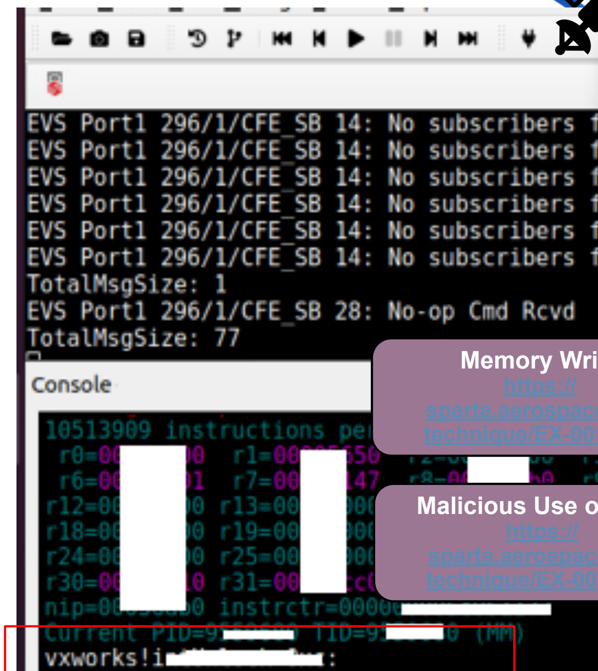
Confirming Input Results Provides Desired Reaction



Initiating the Crash from the Ground

Mapping the TTPs

- Sending No-Op followed by Magic Packet to crash the spacecraft processor
  - *This is where having direct memory access via the spacecraft FSW can be dangerous and must be protected*
    - The inherit trust between ground systems and spacecraft MUST be accounted for and better protections on-board the spacecraft are necessary moving forward
      - *Too many instances where the ground can issue legitimate commands to degrade/deny/destroy the spacecraft*
        - Must extend fault management to account for this truth

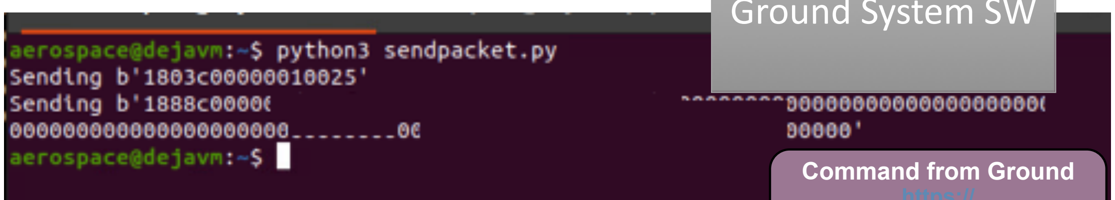


Memory Write https://sparta.aerospace.org/technique/EX-0012/03/

Malicious Use of FSW https://sparta.aerospace.org/technique/EX-0009/01/

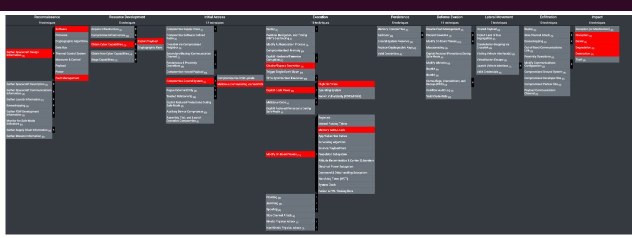
Space-Ground Link (UDP)

Ground System SW

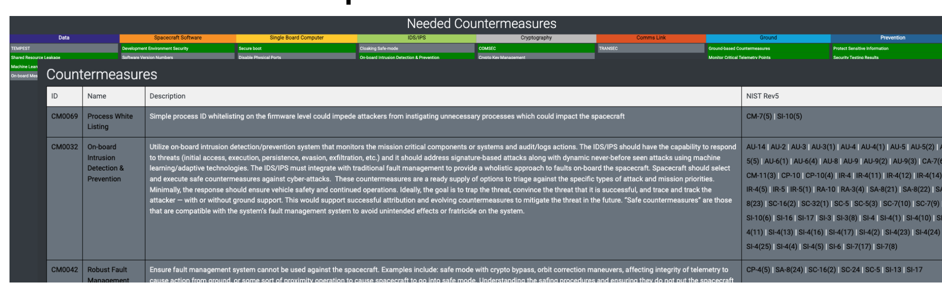


Disrupt/Denial https://sparta.aerospace.org/technique/IMP-0002/ https://sparta.aerospace.org/technique/IMP-0003/

Command from Ground https://sparta.aerospace.org/technique/IA-0007/02/



Example SPARTA Countermeasures



Supply Chain Injection – Boot Sequence (RTS)

RTS001 loads after boot

2.2.7 RTS Tables

RTS tables are a sequence of Relative Time Sequence commands. The purpose of Relative Time Sequence commands is to be able to specify commands to be executed at a specific time *after* ("relative to") an ATS.

For Relative Time Command Sequence commands there is a field that represents the time in seconds that the command will *delay* before executing. This delay is relative to the time when the previous Relative Time Tagged Command (RTC) was executed. In the case of the first command of the sequence, this time is relative to when the sequence was started.

More details of timing and format for RTS tables are shown in Chapter 3.

3.4.5 Naming Conventions for RTSs

Because RTSs can be loaded at startup, the files for those RTSs must be in a predetermined location (CFS SC Configuration Parameter SC_RTS_FILE_NAME).

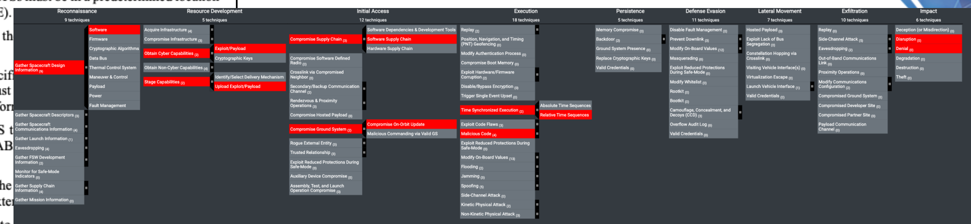
This location must be in non-volatile memory. Otherwise, th
On reset.

Also, the RTS table file must be named according to a specif
Parameter SC_RTS_TABLE_NAME). The file name must
Configuration Parameter SC_RTS_TABLE_NAME) platfor

Next, must be a three digit number indicating which RTS t
".tbl". An example of this for RTS No.1, with SC_RTS_TAB
be: 'RTS_TBL001.tbl'.

In addition to the file naming convention, the name of the
should be the same as the file name, without the path or exte

Remember to also have the application name prefixed to the name of the table. For the file 'RTS_TBL001.tbl', its table name should be 'SC.RTS_TBL001, if the name of the application is "SC".



RTS001

```
/*
** RTS Table Data
*/
uint16 RTS_Table001[SC_RTS_BUFF_SIZE] =
{
/* cmd time,  <---------------------------- cmd pkt primary header ---------------------------->    <----- cmd pkt 2nd header ---->        <-- opt data --->
  1,          CFE_MAKE_BIG16(DS_CMD_MID),         CFE_MAKE_BIG16(PKT_FLAGS), CFE_MAKE_BIG16(5),    CFE_MAKE_BIG16(DS_SET_APP_STATE_CC),    0x0001, 0x0000, /
  1,          CFE_MAKE_BIG16(TO_LAB_CMD_MID),     CFE_MAKE_BIG16(PKT_FLAGS), CFE_MAKE_BIG16(21),   CFE_MAKE_BIG16(TO_DEBUG_ENABLE_CC),     0x0031, 0x3237, 0
  1,          CFE_MAKE_BIG16(SAMPLE_APP_CMD_MID), CFE_MAKE_BIG16(PKT_FLAGS), CFE_MAKE_BIG16(1),    CFE_MAKE_BIG16(SAMPLE_APP_NOOP_CC),     // Sample Instrum
  5,          CFE_MAKE_BIG16(LC_CMD_MID),         CFE_MAKE_BIG16(PKT_FLAGS), CFE_MAKE_BIG16(5),    CFE_MAKE_BIG16(LC_SET_LC_STATE_CC),     0x0001, 0x0000, /

};
```

Compromise Supply Chain: Software Supply Chain
https://sparta.aerospace.org/technique/IA-0001/02/

```
** RTS Table Data
*/
uint16 RTS_Table001[SC_RTS_BUFF_SIZE] =
{
/* cmd time,  <----------------------------- cmd pkt primary header ----------------------------->    <----- cmd pkt 2nd header ---->       <-- opt data ---> */
  1,          CFE_MAKE_BIG16(DS_CMD_MID),        CFE_MAKE_BIG16(PKT_FLAGS), CFE_MAKE_BIG16(5),     CFE_MAKE_BIG16(DS_SET_APP_STATE_CC),   0x0001, 0x0000, // Enable DS
  1,          CFE_MAKE_BIG16(TO_LAB_CMD_MID),    CFE_MAKE_BIG16(PKT_FLAGS), CFE_MAKE_BIG16(21),    CFE_MAKE_BIG16(TO_DEBUG_ENABLE_CC),    0x0031, 0x3237, 0x2E30, 0x2E30, 0x2E31, 0x
  1,          CFE_MAKE_BIG16(SAMPLE_CMD_MID),    CFE_MAKE_BIG16(PKT_FLAGS), CFE_MAKE_BIG16(1),     CFE_MAKE_BIG16(SAMPLE_APP_NOOP_CC),    // Sample Instrument NOOP
  5,          CFE_MAKE_BIG16(LC_CMD_MID),        CFE_MAKE_BIG16(PKT_FLAGS), CFE_MAKE_BIG16(5),     CFE_MAKE_BIG16(LC_SET_LC_STATE_CC),    0x0001, 0x0000, // Enable LC
  6,          CFE_MAKE_BIG16(0x18A9),         CFE_MAKE_BIG16(PKT_FLAGS), CFE_MAKE_BIG16(1),     CFE_MAKE_BIG16(0x0000),         // SC NOOP - Test Command
  7,          CFE_MAKE_BIG16(0x1806),            CFE_MAKE_BIG16(PKT_FLAGS), CFE_MAKE_BIG16(3),     CFE_MAKE_BIG16(0x0200),      0x0002 //Reset ATTACK
```

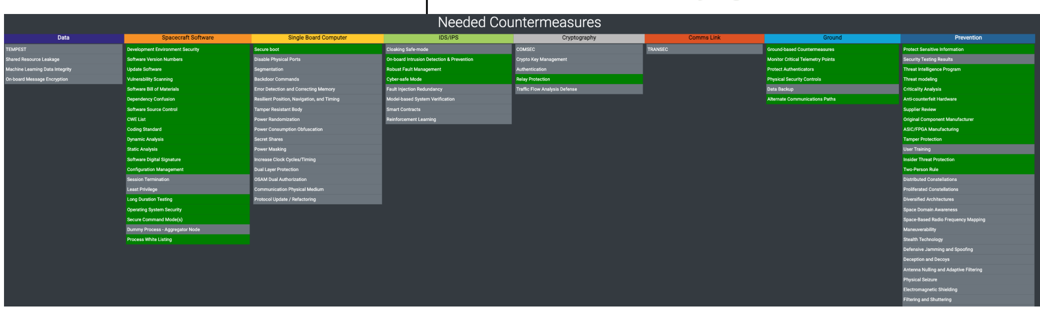
```
EVS Port1 42/1/SC 73: RTS Number 001 Started
EVS Port1 42/1/SCH 21: Major Frame Sync too noisy (Slot 1). Disabling synchronization.
EVS Port1 42/1/TO_LAB 3: TO telemetry output enabled for IP 1
EVS Port1 42/1/SAMPLE 11: SAMPLE: NOOP command received
EVS Port1 42/1/LC 20: Set LC state command: new state = 1
EVS Port1 42/1/SC 52: No-op command. Version 2.5.0.0
EVS Port1 42/1/SC 06: RTS 001 Execution Completed
2000-001-00:00:24.26000 POWERON RESET called from CFE_ES_ResetCFE (Commanded).
CFE_PSP: Exiting cFE with POWERON Reset status.
CFE_PSP: Critical Data Store Shared memory segment removed
Reset Area Shared memory segment removed
User Reserved Area Shared memory segment removed
```

Reboot command but could be "anything" – like reaction wheels?

Inject Malicious Code & Time Synchronized Execution: Relative Time Sequences
https://sparta.aerospace.org/technique/EX-0010/
https://sparta.aerospace.org/technique/EX-0008/02/

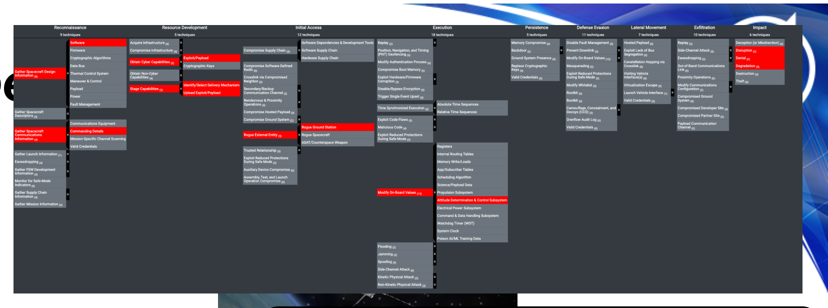
```
r failed to receive message, connection was shutdown in NosEngine::Transport::Connection::process_stop(std::uni
r caught exception while receiving message, will stop receiving, message connection was shutdown, before async
r failed to receive message, connection was shutdown in NosEngine::Transport::Connection::process_stop(std::uni
r caught exception while receivin
r failed to receive message, conn
r caught exception while receiving
```

Disrupt/Denial
https://sparta.aerospace.org/technique/IMP-0002/
https://sparta.aerospace.org/technique/IMP-0003/



Rogue Ground Station – Attacking Reaction Wheel

Spinning a CubeSat Uncontrollably



- Many CubeSats do not implement strong, sometimes any, authentication / encryption – therefore, can could be vulnerable to command link intrusion from Rogue Ground Station
- Requires reconnaissance on spacecraft

Modify On-Board Values: Attitude Determination & Control
https://sparta.aerospace.org/technique/EX-0012/08/

1992c000000303001400

Gather Spacecraft Design Information: Software
https://sparta.aerospace.org/technique/REC-0001/01/

Gather Spacecraft Communications Information: Commanding Details
https://sparta.aerospace.org/technique/REC-0003/02/

Rogue Ground System SW

https://github.com/nasa/nos3

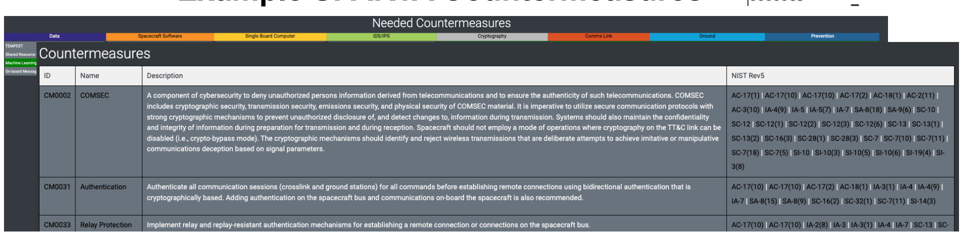
Command Link Intrusion from Rogue Ground
https://sparta.aerospace.org/technique/IA-0008/01/

- This attack creates a CCSDS frame to send to spacecraft from a rogue ground station

```
0000000 0d0a 0a0d 0060 0000 3c4d 1a2b 0001 0000
0000010 ffff ffff ffff ffff 0004 003a 6445 7469
0000020 6163 2070 5728 7269 7365 6168 6b72 2029
0000030 2e33 2e32 2033 4728 7469 7620 2e33 2e32
0000040 2033 6170 6b63 6761 6465 6120 2073 2e33
0000050 2e32 2d33 2931 0000 0000 0000 0060 0000
0000060 0001 0000 0014 0000 0001 0000 0000 0004
0000070 0014 0000 0006 0000 0054 0000 0000 0000
0000080 f7a5 0005 23d7 faa0 0032 0000 0032 0000
0000090 0000 0000 0000 0000 0000 0000 0008 0045
00000a0 2400 58a6 0040 1140 6e96 007f 0100 007f
00000b0 0100 acbc 9413 1000 23fe 9219 00c0 0300
00000c0 0003 0014 0054 0000
00000c8
```

Disrupt/Denial/Degrade
https://sparta.aerospace.org/technique/IMP-0002/
https://sparta.aerospace.org/technique/IMP-0003/
https://sparta.aerospace.org/technique/IMP-0004/

Example SPARTA Countermeasures



Needed Countermeasures

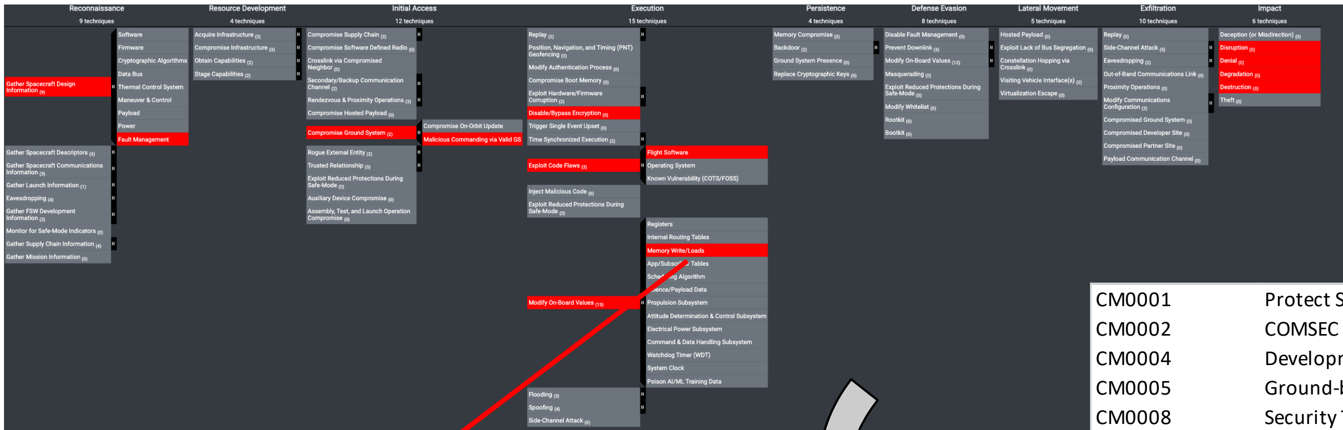
Data | Spacecraft Software | Single Board Computer | IDS/IPS | Cryptography | Comms Link | Ground | Prevention

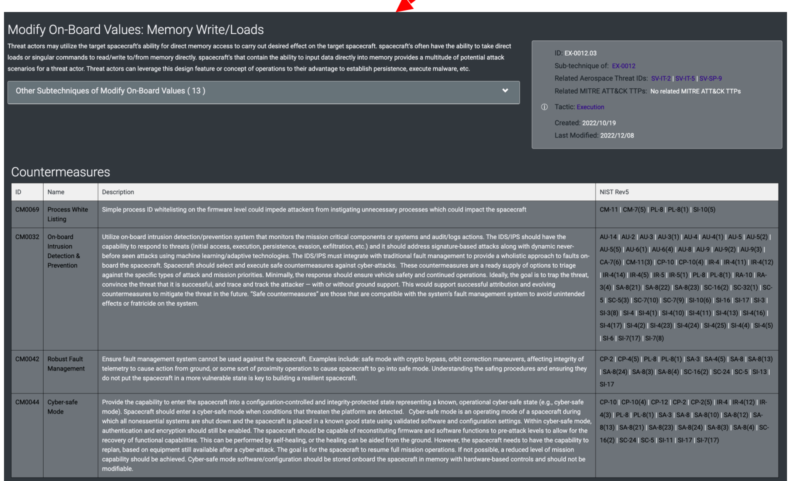
TEMPEST | Shared Resource | Machine Learning | On-board Messag

Countermeasures

ID	Name	Description	NIST Rev5
CM0002	COMSEC	A component of cybersecurity to deny unauthorized persons information derived from telecommunications and to ensure the authenticity of such telecommunications. COMSEC includes cryptographic security, transmission security, emissions security, and physical security of COMSEC material. It is imperative to utilize secure communication protocols with strong cryptographic mechanisms to prevent unauthorized disclosure of, and detect changes to, information during transmission. Systems should also maintain the confidentiality and integrity of information during preparation for transmission and during reception. Spacecraft should not employ a mode of operations where cryptography on the TT&C link can be disabled (i.e., crypto-bypass mode). The cryptographic mechanisms should identify and reject wireless transmissions that are deliberate attempts to achieve imitative or manipulative communications deception based on signal parameters.	AC-17(1) | AC-17(10) | AC-17(10) | AC-17(2) | AC-18(1) | AC-2(11) | AC-3(10) | IA-4(9) | IA-5 | IA-5(7) | IA-7 | SA-8(18) | SA-9(6) | SC-10 | SC-12 | SC-12(1) | SC-12(2) | SC-12(3) | SC-12(6) | SC-13 | SC-13(1) | SC-13(2) | SC-16(3) | SC-28(1) | SC-28(3) | SC-7 | SC-7(10) | SC-7(11) | SC-7(18) | SC-7(5) | SI-10 | SI-10(3) | SI-10(5) | SI-10(6) | SI-19(4) | SI-3(8)
CM0031	Authentication	Authenticate all communication sessions (crosslink and ground stations) for all commands before establishing remote connections using bidirectional authentication that is cryptographically based. Adding authentication on the spacecraft bus and communications on-board the spacecraft is also recommended.	AC-17(10) | AC-17(10) | AC-17(2) | AC-18(1) | IA-3(1) | IA-4 | IA-4(9) | IA-7 | SA-8(15) | SA-8(9) | SC-16(2) | SC-32(1) | SC-7(11) | SI-14(3)
CM0033	Relay Protection	Implement relay and replay-resistant authentication mechanisms for establishing a remote connection or connections on the spacecraft bus.	AC-17(10) | AC-17(10) | IA-2(8) | IA-3 | IA-3(1) | IA-4 | IA-7 | SC-13 | SC-

Mapping Attack Chain to Countermeasures





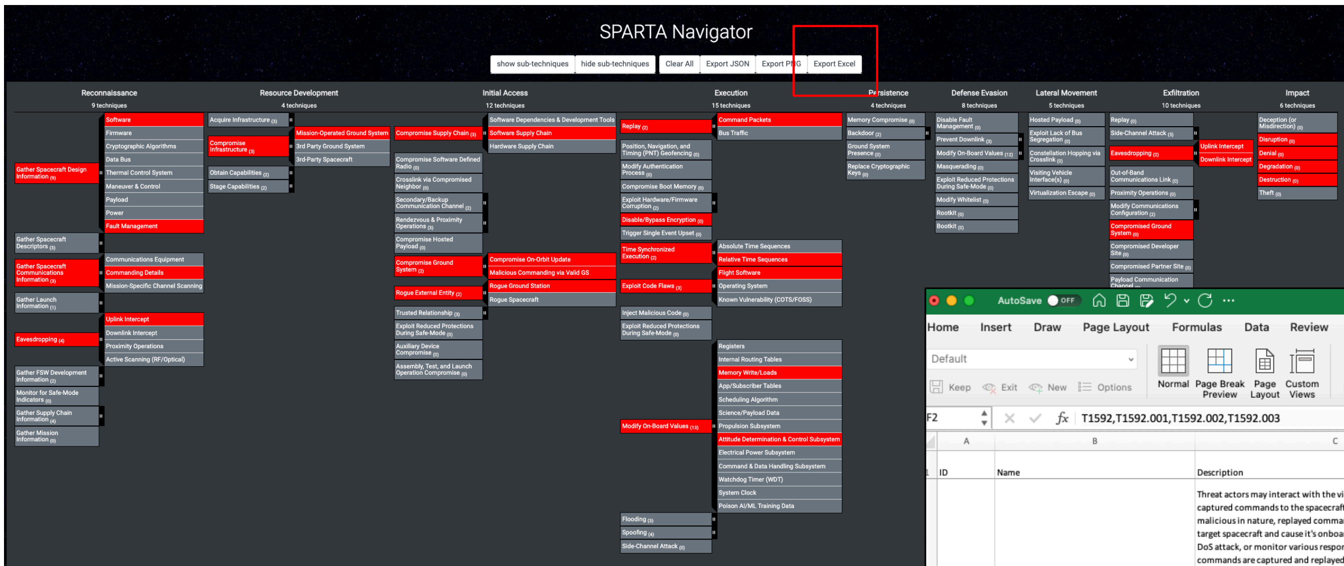
SPARTA has direct mapping from TTP to Countermeasures

Many of these countermeasures likely not feasible for mission that are already launched

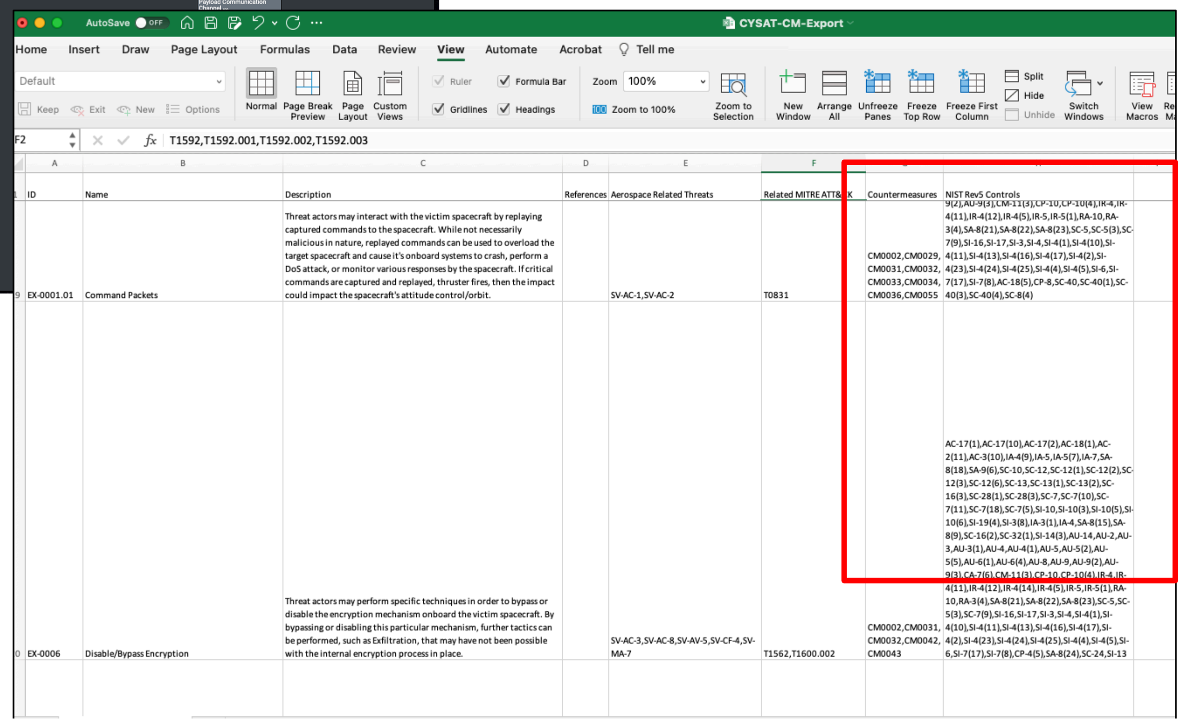
CM0001	Protect Sensitive Information	CM0029	TRANSEC
CM0002	COMSEC	CM0030	Crypto Key Management
CM0004	Development Environment Security	CM0031	Authentication
CM0005	Ground-based Countermeasures	CM0032	On-board Intrusion Detection & Prevention
CM0008	Security Testing Results	CM0033	Relay Protection
CM0010	Update Software	CM0034	Monitor Critical Telemetry Points
CM0011	Vulnerability Scanning	CM0035	Protect Authenticators
CM0012	Software Bill of Materials	CM0039	Least Privilege
CM0013	Dependency Confusion	CM0040	Shared Resource Leakage
CM0014	Secure boot	CM0042	Robust Fault Management
CM0015	Software Source Control	CM0043	Backdoor Commands
CM0016	CWE List	CM0044	Cyber-safe Mode
CM0017	Coding Standard	CM0047	Operating System Security
CM0018	Dynamic Analysis	CM0052	Insider Threat Protection
CM0019	Static Analysis	CM0053	Physical Security Controls
CM0020	Threat modeling	CM0054	Two-Person Rule
CM0021	Software Digital Signature	CM0055	Secure Command Mode(s)
CM0023	Configuration Management	CM0069	Process White Listing
CM0025	Supplier Review	CM0070	Alternate Communications Paths
CM0026	Original Component Manufacturer		

Combining the 4 Attack Chains

SPARTA Navigator – Extracting Countermeasures / NIST Controls



https://sparta.aerospace.org/navigator



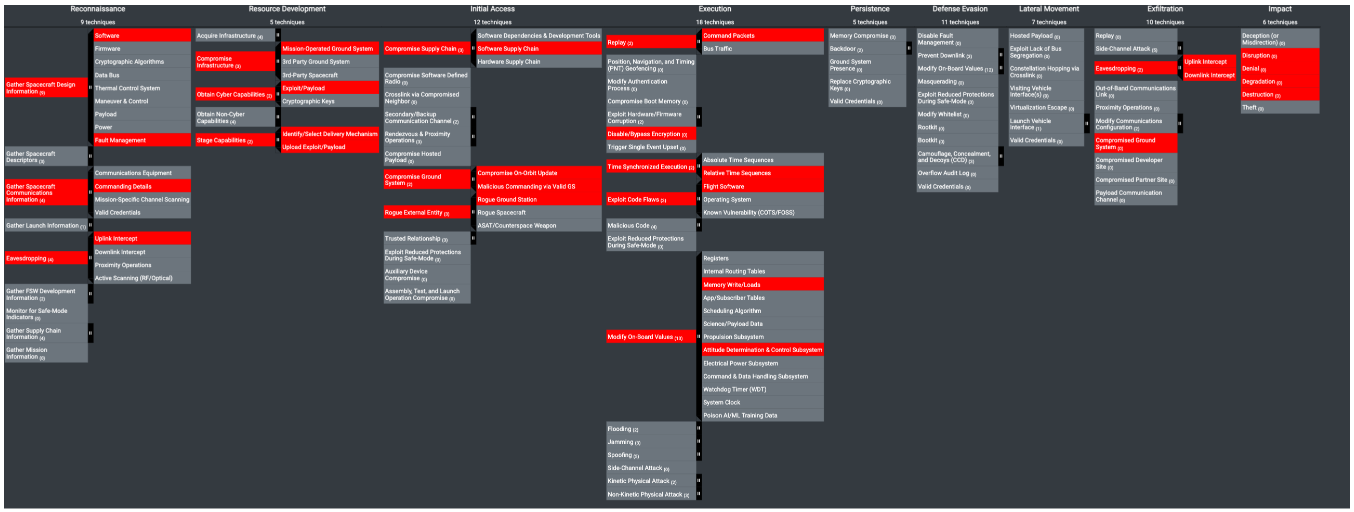
CM0001	CM0015	CM0028	CM0043
CM0002	CM0016	CM0029	CM0044
CM0003	CM0017	CM0030	CM0046
CM0004	CM0018	CM0031	CM0047
CM0005	CM0019	CM0032	CM0052
CM0007	CM0020	CM0033	CM0053
CM0008	CM0021	CM0034	CM0054
CM0009	CM0022	CM0035	CM0055
CM0010	CM0023	CM0036	CM0066
CM0011	CM0024	CM0038	CM0069
CM0012	CM0025	CM0039	CM0070
CM0013	CM0026	CM0040	CM0072
CM0014	CM0027	CM0042	CM0073

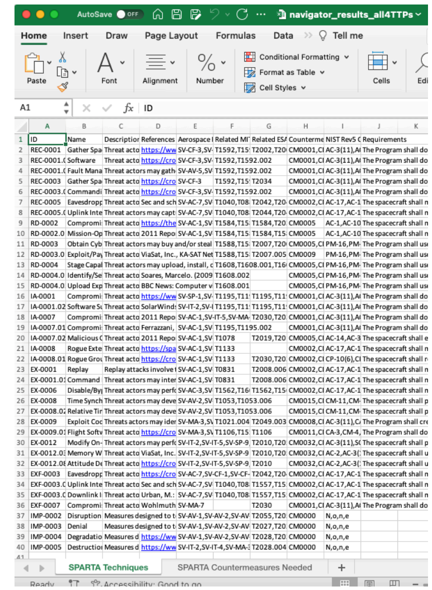
AC-3(11)	SA-10(7)	MA-3(1)	AC-3	SC-28(11)	CA-3(6)	SI-4(15)	CP-9	SI-10(5)
AC-4(23)	SA-11	MA-3(2)	AC-3(13)	SC-28(3)	CA-3(7)	SI-4(16)	CP-9(1)	SI-3(8)
AC-4(25)	SA-11(2)	MA-3(3)	AC-3(15)	SC-3	CA-7	SI-4(17)	CP-9(2)	PE-19
CM-12	SA-11(9)	MA-4	AC-3(4)	SC-38	CA-7(1)	SI-4(2)	CP-9(3)	PE-19(1)
CM-12(1)	SA-15	MA-4(1)	AC-4	SC-39	CA-7(6)	SI-4(20)	IA-11	PE-21
PM-11	SA-15(3)	MA-4(3)	AC-4(24)	SC-4	CA-8	SI-4(22)	IA-12	CP-11
PM-17	SA-15(7)	MA-4(6)	AC-4(26)	SC-45	CA-9	SI-4(23)	IA-12(1)	PM-16
SA-3(1)	SA-17	MA-4(7)	AC-4(31)	SC-45(1)	CM-10(1)	SI-4(24)	IA-12(2)	SA-15(8)
SA-3(2)	SA-2	MA-5(1)	AC-4(32)	SC-45(2)	CM-11	SI-4(25)	IA-12(3)	SC-32(1)
SA-4(12)	SA-22	MA-6	AC-6	SC-49	CM-11(2)	SI-4(4)	IA-12(4)	SA-10(3)
SA-5	SA-3	MA-7	AC-6(1)	SC-5	CM-11(3)	SI-4(5)	IA-12(5)	SA-10(4)
SA-9(7)	SA-4	MP-2	AC-6(10)	SC-5(1)	CM-14	SI-5	IA-12(6)	CA-8(3)
SI-21	SA-4(1)	MP-3	AC-6(2)	SC-5(2)	CM-2	SI-5(1)	IA-2	CM-4(1)
SI-23	SA-4(10)	MP-4	AC-6(3)	SC-5(3)	CM-2(2)	SI-6	IA-2(1)	SA-11(1)
SR-12	SA-4(2)	MP-5	AC-6(5)	SC-50	CM-2(3)	SI-7	IA-2(12)	SA-11(4)
SR-7	SA-4(3)	MP-5(4)	AC-6(8)	SC-51	CM-2(7)	SI-7(1)	IA-2(2)	SA-11(5)
AC-1	SA-4(5)	MP-6	AC-6(9)	SC-7	CM-3	SI-7(17)	IA-2(5)	SA-11(6)
AC-10	SA-4(7)	MP-6(3)	AC-7	SC-7(10)	CM-3(1)	SI-7(2)	IA-2(6)	SA-11(7)
AC-11	SA-4(9)	MP-7	AC-8	SC-7(11)	CM-3(2)	SI-7(5)	IA-2(8)	SA-11(8)
AC-11(1)	SA-8	PE-3(7)	AT-2(4)	SC-7(12)	CM-3(5)	SI-7(7)	IA-3	SA-15(5)
AC-12	SA-8(14)	PL-10	AT-2(5)	SC-7(13)	CM-3(7)	SI-7(8)	IA-3(1)	CM-7(4)
AC-12(1)	SA-8(15)	PL-11	AT-2(6)	SC-7(14)	CM-3(8)	SR-1	IA-4	RA-5(3)
AC-14	SA-8(18)	PL-8	AT-3	SC-7(18)	CM-4	SR-10	IA-4(9)	CM-8(7)
AC-16	SA-8(21)	PL-8(1)	AT-3(2)	SC-7(21)	CM-5(1)	SR-11	IA-5	SI-7(12)
AC-16(6)	SA-8(22)	PL-8(2)	AT-4	SC-7(25)	CM-5(5)	SR-11(1)	IA-5(1)	SI-7(15)
AC-17	SA-8(23)	PL-9	AU-10	SC-7(29)	CM-6	SR-11(2)	IA-5(13)	CM-5
AC-17(1)	SA-8(24)	PM-16(1)	AU-11	SC-7(3)	CM-6(1)	SR-11(3)	IA-5(14)	SI-7(9)

Combining the 4 Attack Chains

SPARTA Navigator – Extracting Countermeasures / NIST Controls

https://sparta.aerospace.org/navigator

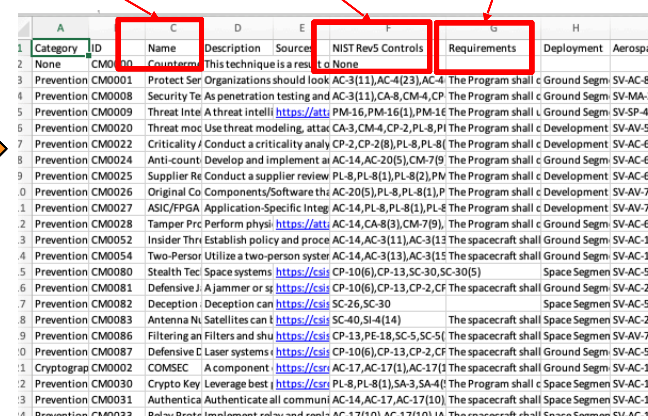




Countermeasure

NIST 800-53

Sample "Shalls"



Let's Apply This to a "Real" Event

CySat 2023 – OPS-SAT Hacking Demonstration

Full Analysis: **https://medium.com/the-aerospace-corporation/hacking-an-on-orbit-satellite-an-analysis-of-the-cysat-2023-demo-ae241e5b8ee5**

- Took place on April 26-27th in Paris, France
- Cybersecurity researchers demonstrated how they seized control of a European Space Agency (ESA) satellite.
  - *For those interested, a full retrospective of the previous 2022 event is available here.*
- Prior to CYSAT '23, researchers from the Thales Group worked in collaboration with ESA members to perform the structured experiment, which was unveiled at CYSAT '23.
  - *The experiment involved performing a cyber-attack against ESA's OPS-SAT, a nanosatellite that was launched in December 2019, and contains "an experimental computer ten times more powerful than any current ESA spacecraft."*

The CYSAT '23 cyber exercise builds upon similar events like the Hack-a-Sat program sponsored by the United States Air Force and United States Space Force that has occurred every year since 2020. Hack-a-Sat 4 in 2023 will leverage a 3U CubeSat called moonlighter in August 2023 at DefCon 31. The CubeSat's concept has a "cyber payload" that is independently recoverable via an alternate communication path which has been developed to train defensive cybersecurity researchers on a controlled, operational system.

The SPARTA team analyzed Thales Group's CYSAT '23 presentation material, as well as an article from The Record, to deconstruct the experiment and extract lessons learned and potential countermeasures to prevent such attacks. To accomplish this, SPARTA was leveraged to identify the tactics, techniques, and associated countermeasures associated with the experiment/attack.

OPS-SAT Mission

Overview

What is the OPS-SAT Space Lab?

esa

OPS-SAT-1 theme: Communication Protocols

OPS-SAT-2 theme: Optical and Quantum Communication

Images: ESA

OPS-SAT Space Lab is an **ESA service** to help accelerate innovation in ops related areas.

- It uses **powerful, reconfigurable** space elements that can be used for in-flight experimentation **not possible or desirable** on other missions
- The service provides access to these labs for **all** European industry and institutions, using a **fast, cost free, non bureaucratic process**
- ESA assumes the **risk and cost** of executing these in-flight experiments

Thales Cyber Security Experiment Context

esa

The OPS-SAT mission is a specially created environment that lends itself to performing in flight demonstrations of cyber security

- The ground infrastructure used for these exercises is completely isolated from that used by operational missions
- The satellite has been designed with the idea of an evil experimenter in mind. Therefore the bus is constantly monitoring the behaviour of the system and can shut it down if necessary. The emphasis is not on prevention but on recovery
- On-board operations are conducted in RAM only. Hence the system can be recovered by a power cycle of the experimental processor (SEPP)
- ESA was in control of system at all times, actively assisting the Thales team to perform the cyber security experiment.

The Attack – An Abridged Version

- Initial Access: researchers were given access to the payload to execute software which is the design of OPS-SAT. Users get access to the payload interface to run experiments.

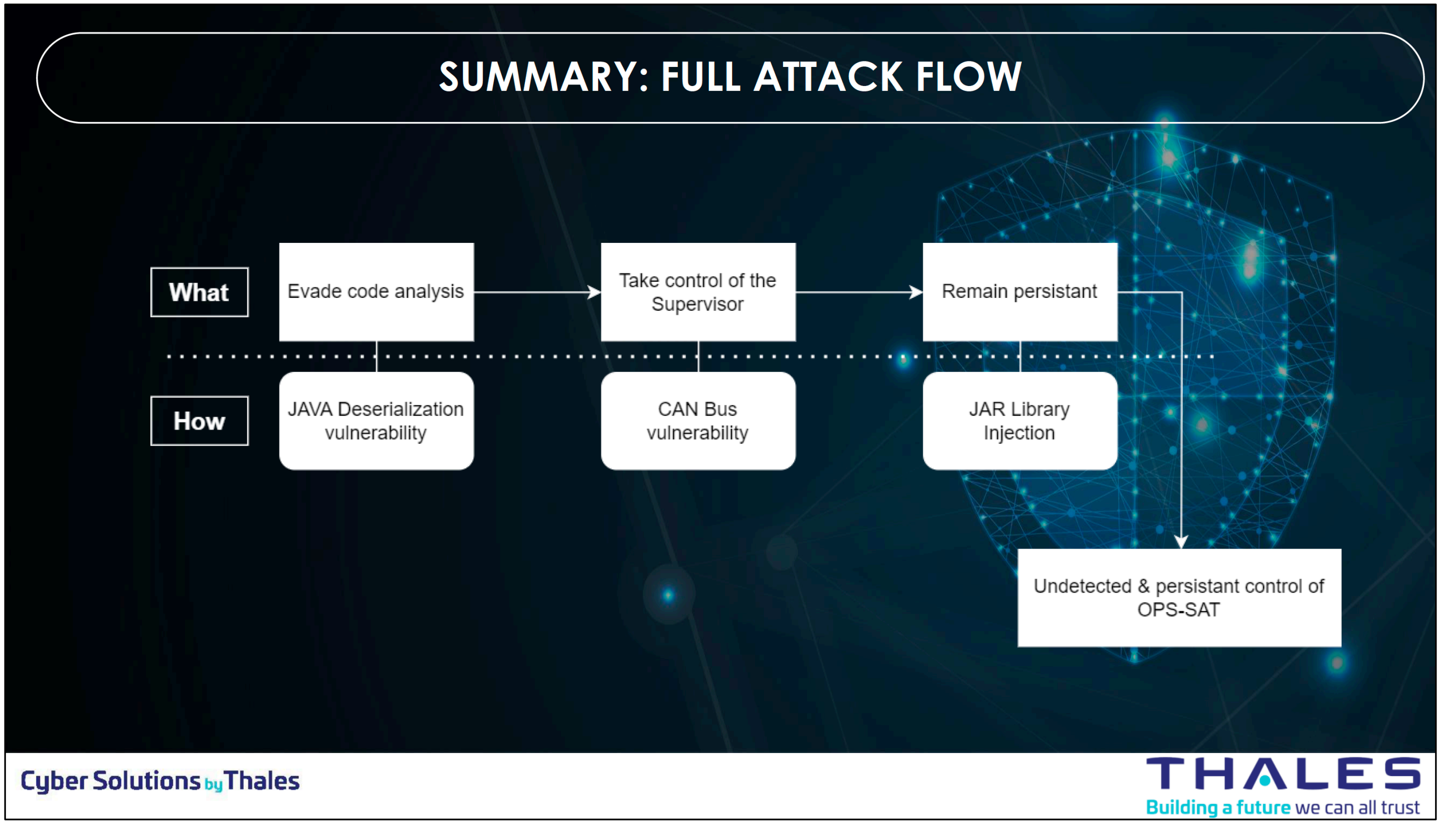
  As with virtually all cyber-attacks, significant reconnaissance and resource development are required to obtain initial access, which in this case was a simulated software supply chain attack via the hosted payload.

  - **Reconnaissance:** Gather Spacecraft Communications Information: Valid Credentials
  - **Resource Development:** Exploit/Payload
  - **Resource Development:** Identify/Select Delivery Mechanism
  - **Resource Development:** Upload Exploit/Payload
  - **Initial Access:** Compromise Hosted Payload
  - **Initial Access:** Compromise Supply Chain: Software Supply Chain

- The inject – simulated supply chain injection, the implanted a vulnerable piece of code they could later exploit.
- By injecting a vulnerability into the software, it provides defensive evasion in addition to code execution
  - *Exploited uploaded code with the deserialization vulnerability to execute arbitrary commands/code on the operating system. This technique was ultimately used to escalate to root privilege on the spacecraft.*
- CAN spacecraft bus not properly implementing any segmentation – payload could send message on bus
  - ***Execution:*** *Exploit Code Flaws: Operating System* & ***Lateral Movement:*** *Exploit Lack of Bus Segregation*
- **Persistence**: Backdoor: Software was used when injecting code into JAVA library
- Once persistence and escalation occurred, the researchers proceeded to attack the "mission" where they elected to affect the integrity of the imagery collected by the camera. (e.g., **Execution:** Modify On-Board Values: Science/Payload Data)

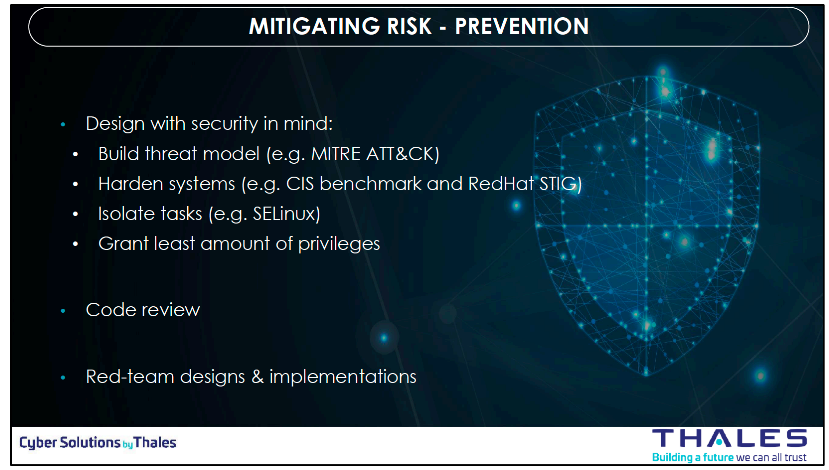
Full Analysis: ***https://medium.com/the-aerospace-corporation/hacking-an-on-orbit-satellite-an-analysis-of-the-cysat-2023-demo-ae241e5b8ee5***

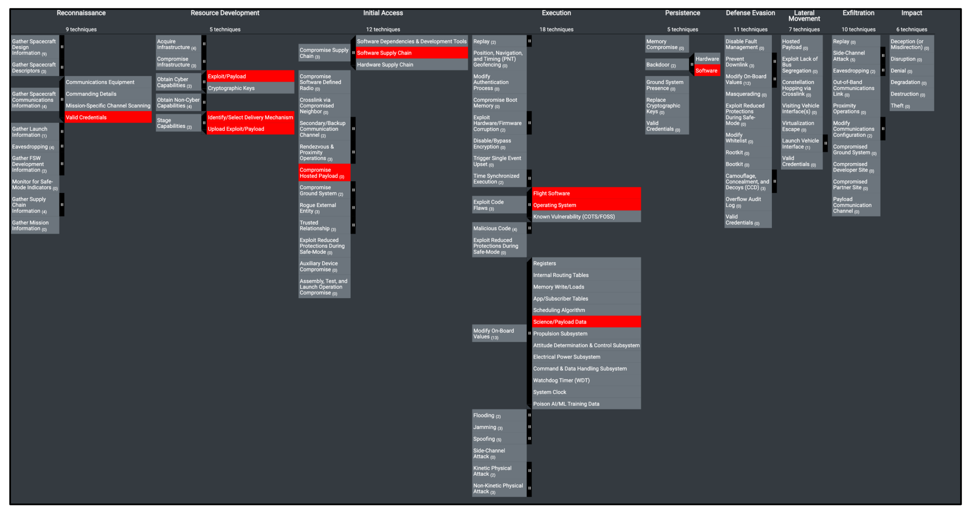
Full Attack Flow Summarized



So What? How Do We Prevent?

- The Thales Group presentation provided the high-level guidance, but SPARTA can be leveraged for detailed countermeasure guidance.
- Using the SPARTA Navigator to create the attack chain and then exporting the data into Excel enables countermeasure identification.
- Analysis was performed to confirm the associated countermeasure has application for specific TTPs.
  - *SPARTA helps by providing a menu of countermeasures sorted into defense-in-depth categories that can help with reducing the risk of TTPs.*
- Mapping the attack chain to SPARTA TTPs, the below graphic from [SPARTA navigator] is generated.





Countermeasures

On Ground – Preventative

- Eight countermeasures were identified
- Five of the eight would be countermeasures on the ground that would ideally prevent the vulnerable software from making its way onto the spacecraft.
- The remaining three countermeasures are on-board countermeasures that would help protect and/or detect the spacecraft from the TTPs executed during the experiment.

CM0016	CWE List	Create prioritized list of software weakness classes (e.g., Common Weakness Enumerations), based on system-specific considerations, to be used during static code analysis for prioritization of static analysis results.	RA-5,SA-11,SA-11(1),SA-15(7)	Enables a structured testing approach when doing static code analysis. For example, if testing were to look for CWE-502 and/or CWE-913 on the payload software before uploading to the spacecraft; initial access / execution of vulnerable code would not have been enabled.
CM0017	Coding Standard	Define acceptable coding standards to be used by the software developer. The mission should have automated means to evaluate adherence to coding standards. The coding standard should include the acceptable software development language types as well. The language should consider the security requirements, scalability of the application, the complexity of the application, development budget, development time limit, application security, available resources, etc. The coding standard and language choice must ensure proper security constructs are in place.	PL-8,PL-8(1),SA-11,SA-15,SA-3,SA-4(9),SA-8	Forcing developers to follow and prove they have strict security coding standards would likely prevent the deserialization vulnerability from being able to be implemented. For example, see coding standard rule SER03-J. Do not serialize unencrypted sensitive data.
CM0019	Static Analysis	Perform static source code analysis for all available source code looking for system-relevant weaknesses (see CM0016) using no less than two static code analysis tools.	RA-3,RA-5,SA-11,SA-11(1),SA-11(4),SA-15(7),SA-3,SA-8	Static analysis tools could be configured to detect the previously mentioned CWE-502 and/or CWE-913.
CM0018	Dynamic Analysis	Employ dynamic analysis (e.g., using simulation, penetration testing, fuzzing, etc.) to identify software/firmware weaknesses and vulnerabilities in developed and incorporated code (open source, commercial, or third-party developed code). Testing should occur (1) on potential system elements before acceptance; (2) as a realistic simulation of known adversary tactics, techniques, procedures (TTPs), and tools; and (3) throughout the lifecycle on physical and logical systems, elements, and processes. FLATSATs as well as digital twins can be used to perform the dynamic analysis depending on the TTPs being executed. Digital twins via instruction set simulation (i.e., emulation) can provide robust environment for dynamic analysis and TTP execution.	CA-8,CP-4(5),RA-3,RA-5(11),SA-11,SA-11(5),SA-11(8),SA-11(9),SA-3,SA-8,SC-2(2),SC-7(29),SI-3,SR-6(1),SR-6(1)	Before uploading the payload software, fuzzing / dynamic analysis may have been able to flush out the vulnerability prior to uploading the payload.
CM0020	Threat modeling	Use threat modeling, attack surface analysis, and vulnerability analysis to inform the current development process using analysis from similar systems, components, or services where applicable. Reduce attack surface where possible based on threats.	CA-3,CM-4,CP-2,PL-8,PL-8(1),RA-3,SA-11,SA-11(2),SA-11(6),SA-15(6),SA-15(8),SA-2,SA-3,SA-4(9),SA-8	If proper threat modeling would have been performed, then the spacecraft could have anticipated that an attacker may get code execution. This would have driven more of a defense in depth approach where you assume breach on the spacecraft. The threat model would assume the ground security on checking software prior to loading would by bypassed therefore, on-board intrusion detection, least privilege, segmentation, etc. would likely have had more focus.

Countermeasures

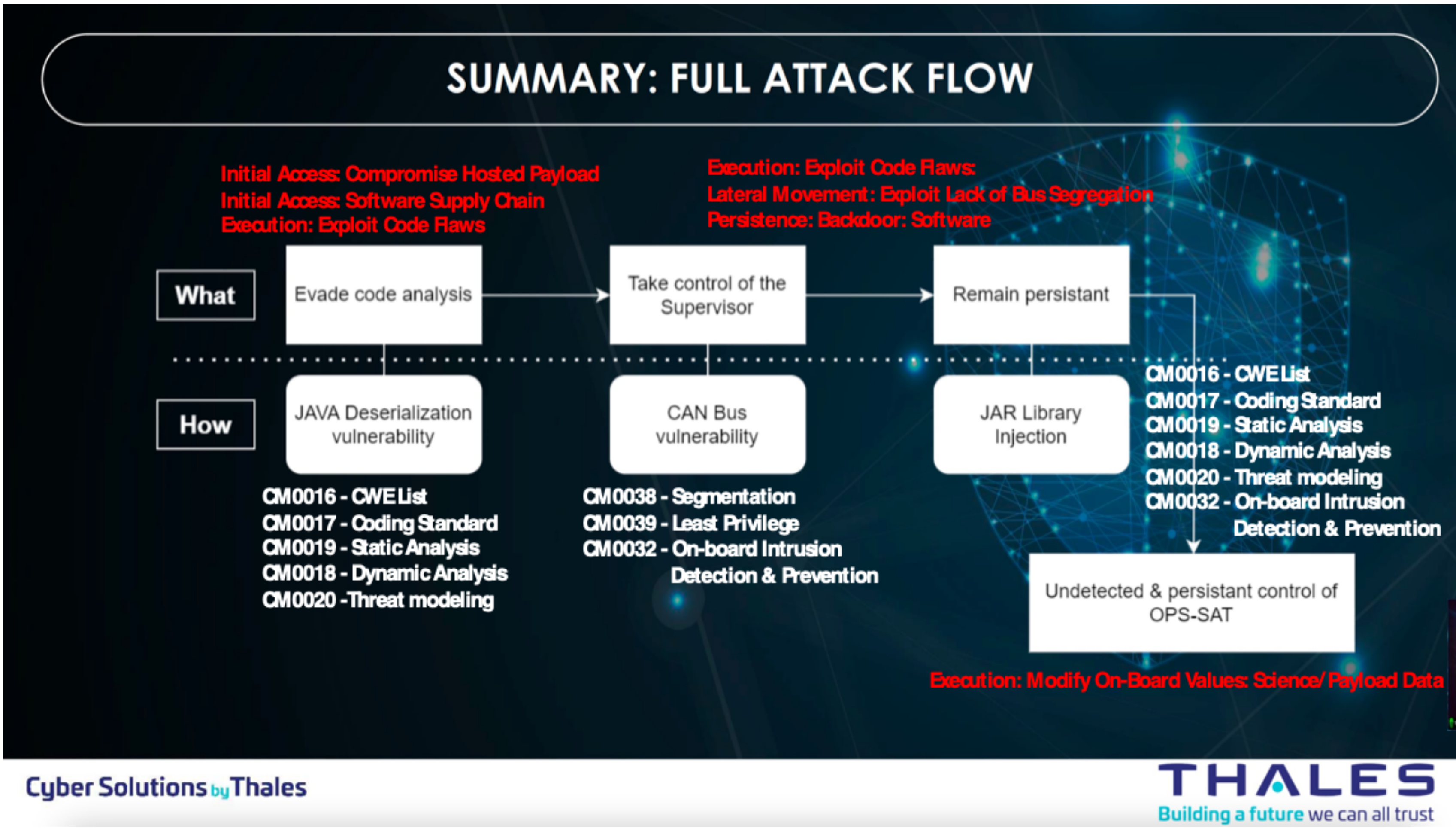
In Space

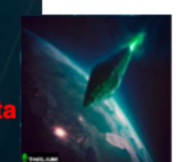
CM0032	On-board Intrusion Detection & Prevention	Utilize on-board intrusion detection/prevention system that monitors the mission critical components or systems and audit/logs actions. The IDS/IPS should have the capability to respond to threats (initial access, execution, persistence, evasion, exfiltration, etc.) and it should address signature-based attacks along with dynamic never-before seen attacks using machine learning/adaptive technologies. The IDS/IPS must integrate with traditional fault management to provide a wholistic approach to faults on-board the spacecraft. Spacecraft should select and execute safe countermeasures against cyber-attacks. These countermeasures are a ready supply of options to triage against the specific types of attack and mission priorities. Minimally, the response should ensure vehicle safety and continued operations. Ideally, the goal is to trap the threat, convince the threat that it is successful, and trace and track the attacker — with or without ground support. This would support successful attribution and evolving countermeasures to mitigate the threat in the future. "Safe countermeasures" are those that are compatible with the system's fault management system to avoid unintended effects or fratricide on the system.	AU-14,AU-2,AU-3,AU-3(1),AU-4,AU-4(1),AU-5,AU-5(2),AU-5(5),AU-6(1),AU-6(4),AU-8,AU-9,AU-9(2),AU-9(3),CA-7(6),CM-11(3),CP-10,CP-10(4),IR-4,IR-4(11),IR-4(12),IR-4(14),IR-4(5),IR-5,IR-5(1),PL-8,PL-8(1),RA-10,RA-3(4),SA-8(21),SA-8(22),SA-8(23),SC-16(2),SC-32(1),SC-5,SC-5(3),SC-7(10),SC-7(9),SI-10(6),SI-16,SI-17,SI-3,SI-3(8),SI-4,SI-4(1),SI-4(10),SI-4(11),SI-4(13),SI-4(16),SI-4(17),SI-4(2),SI-4(23),SI-4(24),SI-4(25),SI-4(4),SI-4(5),SI-6,SI-7(17),SI-7(8)	If an on-board security IDS were implemented there is high probability the escalation / lateral movement across the CAN bus would have been detected as the methods used are well known techniques.

CM0038	Segmentation	Identify the key system components or capabilities that require isolation through physical or logical means. Information should not be allowed to flow between partitioned applications unless explicitly permitted by security policy. Isolate mission critical functionality from non-mission critical functionality by means of an isolation boundary (implemented via partitions) that controls access to and protects the integrity of, the hardware, software, and firmware that provides that functionality. Enforce approved authorizations for controlling the flow of information within the spacecraft and between interconnected systems based on the defined security policy that information does not leave the spacecraft boundary unless it is encrypted. Implement boundary protections to separate bus, communications, and payload components supporting their respective functions.	AC-4,AC-4(14),AC-4(2),AC-4(24),AC-4(26),AC-4(31),AC-4(32),AC-4(6),AC-6,CA-3,CA-3(7),PL-8,PL-8(1),SA-3,SA-8,SA-8(13),SA-8(15),SA-8(18),SA-8(3),SA-8(4),SA-8(9),SC-16(3),SC-2(2),SC-3,SC-32(1),SC-39,SC-4,SC-49,SC-50,SC-6,SC-7(20),SC-7(21),SC-7(29),SC-7(5),SI-17	The CAN bus on the spacecraft does not properly segment the payload and the rest of the spacecraft. The lack of segmentation was exploited which enabled the execution of code running as *root* in this example. Without proper segmentation, escalation would have likely been stopped. This is a serious problem/concern on many spacecraft buses (e.g., CAN, 1553, etc.). Bus architectures need to implement more of a zero-trust model where the assume breach mentality is used to engineer the solutions.
CM0039	Least Privilege	Employ the principle of least privilege, allowing only authorized processes which are necessary to accomplish assigned tasks in accordance with system functions. Ideally maintain a separate execution domain for each executing process.	AC-2,AC-3(13),AC-3(15),AC-4(2),AC-6,CA-3(6),CM-7,CM-7(4),CM-7(8),PL-8,PL-8(1),SA-17(7),SA-3,SA-4(9),SA-8,SA-8(13),SA-8(14),SA-8(15),SA-8(3),SA-8(4),SA-8(9),SC-2(2),SC-32(1),SC-49,SC-50,SC-7(29)	The *'space shell root'* process/application runs as root and accepts input which enables escalation. If this application would have been running with limited privileges, then this specific escalation vector would have been stopped. Many spacecrafts run applications or the entire flight software with "root like" permissions and do not properly segment memory, file permissions, process isolation, etc. This lack of proper privilege management can enable many other attacks as shown by the TTPs tied to countermeasure CM0039 – Least Privilege.

Takeaways cont.

Attack Flow with SPARTA Overlays





Takeaways

Must Understand the Entire Attack Chains

- Countermeasures can be deployed that can disrupt/degrade steps of the attack chain
  - Reconnaissance or Resource Development is the precursor to almost all attacks
    - ~60% of the attacks from CyberInflight's space attack database
- For attacks focusing on space segment
  - *Initial access can be difficult and maybe the most difficult step historically but with supply chain, insider threat, compromised ground, etc. the likelihood of is increasing*
  - *As shown with the previously mentioned attack chains against spacecraft are not resilient against* Execution, Persistence, Defense Evasion, & Lateral Movement
    - Lack of process isolation/segmentation, overly permissive files/least privilege, running everything as root, lack of intrusion detection, logging, secure boot, software digital signatures, etc.
- CySat experiment, Hack-a-Sat events, past DefCon attack chains are contrived/controlled tests
  - *However, there are validity in the TTPs used and the vulnerabilities exploited*
  - *Validates many of the TTPs within SPARTA are accurate and the associated countermeasures in SPARTA can aide in TTP mitigation.*
  - *These experiments/tests also validates the importance of defense-in-depth*

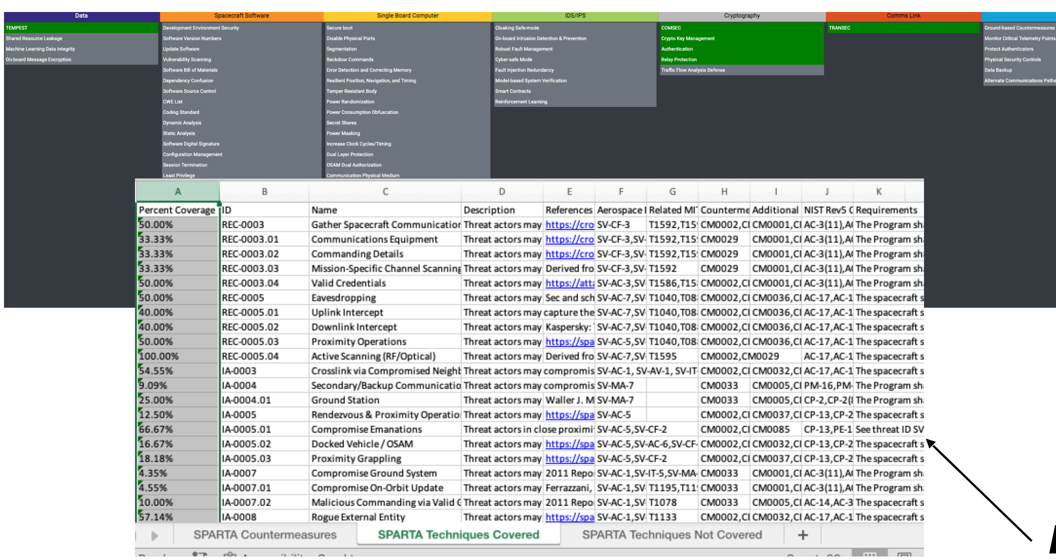
Since the ground controls often fail to catch the software injects or malicious commanding, it is recommended to implement on-board countermeasures like segmentation, least privilege, on-board IDS, etc. to prevent the TTPs used in the attack chains.

CM0009: Threat Intelligence Program
CM0002: COMSEC
CM0039: Least Privilege
CM0069: Process Whitelisting
CM0034: Monitor Critical Telemetry Points
CM0032: On-board Intrusion Detection & Prevention
CM0042: Robust Fault Management
CM0044: Cyber-safe Mode
CM0038: Segmentation
CM0029: TRANSEC

SPARTA Countermeasure Mapper / Defensive Gap Analyzer

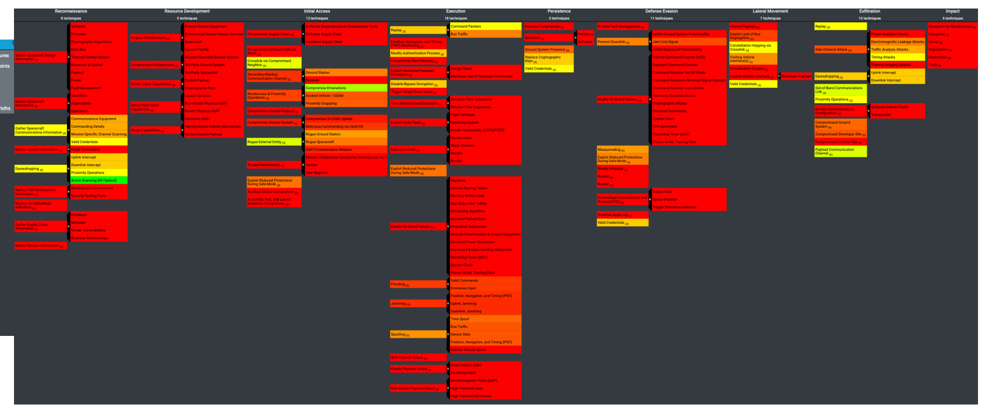
https://sparta.aerospace.org/countermeasures/mapper

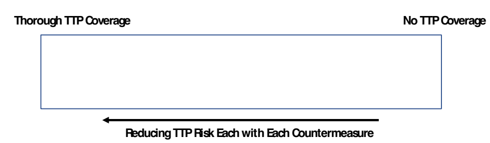
- Attack chains built in SPARTA's navigator can help identify countermeasures against the TTPs used in the attack
  - *Many users do not know TTPs, they only know the countermeasures they have implemented (or plan to)…*
- The SPARTA capability enables a graphical mechanism to select and deselect countermeasures from SPARTA's defense-in-depth view, *as the starting point*, to drive TTP mitigation & security planning
  - *It can export the data into Excel which provides tabs for coverage and gaps from a TTP perspective, including NIST controls*
- Below depicts the TTPs that have some mitigation when only applying COMSEC/TRANSEC/TEMPEST
  - ***Green/Yellow/Orange** indicates some level of coverage where **Red** indicates no coverage of the TTP*



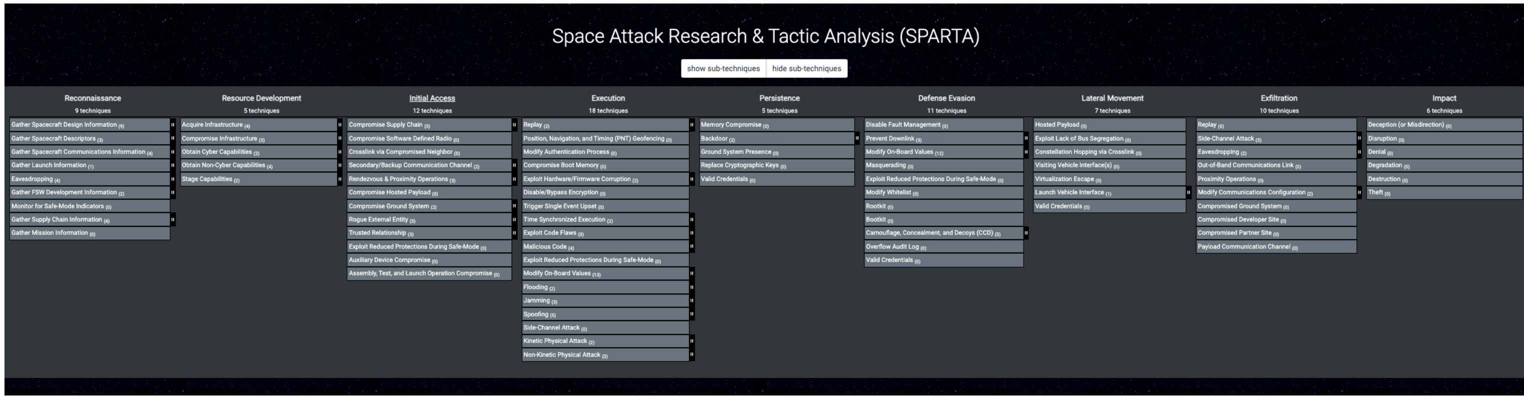
Percent Coverage	ID	Name	Description	References	Aerospace I	Related MI	Counterme	Additional	NIST Rev5 C	Requirements
50.00%	REC-0003	Gather Spacecraft Communicatior	Threat actors may	https://cro	SV-CF-3	T1592,T15	CM0002,CI	CM0001,CI	AC-3(11),A	The Program sh
33.33%	REC-0003.01	Communications Equipment	Threat actors may	https://cro	SV-CF-3,SV-	T1592,T15	CM0029	CM0001,CI	AC-3(11),A	The Program sh
33.33%	REC-0003.02	Commanding Details	Threat actors may	https://cro	SV-CF-3,SV-	T1592,T15	CM0029	CM0001,CI	AC-3(11),A	The Program sh
33.33%	REC-0003.03	Mission-Specific Channel Scanning	Threat actors may	Derived fro	SV-CF-3,SV-	T1592	CM0029	CM0001,CI	AC-3(11),A	The Program sh
50.00%	REC-0003.04	Valid Credentials	Threat actors may	https://att	SV-AC-3,SV	T1586,T15	CM0002,CI	CM0001,CI	AC-3(11),A	The Program sh
50.00%	REC-0005	Eavesdropping	Threat actors may	Sec and sch	SV-AC-7,SV	T1040,T08	CM0002,CI	CM0036,CI	AC-17,AC-1	The spacecraft s
40.00%	REC-0005.01	Uplink Intercept	Threat actors may	capture the	SV-AC-7,SV	T1040,T08	CM0002,CI	CM0036,CI	AC-17,AC-1	The spacecraft s
40.00%	REC-0005.02	Downlink Intercept	Threat actors may	Kaspersky:	SV-AC-7,SV	T1040,T08	CM0002,CI	CM0036,CI	AC-17,AC-1	The spacecraft s
50.00%	REC-0005.03	Proximity Operations	Threat actors may	https://spa	SV-AC-5,SV	T1040,T08	CM0002,CI	CM0036,CI	AC-17,AC-1	The spacecraft s
100.00%	REC-0005.04	Active Scanning (RF/Optical)	Threat actors may	Derived fro	SV-AC-7,SV	T1595	CM0002,CM0029		AC-17,AC-1	The spacecraft s
54.55%	IA-0003	Crosslink via Compromised Neighb	Threat actors may	compromis	SV-AC-1, SV-AV-1, SV-IT		CM0002,CI	CM0032,CI	AC-17,AC-1	The spacecraft s
9.09%	IA-0004	Secondary/Backup Communicatio	Threat actors may	compromis	SV-MA-7		CM0033	CM0005,CI	PM-16,PM-	The Program sh
25.00%	IA-0004.01	Ground Station	Threat actors may	Waller J. M	SV-MA-7		CM0033	CM0005,CI	CP-2,CP-2(	The Program sh
12.50%	IA-0005	Rendezvous & Proximity Operatio	Threat actors may	https://spa	SV-AC-5		CM0002,CI	CM0037,CI	CP-13,CP-2	The spacecraft s
66.67%	IA-0005.01	Compromise Emanations	Threat actors in	close proximi	SV-AC-5,SV-CF-2		CM0002,CI	CM0085	CP-13,PE-1	See threat ID SV
16.67%	IA-0005.02	Docked Vehicle / OSAM	Threat actors may	https://spa	SV-AC-5,SV-AC-6,SV-CF-		CM0002,CI	CM0032,CI	CP-13,CP-2	The spacecraft s
18.18%	IA-0005.03	Proximity Grappling	Threat actors may	https://spa	SV-AC-5,SV-CF-2		CM0002,CI	CM0037,CI	CP-13,CP-2	The spacecraft s
4.35%	IA-0007	Compromise Ground System	Threat actors may	2011 Repo	SV-AC-1,SV-IT-5,SV-MA-		CM0033	CM0001,CI	AC-3(11),A	The Program sh
4.55%	IA-0007.01	Compromise On-Orbit Update	Threat actors may	Ferrazzani,	SV-AC-1,SV	T1195,T11	CM0033	CM0001,CI	AC-3(11),A	The Program sh
10.00%	IA-0007.02	Malicious Commanding via Valid C	Threat actors may	2011 Repo	SV-AC-1,SV	T1078	CM0033	CM0005,CI	AC-14,AC-3	The spacecraft s
57.14%	IA-0008	Rogue External Entity	Threat actors may	https://spa	SV-AC-1,SV	T1133	CM0002,CI	CM0032,CI	AC-17,AC-1	The spacecraft s

Excel Output





https://sparta.aerospace.org



SPARTA
SPACE ATTACK RESEARCH & TACTIC ANALYSIS

Sample Media Links:

- https://cyberscoop.com/space-satellite-cybersecurity-sparta/
- https://www.darkreading.com/ics-ot/space-race-defenses-satellite-cyberattacks
- https://thecyberwire.com/podcasts/daily-podcast/1715/notes & https://thecyberwire.com/newsletters/signals-and-space/6/21

Overview Briefings:

- Hacking Spacecraft using Space Attack Research & Tactic Analysis (April 2023)
- In-depth Overview - Space Attack Research & Tactic Analysis (November 2022)

Key SPARTA Links:

- Getting Started with SPARTA: https://sparta.aerospace.org/resources/getting-started | https://sparta.aerospace.org/resources/
- Understanding Space-Cyber TTPs with the SPARTA Matrix: https://aerospace.org/article/understanding-space-cyber-threats-sparta-matrix
- Leveraging the SPARTA Matrix: https://aerospace.org/article/leveraging-sparta-matrix
- Use Case w/ PCspooF:
  - https://aerospacecorp.medium.com/sparta-cyber-security-for-space-missions-4876f789e41c
  - https://medium.com/the-aerospace-corporation/a-look-into-sparta-countermeasures-358e2fcd43ed
- FAQ: https://sparta.aerospace.org/resources/faq
- Matrix: https://sparta.aerospace.org
- Navigator: https://sparta.aerospace.org/navigator | Countermeasure Mapper: https://sparta.aerospace.org/countermeasures/mapper
- Related Work: https://sparta.aerospace.org/related-work/did-space with ties into TOR 2021-01333 REV A

Other Aerospace Papers and Resources

Many Were Input into SPARTA

- Indiana University Space Cybersecurity Digital Badge - https://kelley.iu.edu/programs/executive-education/programs-for-individuals/digital-badges/cybersecurity-foundations.html

- DefCON Presentations:
  - DEF CON 2020: Exploiting Spacecraft
  - DEF CON 2021: Unboxing the Spacecraft Software BlackBox Hunting for Vulnerabilities
  - DEF CON 2022: Hunting for Spacecraft Zero Days using Digital Twins

- Papers/Articles:
  - 2019: Defending Spacecraft in the Cyber Domain
  - 2020: Establishing Space Cybersecurity Policy, Standards, & Risk Management Practices
  - 2021: Cybersecurity Protections for Spacecraft: A Threat Based Approach
  - 2021: The Value of Space
  - 2022: Protecting Space Systems from Cyber Attack

- July 2022 Congressional Testimony:
  - Video: https://science.house.gov/hearings?ID=996438A6-A93E-4469-8618-C1B59BC5A964
  - Written Testimony: https://republicans-science.house.gov/_cache/files/2/9/29fff6d3-0176-48bd-9c04-00390b826aed/A8F54300A11D55BEA5AF2CE305C015BA.2022-07-28-bailey-testimony.pdf

Theoretical Attack Chain - PCspooF

Example Attack Chains from the Past

2022 TTE Vulnerability - PCspooF

- Research paper by Andrew Loveless, Linh Thi Xuan Phan, Ronald Dreslinski and Baris Kasikci describing an attack dubbed PCspooF. The academic paper expertly articulates a [vulnerability in and exploit of Time-Triggered Ethernet (TTE)](), which is used as a bus service for a variety of spacecraft including NASA's Orion capsule, NASA's Lunar Gateway space station, and ESA's Ariane 6 launcher — among others.

PCSPOOF: Compromising the Safety of Time-Triggered Ethernet

Andrew Loveless[*‡] Linh Thi Xuan Phan[†] Ronald Dreslinski[*] Baris Kasikci[*]
[*]University of Michigan [†]University of Pennsylvania [‡]NASA Johnson Space Center
[*]{loveless, rdreslin, barisk}@umich.edu [†]linhphan@seas.upenn.edu

***Abstract*—Designers are increasingly using mixed-criticality networks in embedded systems to reduce size, weight, power, and cost. Perhaps the most successful of these technologies is Time-Triggered Ethernet (TTE), which lets critical time-triggered (TT) traffic and non-critical best-effort (BE) traffic share the same switches and cabling. A key aspect of TTE is that the TT part of the system is *isolated* from the BE part, and thus BE devices have no way to disrupt the operation of the TTE devices. This isolation allows designers to: (1) use untrusted, but low cost, BE hardware, (2) lower BE security requirements, and (3) ignore BE devices during safety reviews and certification procedures.**

We present PCSPOOF, the first attack to break TTE's isolation guarantees. PCSPOOF is based on two key observations. First, it is possible for a BE device to infer private information about the TT part of the network that can be used to craft malicious synchronization messages. Second, by injecting electrical noise into a TTE switch over an Ethernet cable, a BE device can trick the switch into sending these malicious synchronization messages to other TTE devices. Our evaluation shows that successful attacks are possible in seconds, and that each successful attack can cause TTE devices to lose synchronization for up to a second and drop tens of TT messages — both of which can result in the failure of critical systems like aircraft or automobiles. We also show that, in a simulated spaceflight mission, PCSPOOF causes uncontrolled maneuvers that threaten safety and mission success. We disclosed PCSPOOF to aerospace companies using TTE, and several are implementing mitigations from this paper.

***Index Terms*—Time-Triggered Ethernet, packet-in-packet attacks, electromagnetic interference, embedded systems**

I. INTRODUCTION

Increasingly, embedded systems are using *mixed-criticality* network technologies that allow traffic with different timing and fault tolerance requirements to coexist in the same physical network [1]–[4]. These technologies let designers reduce size, weight, power, and cost by sharing the same network between critical and non-critical parts of the system. For example, aircraft can share one network between vehicle control systems and passenger Wi-Fi and entertainment systems [5], [6]; spacecraft can share one network between life support systems and onboard experiments [7], [8]; and manufacturing plants can share one network between robot control systems and data collection systems [9].

One of the most successful mixed-criticality network technologies is *Time-Triggered Ethernet (TTE)* [2], Today, TTE serves as the network backbone for several spacecraft, including NASA's Orion capsule [10], NASA's Lunar Gateway space station [7], and ESA's Ariane 6 launcher [11]. TTE is also widely used in aircraft [12]–[14], energy generation systems [15], and industrial control systems [16], [17], and is a leading contender to replace CAN bus and FlexRay as the standard network technology in future automobiles [18], [19].

TTE has several properties that make it attractive for safety and mission-critical applications. Most notably, TTE follows a *time-triggered (TT)* paradigm, in which devices are tightly synchronized, and they send messages and execute software according to a predetermined schedule. This TT approach reduces message latencies to hundreds of microseconds and jitter to near-zero [20], [21], making TTE appropriate for even the tightest control loops. TTE also provides fault tolerance by replicating the whole network to form multiple *planes*, and by forwarding messages over all planes simultaneously [22].

In addition, TTE enables mixed-criticality architectures by being 100% compatible with standard Ethernet [23]. This means that *non-critical* systems, which typically use standard Ethernet hardware to lower costs [24], can send messages over the same cabling as the critical TTE devices. Unlike TT traffic, standard Ethernet traffic is forwarded on a *best-effort (BE)* basis, filling in space *around* the TT traffic [23]. Also, standard Ethernet traffic typically only travels over a single network plane, so does not have any fault tolerance guarantees [7].

A key aspect of TTE's mixed-criticality design is that the TT part of the system is *isolated* from the BE part. In other words, no matter how the BE devices behave, they should not be able to disrupt synchronization between TTE devices, or the timely or successful delivery of TT traffic [25]. This isolation is commonly used as justification for several cost-cutting measures, including: (1) procuring BE devices from relatively untrusted (but low cost) suppliers [26], [27]; (2) relaxing security requirements for BE devices [28]; and (3) reducing the scope of analysis and certification of a system to focus solely on the TTE devices [29]. For example, on NASA spacecraft, onboard experiments are often provided by university research groups, are operated by the university students with minimal NASA involvement, and are not considered in safety reviews or the certification process of the overall vehicle [30], [31].

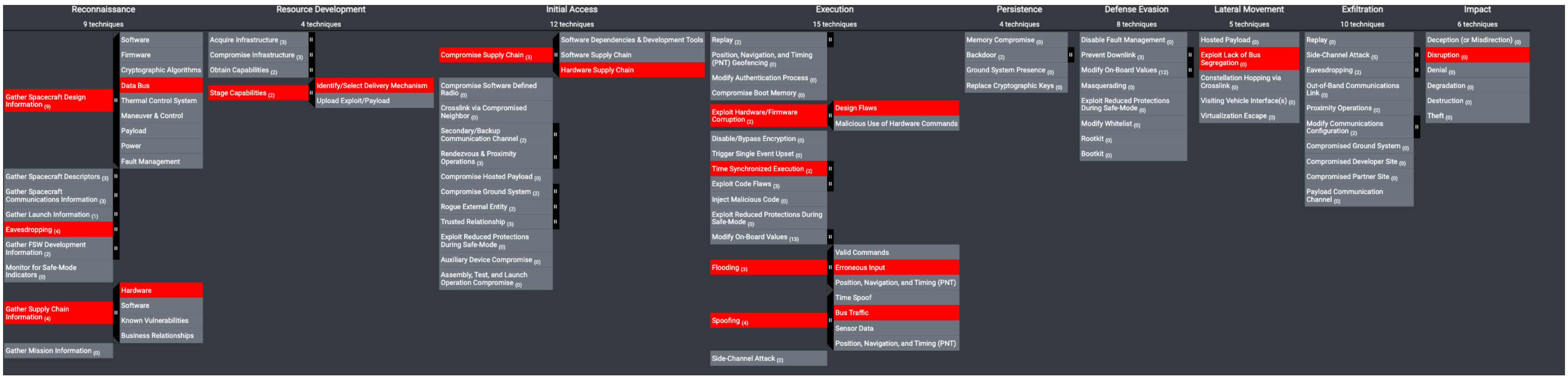
In this paper, we present PCSPOOF, a new attack that breaks TTE's isolation guarantees for the first time — allowing a single malicious BE device on a single plane to disrupt synchronization and communication between TTE devices on all planes. PCSPOOF is based on two key observations:

First, it is possible for a malicious BE device to *infer* private information about the TTE network that is needed to construct valid TTE synchronization messages, called *protocol control*

SPARTA
SPACE ATTACK RESEARCH & TACTIC ANALYSIS

Example Attack Chains from the Past

PCspooF Potential Attack Chain



Introducing SPARTA using PCSpooF: Cyber Security for Space Missions - https://medium.com/the-aerospace-corporation/sparta-cyber-security-for-space-missions-4876f789e41c
A Look into SPARTA Countermeasures - https://medium.com/the-aerospace-corporation/a-look-into-sparta-countermeasures-358e2fcd43ed

PCspooF Countermeasure Samples

Quick Way to Identify Potential Mitigations

Introducing SPARTA using PCSpooF: Cyber Security for Space Missions - https://medium.com/the-aerospace-corporation/sparta-cyber-security-for-space-missions-4876f789e41c

A Look into SPARTA Countermeasures - https://medium.com/the-aerospace-corporation/a-look-into-sparta-countermeasures-358e2fcd43ed

Original Component Manufacturer

Components that cannot be procured from the original component manufacturer or their authorized franchised distribution network should be approved by the supply cha… prevent and detect counterfeit and fraudulent parts and materials.

Best Segment for Countermeasure Deployment

- Development Environment

Informational References

- AC-20(5) - Use of External Systems | Portable Storage Devices — Prohibit…
- PM-30 - Supply Chain Risk Management Strategy
- PM-30(1) - Supply Chain Risk Management Strategy | Suppliers of Critica… essential Items
- RA-3(1) - Risk Assessment | Supply Chain Risk Assessment
- SR-1 - Policy and Procedures
- SR-11 - Component Authenticity
- SR-2 - Supply Chain…
- SR-2(1) - Supply Ch…
- SR-3 - Supply Chain…
- SR-3(1) - Supply Ch…

Techniques

ID	Name
IA-0001	Compromise Supply Chain
.03	Hardware Supply Chain
IA-0002	Comprom…

Segmentation

Identify the key system components or capabilities that require isolation through physical or logical means. Information should not be allowed to flow between partitioned applications unless explicitly permitted by security policy. Isolate mission critical functionality from non-mission critical functionality by means of an isolation boundary (implemented via partitions) that controls access to and protects the integrity of, the hardware, software, and firmware that provides that functionality. Enforce approved authorizations for controlling the flow of information within the spacecraft and between interconnected systems based on the defined security policy that information does not leave the spacecraft boundary unless it is encrypted. Implement boundary protections to separate bus, communications, and payload components supporting their respective functions.

ID: CM0038
Created: 2022/10/19
Last Modified: 2022/10/19

Sources

- https://attack.mitre.org/mitigations/M1030/

- …16(3) - Transmission of Security and Privacy Attributes | Cryptographic Binding
- …2(2) - Separation of System and User Functionality | Disassociability
- …3 - Security Function Isolation
- …32(1) - System Partitioning | Separate Physical Domains for Privileged Functions
- …39 - Process Isolation
- …4 - Information in Shared System Resources
- …49 - Hardware-enforced Separation and Policy Enforcement
- …50 - Software-enforced Separation and Policy Enforcement
- …6 - Resource Availability
- …7(21) - Boundary Protection | Isolation of System Components
- …7(29) - Boundary Protection | Separate Subnets to Isolate Functions

Dynamic Analysis

Employ dynamic analysis (e.g., using simulation, penetration te… commercial, or third-party developed code). Testing should occ… procedures (TTPs), and tools; and (3) throughout the lifecycle…

Best Segment for Countermeas…

- Ground Segment and Development Environment

Informational References

- CA-8 - Penetration Testing
- CP-4(5) - Contingency Plan Testing | Self-challenge
- RA-5(11) - Vulnerability Monitoring and Scanning | Public…
- SA-11(5) - Developer Testing and Evaluation | Penetration…
- SA-11(8) - Developer Testing and Evaluation | Dynamic C…
- SA-11(9) - Developer Testing and Evaluation | Interactive…
- SC-2(2) - Separation of System and User Functionality | D…
- SC-7(29) - Boundary Protection | Separate Subnets to Isol…
- SR-6(1) - Supplier Assessments and Reviews | Testing an…

Techniques Addressed by Cour…

ID	Name	Description
IA-0001	Compromise Supply Chain	Threat actors may ma…
.02	Software Supply Chain	Threat actors may ma… manipulation of the u…
.03	Hardware Supply Chain	Threat actors may ma… when they modify the…
IA-0007	Compromise Ground Station	Threat actors may initially compromise the ground station in order to access the target SV. Once compromised, the threat actor can perform a multitude of initial access techniques, including replay, comp… encryption keys, and compromising authentication schemes.
.01	Compromise On-Orbit…	Threat actors may manipulate and modify on-orbit updates before they are sent to the target SV. This attack can be done in a number of ways, including manipulation of source code, manipulating environ…

On-board Intrusion Detection & Prevention

Utilize on-board intrusion detection/prevention system that monitors the mission critical components or systems and audit/logs actions. The IDS/IPS should have the capability to respond to threats and it should address signature-based attacks along with dynamic never-before seen attacks using machine learning/adaptive technologies. The IDS/IPS must integrate with traditional fault management to provide a wholistic approach to faults on-board the spacecraft. Spacecraft should select and execute safe countermeasures against cyber-attacks. These countermeasures are a ready supply of options to triage against the specific types of attack and mission priorities. Minimally, the response should ensure vehicle safety and continued operations. Ideally, the goal is to trap the threat, convince the threat that it is successful, and trace and track the attacker — with or without ground support. This would support successful attribution and evolving countermeasures to mitigate the threat in the future. "Safe countermeasures" are those that are compatible with the system's fault management system to avoid unintended effects or fratricide on the system.

ID: CM0032
Created: 2022/1…
Last Modified: 2…

Sources

- https://attack.mitre.org/mitigations/M1031/

Best Segment for Countermeasure Deployment

- Space Segment

Informational References

- AU-14 - Session Audit
- 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-5(2) - Response to Audit Logging Process Failures | Real-time Alerts
- AU-5(5) - Response to Audit Logging Process Failures | Alternate Audit Logging Capability
- 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-9(3) - Protection of Audit Information | Cryptographic Protection
- CA-7(6) - Continuous Monitoring | Automation Support for Monitoring
- CM-11(3) - User-installed Software | Automated Enforcement and Monitoring
- CP-10 - System Recovery and Reconstitution
- CP-10(4) - System Recovery and Reconstitution | Restore Within Time Period
- 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

Techniques Addressed by Countermeasure

Authentication

Authenticate all communication sessions (crosslink and ground stations) for all commands before establishing remote connections using bidirectional authentication that is cryptographically based. Adding authentication on the spacecraft bus and communications on-board the spacecraft is also recommended.

ID: CM0031
Created: 2022/10/19
Last Modified: 2022/10/19

Best Segment for Countermeasure Deployment

- Space Segment

Informational References

- AC-17(10) - Remote Access | Authenticate Remote Commands
- AC-17(2) - Remote Access | Protection of Confidentiality and Integrity Using Encryption
- AC-18(1) - Wireless Access | Authentication and Encryption
- IA-3(1) - Device Identification and Authentication | Cryptographic Bidirectional Authentication
- IA-4 - Identifier Management
- IA-4(9) - Identifier Management | Attribute Maintenance and Protection
- IA-7 - Cryptographic Module Authentication
- SA-8(15) - Security and Privacy Engineering Principles | Predicate Permission
- SA-8(9) - Security and Privacy Engineering Principles | Trusted Components
- SC-16(2) - Transmission of Security and Privacy Attributes | Anti-spoofing Mechanisms
- SC-32(1) - System Partitioning | Separate Physical Domains for Privileged Functions
- SC-7(11) - Boundary Protection | Restrict Incoming Communications Traffic
- SI-14(3) - Non-persistence | Non-persistent Connectivity

Techniques Addressed by Countermeasure

ID	Name	Description
IA-0003	Crosslink via Compromised Neighbor	Threat actors may compromise a victim SV via the crosslink communications of a neighboring SV that has been compromised. SVs in close proximity are able to send commands back and forth. Threat a… compromise other SVs once they have access to another that is nearby.
EX-0001	Replay	Replay attacks involve threat actors recording previously data streams and then resending them at a later time. This attack can be used to fingerprint systems, gain elevated privileges, or even cause a den…
.01	Command Packets	Threat actors may interact with the victim SV by replaying captured commands to the SV. While not necessarily malicious in nature, replayed commands can be used to overload the target SV and cause it… attack, or monitor various responses by the SV. If critical commands are captured and replayed, thruster fires, then the impact could impact the SV's attitude control/orbit.
EX-0006	Disable/Bypass…	Threat actors may perform specific techniques in order to bypass or disable the encryption mechanism onboard the victim SV. By bypassing or disabling this particular mechanism, further tactics can be p…

…within visual contact or close…
…to deploy malware to latera…
…has the ability to connect vi…
…pecific command set. The co… …to command hosted paylo…