

SATBLEED: Security of Commoditized Communication Modules in Satellites

Ulysse Planta*, Julian Rederlechner*, Martin Strohmeier†, Mathias Fischer‡ and Ali Abbasi*

* CISPA - Helmholtz Center for Information Security, firstname.lastname@cispa.de

† Cyber-Defence Campus, armasuisse Science + Technology, firstname.lastname@armasuisse.ch

‡ University of Hamburg, firstname.lastname@uni-hamburg.de

Abstract—Substantial reduction in launch and manufacturing costs has resulted in the accelerated deployment of small satellite missions, with commercial off-the-shelf (COTS) components becoming the prevailing standard for specific subsystems. However, this modular architecture introduces critical security risks, most notably in the Communication Subsystem (COM), which is continuously exposed by design and implicitly trusted as the entry point for command and control. We construct a tailored threat taxonomy for attacks targeting the COM subsystem and analyze representative COM systems from various vendors. Our findings uncover severe vulnerabilities across firmware, protocols, and architectural designs. This work presents the first in-depth security evaluation of widely deployed COTS COM modules employed in small satellites, identifying vulnerabilities affecting dozens of missions. To assess the real-world impact, we correlate our discoveries with open-source telemetry data, inferring at least 28 vulnerable missions in orbit that are susceptible to hostile takeover. Our work reveals that satellite COM subsystems form an attractive and dangerously neglected attack surface, necessitating urgent attention from the community.

1. Introduction

The New Space era, starting in the 2000s [1], transformed small satellite missions through dramatically reduced launch and manufacturing costs. Both civil and military operators increasingly adopted commercial off-the-shelf (COTS) components and rideshare launches as bespoke satellite designs and missions became economically uncompetitive in a much-accelerated innovation environment. This commodification of, especially, Low Earth Orbit (LEO) platforms has driven rapid deployment, affordable in-orbit replacements of existing satellite platforms, and a shift away from comprehensive fault handling. The result is a modular architecture in which subsystems for attitude control, energy management, communications, and data processing are purchased as flight-proven building blocks and integrated by mission-specific software.

Unfortunately, satellite systems are increasingly targeted in cyber attacks and operations [2], [3], [4], [5], and New Space systems are especially vulnerable [1]. While modularity accelerates development, it also relocates the security challenge to the onboard Command and Data Handling Sys-

tem (CDHS) software on a satellite platform. Each CDHS must be customized to coordinate heterogeneous subsystems, and the absence of standardization creates different vulnerabilities across many missions [6], [7]. In contrast, widely reused subsystems offer adversaries a scalable attack surface.

Among these widely-reused subsystems, the Communication Subsystem (COM) is particularly critical and at the same time the most exposed. First, the COM necessarily remains continuously accessible for mission control and cannot be shut off like other subsystems. Second, it provides the only uplink and downlink interface for commanding the satellite, effectively acting as the perimeter guard to an internally fully trusted satellite system. Third, as we show, this guard function is often poorly implemented, and the COM facilitates attacks on internal components. In combination, adversaries can therefore acquire popular COTS COM hardware, reverse-engineer its firmware and protocols, and exploit the found weaknesses on deployed commercial constellations used by a wide range of civilian and military actors [8]. In addition, vulnerabilities are difficult to patch in-orbit due to data rate and safety constraints.

Even worse, COM vulnerabilities in satellites are not simply limited to ground-based attacks. Any other satellite equipped with a compatible transceiver may launch lateral attacks on nearby spacecraft. This satellite-to-satellite threat has received little academic attention despite its postulated feasibility [9], [10], [11].

While significant academic research has been conducted in recent years on the security of CDHS and satellite communication itself (e.g., [6], [10], [12], [13]), the most critical onboard communications modules at the very heart of every satellite mission have not undergone direct security evaluation.

In this paper, we fill that gap. We develop a threat taxonomy tailored to spaceborne communication subsystems and select representative examples from common vendors used in over one hundred missions to perform an in-depth security assessment, mapping the findings to our taxonomy. Among other weaknesses, we discover a critical flaw that we call **SATBLEED** in a widely-used COM subsystem. This vulnerability allows an unauthenticated attacker to extract cryptographic material from ground stations by simply imitating a satellite.

We correlate our findings with open-source telemetry

from the SatNOGS platform [14], a global network of over 400 ground stations, and vendor disclosures to quantify real-world impact on commercial and military missions. Using this approach, we infer that at least 28 satellites are likely vulnerable to complete, persistent takeover, with additional ones exposed to other threats discussed in this work.

Through these insights, we highlight critical vulnerabilities in small-satellite COM subsystems and derive architectural recommendations to improve resilience in the New Space era. Overall, we aim to strengthen the security of the small satellite ecosystem through the following contributions:

- We identify a major research gap in the security of COM subsystems in small satellites, an area largely overlooked in academia and industry.
- We introduce a threat taxonomy tailored to Telemetry Tracking and Control (TT&C) COM subsystems.
- We select representative COM modules from widely used vendors and conduct the first in-depth security evaluation of such subsystems in small satellites, revealing previously undocumented vulnerabilities.
- We are the first to link discovered satellite vulnerabilities to real-world deployments using open-source satellite registry data to identify affected missions.
- We introduce a physical-layer fuzzer, *SyncWordFuzzer*, capable of triggering bugs in sync-word detection in COMs¹.

2. Background

Small satellite missions involve spacecraft with a mass of up to 500 kg. As of April 2025, 12,149 active satellites are in orbit, and 79% of technology development satellites launched before 2023 weigh under 300 kg [15]. CubeSats alone make up 2,730 satellites [16]. Component interoperability across small satellite form factors and the widespread reuse of identical modules amplify the attack surface shared by many missions.

2.1. Satellite Communication

Satellites traditionally consist of two main parts: the *platform* (or satellite bus), which houses core systems such as power, attitude control, and data handling; and the *payload*, which performs the mission-specific functions. Depending on design, platform and payload may be entirely isolated and connected only by simple control voltages or so tightly integrated that the physical boundary is indistinguishable.

Operations rely on a ground segment comprising a Mission Control System (MCS) and one or more Ground Stations (GSs). The MCS schedules and dispatches Telecommands (TCs) to trigger satellite functions, separate from any end-user data communication. GSs uplink TCs and downlink Telemetry (TM) during orbital passes, which for LEO last only minutes and occur several times per day. As the

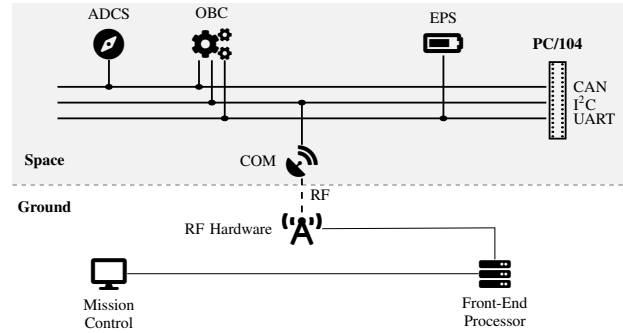


Figure 1. Ground segment (bottom) and Space segment (top). We illustrate the composition of subsystems in a satellite using PC/104.

satellite traverses its orbit, directional antennas at both ends maintain the link; if the platform cannot successfully adapt its attitude, an omnidirectional (usually lower frequency) transceiver often serves as a backup.

In terrestrial radio installations, these TT&C stations are distinct from payload communications stations. Satellite payloads employ high-data-rate transmitters or transponders that relay user traffic without further processing. Besides that, there is a COM for controlling the satellite itself. Payload settings may be adjusted via the dedicated payload link or the platform COM interface.

Satellites often enter beacon mode to maintain connectivity between passes, broadcasting spacecraft status at regular intervals. These so-called beacons are mandatory on amateur-band missions and are used for signal acquisition by the GS. Often beacons carry a small amount of status information about the satellite that is picked up by third-party networks (e.g., SatNOGS [14], TinyGS [17]). TT&C links are typically protected by encryption or authentication mechanisms.

2.2. Small Satellite Subsystems

Unlike larger space missions that typically rely on a more monolithic spacecraft platform from a prime contractor and a payload integrated into it, smaller missions are often assembled from discrete subsystems by the future mission operator itself. Consequently, the conventional distinction between the platform and the payload becomes less pronounced. Instead, small satellites are typically constructed from a collection of interoperable components. These subsystems are selected and integrated based on the specific mission requirements. They are highly integrated and self-contained.

Figure 1 illustrates the space/ground segment architecture of a small satellite mission. Commonly, subsystems are housed in cuboid enclosures with two PC/104 connectors for vertical stacking. The remaining three sides are often used for auxiliary wiring, such as coaxial cables between modules or point-to-point data lines. These interfaces are typically vendor-specific and not interoperable, making the PC/104 connector the primary communication interface within the

1. <https://github.com/EmsecCispa/SatBleed>

stack. While PC/104 provides a mechanical and electrical framework, the actual pin usage is not strictly standardized. Nevertheless, interface conventions exist, with shared buses such as I²C, CAN, and UART (e.g., RS-485) and power rails appearing in predictable locations.

Thus, integrators typically cannot choose which pins are used for which interfaces, leading to multiple devices sharing a bus. This results in a lack of interface isolation, increasing the risk of compromised subsystems, allowing attackers to laterally move to other subsystems.

2.2.1. Subsystem Taxonomy. We follow the taxonomy by Willbold et al. [6] and focus on the security-relevant subsystems:

On-Board Computer (OBC). OBCs provide general-purpose computing capabilities to satellites. There are always-on OBCs, which run critical command handlers and monitor the satellite platform. Typically, this type of OBC utilizes microcontrollers with embedded real-time operating systems such as FreeRTOS [18], Zephyr [19], or RTEMS [20]. Additionally, there are high-end OBCs that may complement the aforementioned systems. These units run on more compute-optimized microprocessors, often Arm Cortex-A SoCs [21]. Frequently, these processors are either part of the payloads or integrated with high data rate transceivers in conjunction with FPGA fabric.

Communication Subsystem (COM). The COM is the gateway between the outside world and the rest of the satellite platform. Thus, it is a basic requirement for the satellite to operate. Usually, small satellites have both omnidirectional communications on UHF/VHF and, optionally, if higher data rates are required, unidirectional antennas with transceivers working in S/X/Ku/Ka-Band. Notably, the same subsystems are often used for the ground-station part of the communication link under the name of Front End Processor (FEP).

Software and Configuration updates to the COM subsystem in particular are risky, as a faulty patch can sever the TT&C link and incapacitate fundamental operations terminally, e.g., thermal regulation, or attitude/orbit control. At the same time, other research has shown that efficient over-the-air (OTA) update mechanisms in general remain an open challenge in these constrained environments [22].

Command and Data Handling System (CDHS). The CDHS subsystem is the combination of hardware and software that processes telecommands. This typically includes software running on an always-on low-power OBC as well as hardware on the satellite that handles low-level commands, particularly for scenarios like hardware failover. In this paper, when referring to the CDHS, we are discussing the system that implements operational commands and executes logic for station-keeping tasks required by the satellite. For CDHS software, there are several open-source and proprietary suites available [23], [24], [25]. Previous papers have already studied the security of those systems

and have concluded that there are general security concerns associated with them [6], [7]. For malicious inputs to reach this hardware and software, they previously have to pass through the COM. This means that in case the COM is enforcing authentication, unauthenticated attackers do not have the capability to reach these components.

Attitude Determination and Control System (ADCS). The ADCS dictates the attitude of the satellite, determining the direction in which it is pointing. To achieve this, it uses sensors to read the current attitude, utilizing the Earth's magnetic field and sun sensors for rough pointing. For more precise positioning, it may employ star trackers. Control over the attitude is then performed by magnetorquers (electromagnets that react against the Earth's magnetic field). These are either used to directly influence the attitude or to desaturate reaction wheels, which can make changes to the attitude more quickly [26]. This makes the ADCS an interesting target for attackers, as they need control over a satellite's attitude to point a directional antenna, prevent sensors from capturing data, or deliberately cause damage to the spacecraft.

Electrical Power System (EPS). The EPS is the system that coordinates the generation and use of electrical power. This includes control over solar charging and switching power rails. Usually, the EPS also handles keeping the batteries within their thermal limits, activating electric heaters for the battery cells when necessary. This makes it an interesting target for sabotage if an attacker aims to artificially cause a satellite mission to fail prematurely.

2.3. Satellite Network Architecture

The simplest type of frame in a space protocol encapsulates a single packet containing one command. This model abstracts the satellite as a command-driven device, receiving TCs as the uplink from the ground segment and providing a response via TM as the downlink [27].

Internally, these commands are disseminated through the onboard communication architecture and parsed by the subsystems responsible for their execution or directly interpreted by the COM. A common drawback of satellite networks is the lack of privilege separation within the onboard software stack. All commands are treated with equal authority once accepted by the satellite, lacking separation of different privilege levels to limit access to sensitive commands and telemetry data. We discuss some specific space protocol designs in Section 5.

3. Criticality of the COM System for Satellites

We discuss the motivation for examining the security of COM systems, focusing on their critical position within the wider satellite ecosystem and the existing research gaps surrounding them.

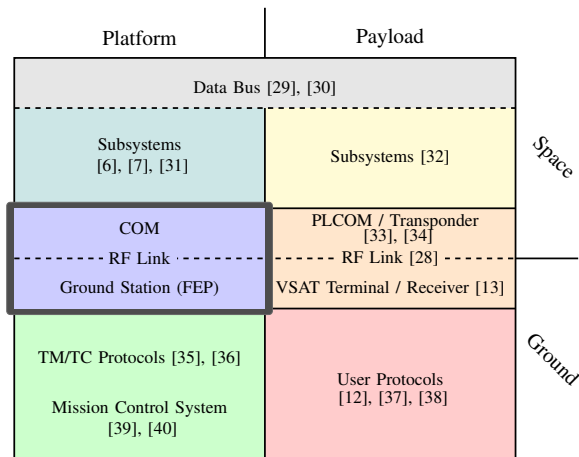


Figure 2. Illustration of the different aspects of space cyber security research. Different targets of space cyber security research have been colored. The highlighted area represents a research gap we aim to fill.

3.1. Systematization of Satellite Security Research

First, we can distinguish *platform* and *payload* with their respective subsystems. Second, we can distinguish ground and space systems. Finally, data buses, interfaces, and links between space and ground are a separate concern. Based on these dimensions, we create a taxonomy comprising all aspects of these subsystems and systematize existing research as illustrated in Figure 2.

3.1.1. Space.

Data Bus: Existing work has analyzed vulnerabilities of high-trust data buses such as CAN or MIL-STD-1553B used in Satellites [29], [30]. Similar to unauthenticated use of such bus systems in aircraft or cars, an attacker with access to the bus can conduct Denial of Service (DoS) and spoofing attacks.

Platform: Subsystems Prior studies systematically analyzed onboard platform components and flight software such as On-Board Computers, Attitude Determination and Control Systems, and Electrical Power Systems, revealing vulnerabilities in firmware update mechanisms and insufficient interface isolation [6], [7], [31].

Payload: Subsystems Previous research on space-segment payload components, including high-data-rate transceivers and sensor interfaces, has identified implementation flaws, and parsing errors [32].

3.1.2. Ground.

Platform: TM/TC Protocols Link-layer Telecommand and Telemetry protocols have been subjected to replay, spoofing, and framing attack analyses, which demonstrate the prevalence of weaknesses in authentication and integrity checks [35], although some implementations have shown fewer obvious weaknesses [36].

Platform: Mission Control Systems Security evaluations of Mission Control System software have uncovered

insufficient access control mechanisms, vulnerabilities in scheduling components, and inadequate logging practices, all of which enable potential unauthorized command injection [39]. Other works have captured malicious interactions with exposed MCS infrastructure [40].

Payload: User Protocols Investigations into end-user satellite broadband and data-service protocols have demonstrated DoS, and confidentiality flaws, leading to proposals for protocol hardening and enhanced encryption strategies [12], [37], [38].

3.1.3. Space-Ground Interfaces.

RF Links: Researchers have investigated payload RF links [41]. Wider surveys on SATCOM RF security such as [28] discuss the issue more generally from a lower-layer perspective.

Payload: PLCOM / Transponder Detailed analyses of payload transponders have revealed that their radio frequency links are susceptible to link-layer attacks such as frame manipulation and jamming [33], [34].

Payload: VSAT Terminal / Receiver Assessments of Very Small Aperture Terminal (VSAT) receivers have exposed hardware and firmware vulnerabilities that adversaries could exploit to compromise commercial user terminals [13], [42].

Platform: COM / TT&C Ground Station Currently, there is no security evaluation of TT&C transceivers or ground-station RF hardware in the literature, as we discuss next.

3.2. Research Gap: COM Systems and TT&C Links

As shown in Figure 2 and our systematization, existing work covers each quadrant, yet we observe a distinct gap in the platform's space-ground interface, even though it represents a high-impact attack surface. There is no work investigating COM components, either space-based or ground-based (FEP), used for TT&C, even though they are the ideal targets for attackers to conduct malicious actions. Possible reasons for this might be that in larger customized satellites, these subsystems are highly sensitive, and therefore, gaining information on them is difficult.

3.3. Importance of the Communication System

Among all subsystems, the COM is paramount, as it provides the sole link for command uplink, telemetry downlink, and, in some cases, internal message routing between modules. It thus occupies a central position guarding the satellite's perimeter. The COM is inherently exposed via an external RF link and therefore cannot be shielded by additional filtering, while maintaining direct access to internal data buses where subsystems exchange messages under the assumption of a trusted environment.

In practice, attitude-control limitations and imprecise antenna pointing lead many satellites to rely on omnidirectional links [43]. While these allow signal reception without active pointing, they also enable drive-by exploitation. When an adversary compromises the COM subsystem, they can control telemetry and telecommand exchange with the ground segment. Even a simple DoS vulnerability may render a satellite uncontrollable and cause mission failure.

The costly, slow, and one-shot nature of launches, combined with reliability requirements, often leads engineers to prioritize safety over security. As a result, diagnostic backdoors and "clear modes" [44], [45] are routinely included to ensure mission continuity. Regarding link protection, little research evaluates the correctness of these often proprietary protocols.

COM subsystems vary significantly in complexity. Simple transceivers send raw frames in near real time, while more advanced units parse packets, handle fragmentation, and maintain routing tables. Store-and-forward devices buffer telemetry for scheduled downlink. These systems may be the only barrier preventing unauthenticated control of the satellite. If such functionality is not hardened, the attack surface grows substantially.

Similar concerns apply to RF transceivers in commercial GS appliances, often called FEP. Vendors typically provide matching GS hardware for their COM subsystems, connected to a RF front-end to communicate with the satellite. This tight coupling means both the spacecraft and its TT&C GS may share vulnerabilities. Attacks from ground-based adversaries targeting the GS and space-based adversaries targeting the spacecraft can remain stealthy, as they can use very low RF power, often undetectable by monitoring authorities.

A final concern is the opacity of the software executing on COMs. Once a component of a satellite has been compromised, it is almost impossible to reset because the usual recovery mechanisms are built with safety in mind and do not withstand an adversary who deliberately manipulates a system. With subsystems allowing lateral movement, an attacker could still keep persistent access without further measures. None of the COMs analyzed features a resilient fallback mechanism.

This paper therefore presents a detailed analysis of hardware, communication protocols, firmware, and operational practices.

4. COM Subsystem Threats

As attacks on space-based assets evolve, a new attacker ecosystem may emerge. The increasing number of spacecraft, and thus identical COMs, enables attack scenarios that depend on a high density of vulnerable satellites.

In this section, we argue that the COM is central to these emerging threats and outline how a more mature attack ecosystem could develop as adversaries pivot to vulnerable small satellites. Given the typically limited lifetime of LEO small-satellite missions, these risks can still be addressed in future designs before widespread exploitation occurs.

4.1. Novel Threats to Space-Based Assets

4.1.1. Space Botnets. A monoculture of COMs implies that a single vulnerability can rapidly compromise a substantial fraction of the satellite population. The threat is exacerbated for platforms that are able, either by design or by coincidence, to conduct satellite-to-satellite communications. Under such conditions, self-propagating "space worms" can emerge, exploiting the same flaw moving between satellites. Propagation speed is only limited by orbital dynamics and the number of targets. Once initiated, the process is effectively unstoppable and may eventually encompass the complete satellite population, potentially without immediate observable effects. The outcome is a large-scale, practically untraceable attack against space-based infrastructure. Similar to DoS botnets on the internet, constellations of radio-frequency jammers consisting of compromised COM systems could show up in space, as suggested in [11]. Once activated, these devices could emit interference that cannot be disabled from the ground. Such a network could systematically block critical downlink or uplink signals [46], thereby disrupting mission operations or degrading the service quality of other satellites. The effectiveness of this threat is amplified, as the COM subsystem usually does not tolerate any other transmitter on the same channel as stated in Section 5.3.

4.1.2. Satellite Spyware. Satellites are designed with the expectation that subsystems may fail, making redundancy a key architectural principle. To enable rapid fault recovery, platforms include multiplexers that allow operators to switch from a primary component to a cold-redundant backup. An advanced adversary can exploit this mechanism to establish persistence. After compromising a COM or OBC, malware may laterally move to a normally powered off backup unit. The attacker can then simulate a failure of the primary element, enabling persistent use of the satellite's main functionality without operator awareness.

The attacker can hide additional power consumption by impersonating the EPS on spacecraft data buses or by corrupting EPS telemetry so reduced battery capacity appears as natural degradation. From the backup unit, spyware can passively collect payload data, inject falsified commands, and exfiltrate information via covert channels embedded in normal downlink frames. Detection is difficult because the redundant unit is expected to remain offline unless needed and may appear to have simply failed. As a result, operators may never realize the attacker maintains a foothold that can be reactivated at any time.

4.2. COM Subsystem Sub-Components

To better understand architectural weaknesses in COM systems and enable a systematic analysis, we derive a threat taxonomy. Existing taxonomies, such as [6], operate at the component level and lack the granularity needed to capture all threats to the COM and corresponding FEP. To address

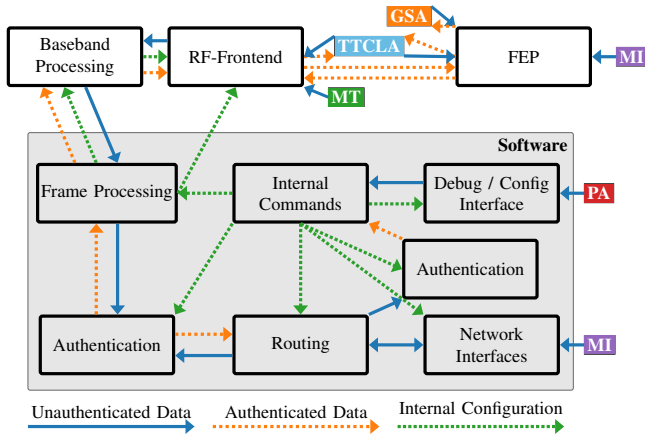


Figure 3. Sub-component based view of data/attack flow for the Communications Subsystem with secure reference data flows.

this, we analyze the system features of 6 COMs and derive a sub-component-based model, shown in Figure 3.

We model information flow within an idealized COM architecture, which can then be compared to real implementations to reveal architectural weaknesses and the consequences of vulnerabilities. Vulnerabilities can be classified by affected sub-components, and exploit chains visualized through data flow. Because COMs involve complex hardware-software interactions, both aspects and their interplay must be modeled.

As shown in Figure 3, numerous entry points exist for unauthenticated, potentially malicious inputs into COM sub-components.

COMs and FEPs face multiple threat vectors. Attackers may directly target systems via the RF link to disrupt operations or gain a foothold. Once access is obtained, they may seek persistence by embedding malicious firmware. Another risk arises when adversaries intercept and modify TC or TM. Based on these scenarios, classify relevant attacker types for COM security.

4.3. Attacker Model

For a satellite mission that contains a model of COM and corresponding FEP, we distinguish the following attacker types according to their capabilities.

PA Short Duration Physical Access Attacker. The attacker can tamper with hardware for a short time period before launch or during development at the supplier. This includes physically manipulating components to access interfaces that are typically not exposed.

TTCLA TT&C Link Adversary. The attacker can intercept the legitimate TT&C uplink from the ground station and operate their own ground station that can communicate with the satellite. An eavesdropper positioned outside the secured perimeter of a ground station can exploit side lobes to capture telecommands [42], analyze them, and replay them at a different time and location. Note that

this attacker does not require the capability to prevent the delivery of messages transmitted.

MT Malicious Transmitter. The attacker can transmit arbitrary messages to the target satellite or GS. Such capabilities can be obtained by repurposing amateur-radio equipment and commercially available hardware that may be ordered online [47]. This category also covers benign ground stations that are compromised and subsequently abused, as well as ground-station-as-a-service offerings operated with malicious intent. Alternatively, the attacker can operate or compromise another satellite to pose a threat via inter-satellite communication.

GSA Ground Station Attacker. The attacker can send arbitrary signals to the GS and receive responses from the GS. The GSA represents a weaker version of the **TTCLA** that is explicitly attacking the GS from the ground and cannot reach the satellite.

MI Low Privileged Malicious Insider. A low-privileged participant in a satellite mission who can cause arbitrary, validly authenticated data to be transmitted via the COM. This attacker might have the ability to inject data through either the satellite or the GS communication systems. For example, they may have access to the FEP or to a computer running mission-control software. Alternatively, in a multi-payload mission, the attacker could be embedded as one of the payloads. In such cases, shared bus architectures may allow them partial access to other payloads.

We generally always name the least capable attacker who could perform an attack. Attackers can be ordered according to the hierarchy below, where stronger attackers imply weaker ones.



4.3.1. Modeling Restrictions. Executing these attacks in practice requires adversaries to overcome significant operational constraints. An attacker must account for narrow timing windows, as LEO orbital passes typically last only minutes, depending on the orbit and the attacker's location, and occur only a few times per day. Furthermore, exercising **Malicious Transmitter** **MT** capabilities (as the weakest attacker of this type) requires constructing a directional antenna, managing Doppler shift, and possessing sufficient transmit power to reach the satellite. While this is easier in VHF/UHF than in other bands due to lower equipment requirements and a smaller absolute Doppler shift, tuning a setup for this is not trivial. Finally, persistent malicious transmissions risk detection of their geolocation by spectrum monitoring authorities. While we abstract these factors to evaluate the underlying hardware weaknesses, they represent practical hurdles that dictate the adversary's necessary sophistication and resources.

4.4. Generalized COM Threat Taxonomy

A taxonomy of threats is visualized in Figure 4. Building on the previously discussed novel threat scenarios, we now shift the focus from large-scale, persistent threat scenarios in Section 4.1 to the underlying threats to the COM that enable such attacks.

To systematically capture these weaknesses, we derive a generalized vulnerability taxonomy for COM systems. The taxonomy is based on the sub-component oriented view of the COM introduced in Figure 3, which models both data flow and trust boundaries (e.g. via Authentication) across hardware and software elements. Each hypothesized threat in the nodes of Figure 4 is associated with the specific sub-component it affects and is further annotated with the weakest attacker model presented in Section 4.3 required to exploit it. Attackers are chosen under the assumption that the underlying data flow of a COM system follows Figure 3 and thus might allow to be exploitable by a weaker version of the annotated attackers for non-ideal implementations in real-world COM devices.

5. Evaluation of COTS COMs

To evaluate the practical validity of our threat taxonomy, we apply it to three real-world COM subsystems. The complete data flows of the analyzed systems is presented in the appendix B.

The threat taxonomy outlined in Figure 4 actively guided the evaluation methodology in this Section. By systematically evaluating the data flows against the threat vectors defined for each sub-component (e.g., Frame Processing, Routing), we proactively identified attack classes, such as the Syncword-based DoS and architectural routing leaks, that might have been overlooked in an unstructured review. Furthermore, because this taxonomy models the core architectural elements of spaceborne communications, it serves as a generalized framework that can be applied to systematically evaluate the security of other COM subsystems across the broader NewSpace ecosystem.

We perform the evaluation under the assumption that the operators made a best effort to secure their systems; we exclude issues arising from fundamentally insecure configurations.

5.1. Identifying Critical Communication System Vendors

To focus our security analysis on the COM modules whose compromise would have the greatest operational and safety impact and to ensure our findings generalize across the small satellite market, we first define selection criteria. Using these criteria, we identify three vendors whose COM, providing a meaningful sample for uncovering systemic weaknesses.

Selection Criteria

- **Continuous Receive / Fallback Operation:** The transceiver must support a low-power continuous receive mode, allowing it to remain active at all times, fitting the role of a COM that will be used as an always active fallback, receiving omnidirectionally. This makes a shutdown as a defense infeasible.
- **Obtainable COTS:** We focus on components obtainable COTS by private third parties, excluding vendors that are exclusively selling to government or military customers.
- **Market Relevance:** The COM must be relevant in the COTS market. We want to examine representative targets that are of interest to attackers.
- **Business Model Diversity:** The list must include both vendors with closed and open ecosystems, ranging from communication-as-a-service providers to those offering standalone hardware for integration with third-party systems. This diversity enables an evaluation of how different integration models impact product security.
- **Security Features:** Each product must incorporate a feature aimed at preventing the issuing of commands from unauthorized parties. We exclude COMs that require a different system to handle TC authentication.
- **VHF or UHF Frequencies:** For each vendor, we focus on VHF/UHF. These bands lower the barrier for adversaries who attempt to build their own ground station and align with the low-power always-on requirement.


5.2. Selected Vendors and Components

For the case study, we selected three COMs from three different vendors based on our selection criteria:

GomSpace NanoCom AX100: A transceiver on the market since 2014 and the only VHF/UHF offering by GomSpace. According to the SatNOGS database, at least 64 active satellites use the NanoCom AX100, with 174 entries recorded as of July 2025. Our analysis shows this COM is deployed across diverse commercial missions, including maritime situational awareness [48], signal intelligence [49], quantum key distribution [50], and a military-funded ICBM launch detection test mission [51]. It is also used in orbital transportation services [52]. Consequently, a vulnerability could, in the worst case, enable the takeover of a transport vehicle and prevent payload deployment, thereby amplifying the impact. Based on public data, GomSpace holds roughly 15% of the CubeSat COTS component market [53] and is consistently ranked among the top vendors [54], [55], [56], [57].

Endurosat UHF II: More than 3000 Endurosat modules are in orbit as of November 2025 [58], with over 120 full satellites delivered. The company offers highly integrated solutions, primarily using proprietary protocols while supporting some standards such as Cubesat Space Protocol (CSP) for the OBC. The UHF II is its only UHF offering, providing low rate basic communication. This vendor is

Hardware		Software					
Baseband Processing	RF-Frontend	Internal Commands	Network Interfaces	Authentication	Routing	Frame Processing	Debug / Config Interface
State Corruption	Cause Physical Damage	State Corruption	Packet Smuggling	Algorithm Side Channels	Memory Corruption	Memory Corruption	Reprogramming
MT	MI	MI PA	MI	TTCLA	MI	MT	PA
Jamming	Jamming	Privilege Escalation	Identity Spoofing	Weak Authentication	Identity Spoofing	State Corruption	Config Modification
MT	MT	MI PA	MI	MT	MI	MT	PA
Resource Exhaustion	Signal Leakage	Memory Corruption	Information Leak	Credential Compromise	Information Leakage	Frame Smuggling	Memory Corruption
MT	MT	MI PA	MI	GSA MT PA	MI PA	MT	PA
			Memory Corruption	Memory Corruption		Resource Exhaustion	
			MI	MT PA		MT	

Figure 4. Taxonomy of subcomponents, threats, and discovered vulnerabilities. Each hypothesized **threat** (body columns) is annotated with the weakest required attacker model. Each threat with a corresponding vulnerability in Table 1 is marked with .

notable for multi-tenant satellite missions [59], where preventing lateral movement is critical, as attackers targeting one payload may compromise others or the COM. Like GomSpace, Endurosat is consistently ranked among leading small-satellite vendors [54], [55], [56], [57].

Libre Space Foundation SatNOGS-COMMS: The SatNOGS-COMMS is a fully open-source hardware COM used in past missions and planned for future ones [60], [61]. Unlike designs that use separate devices on PC/104 or carrier boards, it integrates both S-Band and UHF transceivers into a single unit. Our analysis is based on the current SatNOGS-COMMS main branch and Curium-1 flight software [60]. We focus on the UHF section, as the S-Band component does not meet our criteria in Section 5.1. Given the rapid adoption of prior Libre Space Foundation projects such as the SatNOGS GS network [14], especially by cost-sensitive operators, we expect many future designs to build on this module.

To demonstrate the plausibility of the outcomes in Section 4 and the applicability of a sub-component-based taxonomy, we conducted a security evaluation with the goal of proving that the outcomes mentioned can be achieved by attackers according to our threat model. For this, we evaluated the security of the COM systems previously identified as critical in Section 5.1. We approached the systems from the perspectives of the attackers introduced in the threat model, investigating flaws in different key aspects. Where possible, we also looked at the corresponding FEP product that is responsible for the terrestrial side of the TT&C link. For each COM vendor, we first provide context on how this is used and how the protocols are supposed to work to get an intuition about how the vendor expects clients to use the systems. Then we discuss any weaknesses identified during our security evaluation. Finally, we map the vulnerabilities discovered to our threat taxonomy. You can find an overview of vulnerabilities, including the corresponding attackers and affected models, in Table 1. Appendix Section B contains data and attack flows for the investigated COMs. We also provide vulnerability mappings².

2. <https://github.com/EmsecCispa/SatBleed>

5.3. GomSpace NanoCom AX100

The GomSpace transceiver is based on FreeRTOS 8.0.1 [18], released in 2014 [62], running on an AVR32 microcontroller and lacks any security hardening such as stack-smashing protection.

The NanoCom AX100 operates as a CSP node. CSP is a lightweight protocol stack for small satellites, mirroring the Open Systems Interconnection (OSI) model and treating nearly all spacecraft components as network participants [63]. It spans multiple physical and data link layers, including I²C, CAN, SPI, UART (RS-232), and Ethernet, enabling deployment across heterogeneous hardware. GomSpace builds all devices around this stack.

CSP routing is static, with each node maintaining a configurable routing table. Subnets often separate spacecraft nodes from ground infrastructure, allowing the MCS to send commands addressed to spacecraft nodes into the network, which are then forwarded to the CDHS. Packets may also be sent directly to subsystems, with intermediaries acting as routers. On multi-device buses such as I²C, CSP permits address reuse in underlying protocols [63]. Depending on the architecture, the COM may act as a node with its own address and services or as an interface for another node (e.g., the CDHS).

The GomSpace NanoCom AX100 uses a custom RF framing mode, *ASM+Golay* (named Mode 5), replacing older, now unsupported modes such as AX.25, to transport CSP packets. It typically functions as a CSP node with multiple interfaces and can route traffic (e.g., I²C to CAN). The radio is treated as an interface and can enforce a truncated SHA-1-based HMAC. Packets with missing or incorrect HMACs are discarded regardless of the packet's HMAC flag.

Unlike libCSP, the NanoCom AX100 does not support XTEA-based encryption or encrypted tunnels to explicitly protect the RF link from unauthorized access. Therefore, RF link encryption cannot be implemented using only the COM. An additional device between the PC/104 bus and the COM is required, as in the OPS-SAT mission [6], where the CDHS acts as a gateway.

The NanoCom AX100 provides seven services on ports 0–6, including default libCSP services [63], and exposes

TABLE 1. OVERVIEW OF VULNERABILITIES DISCOVERED

Vendor	Segment	Section Vulnerability (Type ¹)	Attacker (Subcomponent)	Exploitable ²	Confirmed ³	Affected Models ⁴
GomSpace	Space / Ground	5.3.1 Open Debug Port (Impl.)	PA (Debug / Config Interface)	✓	✓	2 (AVR32)
		5.3.2 Replay (Design)	TTCLA (Authentication)	✓	-	3 (CSP HMAC)
		5.3.3 Dangerous TC (Design)	MI (Internal Commands)	✓	✓	3 (CSP)
		5.3.4 Command Injection (Impl.)	MI (Network Interfaces)	✓	-	2 (GOSH & CSP)
		5.3.5 Memory Corruption (Impl.)	MI (Debug / Config Interface)	✓	✓	2 (GOSH)
		5.3.6 Side Channels (Impl.)	TTCLA (RF-Frontend, Auth.)	?	?	-
		5.3.7 Insufficient Signature Length (Design)	MT (Authentication)	✓	-	3 (CSP)
		5.3.8 Debug Backdoor (Design + Impl.)	MI (Internal Commands)	✓	✓	3 (CSP)
		5.3.9 Syncword Based DoS (Impl.)	MT (Frame Processing)	?	?	-
		5.3.10 SATBLEED (Impl.)	GSA (Routing)	✓	✓	1 (AX100 in GS)
Libre Space Foundation	Space	5.4.1 Insufficient Signature Length (Design)	MT (Authentication)	✓	-	1
		5.4.2 TC Authentication Bypass (Design)	TTCLA (Authentication)	✓	-	1
Endurosat (deprecated)	Space / Ground	5.5.1 Open Debug Port (Impl.)	PA (Debug / Config Interface)	✓	✓	2 (MSP430)
		5.5.2 Insecure Update Service (Design)	MI (Internal Commands)	✓	✓	2 (FWUP)

¹ Design = design flaw; Impl. = implementation flaw. ² Realistically exploitable? ³ Vendor-confirmed or PoC-demonstrated. ⁴ Affected models per vendor.

them via the RF interface. Thus, even if routed traffic is encrypted, the transceiver’s own services remain externally accessible unless HMAC is enabled. In OPS-SAT [6], these services were accessible to unauthenticated attackers, meaning that even if the CDHS vulnerability was fixed, the COM would remain exposed.

The NanoCom GS100 UHF ground station FEP by GomSpace shares the same hardware as the NanoCom AX100 in a rack mount device, and firmware details suggest that it is shared among the ground and space segments. In the following, we expect the COM to be used with its official ground station product. We now summarize our findings for the GomSpace COM.

5.3.1. Open Debug Port PA. The device has an open JTAG debug interface and serial debug shell exposed on the outside of the enclosure, allowing attackers with short physical access to tamper with the onboard software of the device.

5.3.2. Replay TTCLA. CSP does not sufficiently protect against the replay of packets. This would have to be enforced by a higher layer in the network stack. Critically, the services on the COM itself do not feature additional replay protection. This means that even if the HMAC verification is enabled with a sufficiently long key, an attacker could still replay critical messages, such as changes to the parameters that contain settings. Note that packets that control the COM internal services cannot employ encryption, meaning that an eavesdropper knows exactly what changes a certain packet causes. Then an attacker could compose multiple legitimate changes to parameters and render the mission unable to be commanded effectively, causing persistent DoS.

5.3.3. Dangerous TC MI. An attacker obtaining a single authenticated TC with their packet payload would already be sufficient to compromise the mission. The attacker could forge a command that reconfigures the COM to disable authentication enforcement, effectively bypassing future security checks. Furthermore, there are many TCs that can

be set to values that lead to persistent DoS, or are simply usable in a manner that renders the system insecure. Also, the rekeying procedure relies on the CDHS encapsulating the new key in an encrypted message, as commands directly to the COM cannot be encrypted. Thus, the only way to securely rekey the COM is by sending the new key to the CDHS in an encrypted message, followed by the CDHS setting the key in the COM using the internal network interface of the COM.

5.3.4. Command Injection MI. By redefining the configuration of the COM, an attacker can modify onto which UART interface the *GOSH* debug shell is exposed. This enables command injection into said shell, which is intended solely for configuration and debugging of the COM. When another CSP node is connected to the COM via RS-232, any frame sent from that node is interpreted by the COM as a debug command. An attacker could exploit this behavior by having a connected node reflect a payload into the debug shell, for example, by using the default *CSP_PING* service, which echoes back the payload it receives.

5.3.5. Memory Corruption MI. The *GOSH* debug shell has the capabilities of sending arbitrary packets and modifying arbitrary parameters and memory. There are also *peek* and *poke* commands that supposedly allow reading and writing memory. The *poke* command in the shell, however, seems not to work in the way we expect. Despite this, an attacker could leverage various debug shell features to gain persistent access or achieve code execution. One example is modifying security-related parameters. The *peek* command contains a memory corruption vulnerability that may also be used to gain code execution.

5.3.6. Timing and RF Leakage Based Side-Channels TTCLA. The GomSpace version of the CSP stack features the same memcmp-based key comparison as in the open-source version libCSP [64]. The firmware uses non-constant-time functions for cryptographically sensitive operations such as comparing the result of the HMAC computation

to the HMAC field in an incoming packet. Furthermore, the transmitter appears to be leaking computation-dependent data over RF during transmission. Figure 8 in the appendix shows that when sending a carrier, the RF emissions differ based on the workload that the processor is currently running, indicating leakage similar to the one described by Camurati et al. [65]. It is unclear whether this is exploitable to further reduce the search space for attacks either attacking the signature of a single packet or the key directly. During reverse engineering, we discovered that every time the firmware needs to compute an HMAC, it retrieves the stored master key and re-runs the key-derivation function rather than deriving the HMAC key once and keeping it in memory for reuse. This design both wastes processing cycles and expands the window for potential side-channel leaks.

5.3.7. Insufficient Signature Length MT. The SHA1-HMAC employed in CSP is truncated to a 32-bit length. On slow links, performing an exhaustive online brute-force attack is generally infeasible due to the low data rate. However, the protocol is also supported on high data-rate transceivers that operate in the multi megabit per second data-rates [66], which makes sending 2^{32} packets a feasible attack. This becomes particularly feasible if an entire constellation shares the same key, allowing guesses to be balanced across many satellites and their ground stations. Assuming a malicious ground station is able to achieve a data rate of 50 Mbit/s, an online attack to forge the authentication tag could take under 10 hours.

5.3.8. Debug Protocol Allowing Arbitrary Memory Write MI. The *CMP* service defined in the open-source libCSP allows arbitrary memory write. However, this functionality shall be disabled or replaced by weaker functionality. In the implementation, it is explicitly marked as extremely dangerous [63]. The GomSpace implementation of the *CMP* service also has this feature; it uses a function that allows access to parameter memory. However, the memory address of the function pointer that gets called to do the memory access is also accessible via this service. This means that an attacker could use this to gain arbitrary code execution or create an arbitrary memory read/write primitive. The vendor did not confirm that this is on purpose and instead answered that this protocol is not supposed to be used. This means that an attacker who can have data processed by the COM can gain code execution.

5.3.9. Syncword-Based DoS MT. In response to the unusually complex sync-word detection approach, likely intended to extend the capabilities of the off-the-shelf RF IC, we developed a fuzzer, *SyncWordFuzzer*, targeting physical layer protocol bugs. The tool generates streams of symbols containing sync-words and occasional valid frames for the system to process. Rather than merely using the RF layer to feed higher protocol layers, it produces inputs designed to trigger faults in the lowest level of frame handling.

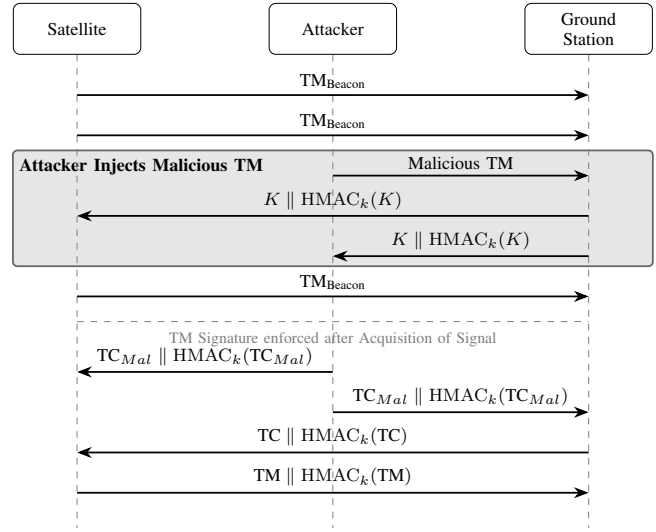


Figure 5. Sequence diagram of an attacker performing a SATBLEED type attack. TM injection into the ground station while it waits for the beacons of the satellite. The line after which TM signatures are enforced may not exist, depending on the mission; however, if a switch is present, it may occur after beacons have been received.

While running *SyncWordFuzzer*, we observed error messages related to RF chip and MCU communication. Over time, the setup triggered a condition where the system continuously received frames at what appeared to be its maximum rate, even without a transmitter. This behavior is distinct from rare idle cases where random noise produces spurious frames.

The effect is likely caused by a race condition that misassigns responsibility for sync-word detection. Under normal operation, the radio IC detects the sync-word and interrupts the MCU, which then handles detection for subsequent frames in software. However, during fuzzing, the COM intermittently stops waiting for a sync-word and treats all incoming data as valid frames. This likely occurs because both the software and RF chip enter a state where each assumes the other is responsible for detection, causing all RF input to be accepted as frames. Since these spuriously detected frames contain random payloads without valid length fields or signatures, they would go unnoticed in practice, aside from degraded reception due to incorrect framing.

This “syncword bug” appears independent of specific physical layer inputs, as timing-accurate replays do not reliably trigger it. Instead, it likely depends on internal timing conditions combined with certain input sequences. This behavior highlights how unconventional transceiver designs can introduce a new class of unauthenticated DoS vulnerabilities.

5.3.10. Logic Error in TM Handling (SATBLEED) TTCLA. GomSpace utilizes the same module in both the satellite and the ground station device, with identical functionality in the ground and space versions of the NanoCom AX100. However, these two deployment scenarios should not necessarily behave identically. In particular,

the ground version exposes the same TC interface over RF as the space version. As a result, the most critical attack we found, SATBLEED (described in Figure 5), becomes possible. An attacker can inject a parameter request into the ground station even though the ground station has not sent a corresponding command. In our attack, the injected request asks for the key used to sign TC. In principle, the ground station should discard this nonsensical request of the spacecraft, which is effectively the spacecraft asking the ground station for its own key, yet a response is nevertheless sent. This should not be understood as merely a consequence of missions disabling signatures. In practice, signatures or encryption may be enforced at a higher layer, with the cryptographic processing equipment placed behind the FEP. In such a setup, the FEP would normally not interpret TM, since, unlike the COM, there is no need to expose a command interface for telemetry over the RF interface, as such an interface is only needed for recovery purposes of the COM. Consequently, the vulnerability persists even in the presence of encrypted TM. We further discuss the impact of SATBLEED in Section 6.

Finally, we noticed that the NanoCom AX100 uses Carrier Sense Multiple Access, and does not start transmitting if it detects an RSSI greater than -95dbm by default, making jamming an effective measure as it will cease transmitting when a low-power signal is received by it.

5.4. Libre Space Foundation SatNOGS-COMMS

The SatNOGS-COMMS is a COM by Libre Space Foundation. The product is a unified UHF and S-Band hardware. It utilizes a combination of an Xilinx Zynq Cortex A9 SoC and an Arm Cortex M7 based microcontroller that is always powered on.

5.4.1. Insufficient Signature Length MT. The SatNOGS-COMMS uses a TC protocol based on Protobuf [67]. When transmitting the same commands over RF, a 4-byte authentication tag is added to each packet. Similar to Vulnerability 5.3.7, the length of this tag may not be sufficient. On top of this framing layer, a mission can add whichever protocols it desires. One of the missions continues by defining commands as single-frame commands for basic operations. It seems plausible that a more complex mission adapts protocols such as CSP.

5.4.2. Authentication Bypass by Spoofing TTCLA. The authentication for the SatNOGS-COMMS is based on a secret that is appended to a TC when transmitted over RF. The secret is then hashed and compared against a stored result. In case the check passes, the frame is processed. This means that if the attacker observes this secret once, they can append it to any payload and have it be authenticated. The key does not change, so a passive eavesdropper can intercept a single TC and use that to send arbitrary TCs. However, this does not mean an attacker can gain persistence, as Over-the-Air updates are cryptographically verified.

5.5. Endurosat UHF II

From Endurosat we analyzed the UHF II transceiver. Unlike the open protocols CSP and the protocol used by the SatNOGS-COMMS, Endurosat uses the so-called Endurosat Protocol Stack I. It is a proprietary protocol stack that lacks an open-source implementation and has no publicly accessible documentation. There do not seem to be any other vendors besides Endurosat using it. Some layers of Endurosat Protocol Stack I can be used to transport CSP to ensure compatibility.

The UHF II has two modes of operation: in one, it behaves like a transparent RF interface, while in the other, it functions as a network node that can be configured over the air. When Endurosat Protocol Stack I data is transported over the radio, there is a special encapsulation layer that carries encrypted data and thus does link encryption. Similar to the CSP, there is an address space of addresses that are used for devices on the network. Also, there is an initial broadcast when joining a network.

After manually reverse engineering the device's circuits and firmware, we chose partial re-hosting for fuzzing the firmware. Additionally, we reverse-engineered an undocumented update protocol, as this system, or a similar one, is used in multi-payload missions where an attacker on an internal data bus is a plausible threat.

5.5.1. Open Debug Port PA. The Spy-Bi-Wire debug and serial bootloader of the microcontroller is accessible without authentication. This allows attackers with short-term access to manipulate firmware. Afterward, there is no capability of restoring the firmware to a known good state.

5.5.2. Firmware Update Without Integrity Check MI. The firmware updates are not cryptographically signed and thus do not protect against a compromised device on the bus pushing a malicious update. Note that updating the software does not imply resetting the device's configuration. Encryption keys stored on the device may be exposed to attackers after gaining code execution through this protocol, sidestepping the limitation of key material being a write-only configuration. Software components also suggest the existence of a second undocumented protocol, which we did not investigate further, as we have already demonstrated that a local attacker can gain code execution.

6. Feasibility and Impact of SATBLEED

To better understand which active systems may be affected by the critical SATBLEED vulnerability described in Section 5.3.10, we estimate how many missions using this COM transmit telemetry beacons in configurations where the GS100 cannot enforce an HMAC field. It is important to distinguish between the existence of the SATBLEED vulnerability, which we have conclusively confirmed, and its impact on active missions, which can only be inferred heuristically.

Because no public information exists on how operators configure and deploy the NanoCom AX100, we rely on publicly available observation data from the SatNOGS project [14]. This open-source dataset includes telemetry beacon recordings from 80 transmitters using the GomSpace proprietary Mode 5 framing with FSK/MSK modulation, corresponding to 64 active missions as of August 2025. We infer the use of the NanoCom AX100 because it is the only known device implementing this mode, and thus attribute these transmissions to missions using this COM, which we decode and analyze as detailed in Appendix A.

Our analysis is based on several assumptions. First, we assume in-orbit COMs and FEPs run firmware versions equivalent to those available in April 2025, in which we identified the vulnerability. Second, we assume the vendor does not provide mission-specific hardware or firmware variants, as we found no evidence of such deviations. Third, we assume the ground infrastructure is vendor-provided and plausibly based on the GS100, representing the simplest deployment model.

However, beacon observations do not reveal all configuration parameters. They do not indicate whether encryption or authentication is applied to all telemetry, nor do they prove the presence or absence of signatures on non-beacon traffic. Instead, they only reflect which ground-station configuration is active for a mission at a given time, which is still sufficient to infer security-relevant behavior.

Under these assumptions, 28 of the 64 active missions in the dataset are plausibly vulnerable to SATBLEED. Of these, 12 transmit beacons without GS100-enforceable HMAC protection, while 16 frequently toggle this configuration.

This methodology has inherent limitations. Most importantly, it relies entirely on passive observation rather than active validation, so the result should be interpreted as a lower bound, with the true number likely higher. Some missions use encrypted or compressed beacons, complicating reliable classification. Moreover, telemetry encryption does not prevent SATBLEED, as the COM processes commands before forwarding them to subsystems that may perform decryption. Finally, our analysis is limited to missions observed by the SatNOGS network.

7. Contextual Factors Behind Security Gaps

Testing Challenges. Security testing of COMs is inherently difficult. The radios tightly integrate hardware and software and must remain operational in orbit despite progressive degradation. Firmware therefore includes extensive coherence and self-diagnostic mechanisms, as partial subsystem failures are expected due to cumulative ionizing dose over time. These constraints limit automated partial re-hosting approaches, where the CPU is emulated without peripherals.

When firmware continuously polls peripherals and timing-critical tasks are shared between the RF front-end and MCU, such as sync-word detection handover, accurate emulation requires precise hardware models unavailable outside the physical device. Even with such models, many radio

subsystems use exotic processor architectures lacking fully functional public instruction set emulators.

Space communication subsystems also employ protocols that diverge from commercial standards. As a result, firmware operates on lower-level data than typical Internet of Things (IoT) systems, processing symbol streams and reconstructing frame boundaries or sequencing. Thus, re-hosting fuzzers struggle to reach deeper protocol layers without significant manual effort. Consequently, most security testing must be conducted on flight-representative hardware, which is costly and time-consuming.

No Separation of Data and Commands Between Systems.

Across all evaluated systems, there is no explicit separation between user data transported by the COM and management commands that bypass the nominal CDHS path and directly address the COM subsystem. The absence of authentication per purpose implies that any entity capable of injecting traffic on the radio link, including payloads unrelated to platform control, can issue commands to the transceiver and potentially seize control. This is reflected in the data flow diagrams, as none of the COMs required data flows to the Internal Command Interface from an internal interface to pass authentication.

Missing Secure Boot Features. Of the three COMs evaluated, two provide no mechanism that assures software integrity at boot time. Only the SatNOGS-COMMS enforces secure boot through cryptographic signatures.

Non-Defensive Use of Cryptographic Libraries. During analysis, we recognized that the SatNOGS-COMMS and NanoCom AX100 are not following the defensive usage of crypto-related code. They are using non-constant-time comparisons. The SatNOGS-COMMS is discarding error values of crypto functions, just assuming that operations were successful. In the space environment, this is especially critical, as high-energy particle-induced faults may leave programs in an invalid state, making defensive error checking especially important.

8. Security Hardening Measures

Moving satellite components in the direction of zero trust is highly important, moving on from architecting satellites to act like a homogeneous system from a security standpoint to systems no longer trusting each other. As a first step in this direction, signatures may be checked on the device acting on the command instead of a device that is forwarding commands. For example, an Orbit Control System could feature authenticated thruster commands that are only valid once or when executed in a certain time period. This could prevent malware on other components from masquerading as a CDHS and having the satellite perform an orbital maneuver.

Introduction of Privilege Separation. One of the issues that all the systems have is that they lack a clear distinction

between data that the systems transport and commands that bypass the normal CDHS and can be issued directly to the COM instead. Introduction of authenticated internal commands, as shown in Figure 3, is vital.

Moving Away From PC/104. By the very working principle of the PC/104 connector, it makes sense to have many subsystems and components sharing the same digital interfaces for communication. The issue is that the components for space systems are usually supported for extended time periods and thus only adopt attack prevention measures slowly. Especially, as for example, new CAN transceivers that prevent attacks will have to be evaluated for successful operation in conditions that they have not been designed for. We suggest a novel bus interface based on optical communication, which maintains the advantages of a shared medium while enabling the isolation of individual nodes by wavelength. Additionally, this approach provides galvanic isolation [68].

Recovery Methods for On-board Software. In typical systems, compromises can often be resolved by redeploying software to restore a known secure state. This approach is not viable for satellites, as even fallback software cannot be assumed to be trustworthy. Therefore, a supervisory system is needed that can be attested as not compromised and capable of restoring the satellite’s state. Such a system introduces challenges, as it must not increase failure risk and must, by design, be able to override core functionality.

Adapting Best Practices from Other Domains. A major issue in COMs is the software monoculture in space. In the early Internet, many publicly accessible systems ran identical software and were frequently exploited. A similar problem is likely for satellites. Experience from the Internet shows that rapid patching and defense-in-depth mechanisms are effective. However, patches are rarely deployed in space hardware due to safety concerns, so mechanisms are needed to apply them safely. Techniques that guarantee patch properties, such as [69], could shift COTS vendor practices and reduce the exposure time of known vulnerabilities. Furthermore, as shown in Section 5, even basic mitigations like stack canaries are not widely used in space components. Vendors should reassess whether the risk of operational issues from such mitigations outweighs the security benefits. The authors of [31] also outline a path toward a more secure and resilient software stack.

Restrictions Posed by the Space Environment. Because mission failure is extremely costly, operators may hesitate to deploy security mechanisms that increase power use, add complexity, or introduce new failure points. This is especially true for mitigations requiring additional system states or computation. In such cases, watchdog timers can disable security features in a controlled way after access is lost due to an attack or failure. This allows operators to decide whether to passivate after losing a security function or continue operations without it and accept the risk.

More generally, security mechanisms should support phased deployment, enabling rollout to part of a system before extending protection to all components. As in conventional supervisory systems, security features also need backups or multiple ways to achieve the same goal, though such redundancy must be carefully hardened.

Finally, proposed mitigations must be evaluated against space environment constraints, including resilience to radiation-induced single-event upsets (SEUs) and potential impacts on component lifetime due to increased processing or resource use.

9. Discussion

Other Affected COMs. Upon disclosure, the affected vendors confirmed to us that other COMs in their portfolio are also affected by some of the vulnerabilities. This means that the total number of impacted missions probably significantly overshadows assumptions based on the use of specific hardware.

9.1. Limitations

Open Source Telemetry Survey. When analyzing the impact of SATBLEED in Section 6, we identify which satellite missions use HMACs in their beacons. Due to the lack of disclosure from mission operators, we rely on indirect indicators, namely the entropy of suspected HMAC fields and the frequency of HMAC flags in the payloads. To avoid false positives caused by bit errors in the flags field, we also confirm the presence of the HMAC field in the header by examining the entropy of that field. Note again that this approach does not work if the mission employs encryption/compression, as the payload would also have high entropy, making it impossible to detect the HMAC field directly before the payload. Therefore, we deliberately exclude missions employing encrypted/compressed telemetry. Importantly, neither mitigates SATBLEED but both obscure the presence of integrity protection. As a result, this exclusion systematically omits missions that may still be vulnerable, ensuring that our findings constitute a strict lower bound on the true impact of SATBLEED.

Representativeness of COM Samples. Although we analyzed major providers with a significant market share, the NewSpace ecosystem is highly diverse. Mission architectures, protocol stacks, and deployment models vary widely across systems. While the subcomponent-based perspective and the resulting threat taxonomy are designed to be generalizable, the concrete issues and vulnerabilities we uncovered are not necessarily so, since some arise from provider-specific implementation choices. Our findings therefore indicate that security oversights are common, but they do not imply that every system is likely vulnerable.

Completeness of Methodology. Our vulnerability discovery process relied on manual reverse engineering, physical hardware inspection, and targeted fuzzing. We do not prove the

absence of further flaws within these devices. Given the lack of transparency in the firmware, it is likely that additional vulnerabilities exist within the analyzed COMs.

10. Conclusion

In this paper, we develop the first comprehensive threat model for COMs and apply it to an in-depth analysis of communications subsystems from three major vendors. Our evaluation uncovers multiple critical vulnerabilities, some of which, when traced through open-source SatNOGS telemetry, reveal 28 vulnerable in-orbit active satellites. Beyond this, our broader testing uncovered systemic weaknesses, ranging from insufficient replay defenses to insecure update mechanisms that collectively expose the unprotected nature of current COM modules. Overall, our work highlights the urgent need to embed security throughout the design lifecycle of COM subsystems. Without such an approach, small satellites will remain vulnerable, undermining their operational reliability and hindering the advancement of resilient space assets.

11. Ethics considerations

Responsible Disclosure. We adhere to responsible disclosure principles and have responsibly disclosed all identified vulnerabilities to their vendors. GomSpace and Endurosat have both acknowledged the reported vulnerabilities and plan to address them in the next generation of their communication modules. GomSpace additionally stated that fixes are also planned for some of their currently available models. Furthermore, we are engaged in discussions with federal institutions to address the vulnerabilities and, in very critical cases, notify the affected missions directly.

Non-Attribution of Vulnerable Missions. Identifying active spacecraft that operate on these vulnerable COM systems, enables malicious actors to target them using the information in our publication. To mitigate this risk, we only present summarized results and statistical analysis, without linking vulnerabilities to individual mission names or operators. Although it is relatively trivial to derive the identity of the missions based on public data, we have decided not to publish the list of the 28 vulnerable missions.

Withholding of Exploit Payloads and Proof-of-Concept Code. No working exploits, payload generators are released as part of this publication. While technical details are described to a certain level, supporting academic reproducibility, we intentionally limit them to prevent immediate weaponization of our work.

Avoiding Harm During Analysis. All experimental evaluations were performed on isolated test setups or on vendor-provided hardware in a controlled environment. We not only avoided transmitting on RF bands, but also took strict precautions to ensure there was no RF leakage from our test setups. Additionally, we did not interact with the communication links of any operational satellite.

Ethical Limitations. Despite our best efforts, we are aware that any publication addressing vulnerabilities in widely used critical infrastructure may carry a certain risk. Attackers with sufficient motivation and technical expertise may be able to replicate or infer parts of our work. However, the benefits of disclosure, vendor response, and public awareness outweigh these risks.

Acknowledgments

The project on which this report is based was supported with funding from the German Federal Ministry of Research, Technology and Space (BMFTR) under the grant number 50YB2607B. The author is responsible for the content of this publication. We further thank the Saarbrücken Graduate School of Computer Science for their funding and support.

12. LLM usage considerations

LLMs were used for editorial purposes in this manuscript, and all outputs were inspected by the authors to ensure accuracy and originality.

References

- [1] M. Manulis, C. P. Bridges, R. Harrison, V. Sekar, and A. Davis, "Cyber security in new space: Analysis of threats, key enabling technologies and challenges," *International Journal of Information Security*, vol. 20, no. 3, pp. 287–311, 2021.
- [2] N. Boschetti, N. G. Gordon, and G. Falco, "Space cybersecurity lessons learned from the viasat cyberattack," in *ASCEND 2022*, 2022, p. 4380.
- [3] L. Saalman, L. Saveleva Dovgal, and F. Su, "Mapping cyber-related missile and satellite incidents and confidence-building measures," Stockholm International Peace Research Institute (SIPRI), SIPRI Insights on Peace and Security No. 2023/10, Nov. 2023, accessed May 27, 2026. [Online]. Available: https://www.sipri.org/sites/default/files/2023-11/2023_10_cyber_mapping_incidents.pdf
- [4] Symantec Threat Hunter Team, "Thrip: Espionage group hits satellite, telecoms, and defense companies," <https://www.security.com/threat-intelligence/thrip-hits-satellite-telecoms-defense-targets>, Jun. 2018, threat Intelligence blog post; 5min read.
- [5] M. P. Flaherty, J. Samenow, and L. Rein, "Chinese hack u.s. weather systems, satellite network," *The Washington Post*, accessed on May 27, 2026.
- [6] J. Willbold, M. Schloegel, M. Vögele, M. Gerhardt, T. Holz, and A. Abbasi, "Space odyssey: An experimental software security analysis of satellites," in *IEEE Symposium on Security and Privacy (SP)*, 2023, pp. 1–19.
- [7] T. Scharnowski, F. Buchmann, S. Wörner, and T. Holz, "A case study on fuzzing satellite firmware," in *Workshop on the Security of Space and Satellite Systems (SpaceSec)*, 2023.
- [8] R. Peled, E. Aizikovich, E. Habler, Y. Elovici, and A. Shabtai, "Evaluating the security of satellite systems," *arXiv preprint arXiv:2312.01330*, 2023.
- [9] G. Falco, "When satellites attack: Satellite-to-satellite cyber attack, defense and resilience," in *Proceedings of the AIAA ASCEND Conference: Accelerating Space Commerce, Exploration, and New Discovery*. American Institute of Aeronautics and Astronautics (AIAA), 2020, p. 4014.

- [10] J. Pavur and I. Martinovic, "The cyber-asat: On the impact of cyber weapons in outer space," in *International Conference on Cyber Conflict (CyCon)*, vol. 900, 2019, pp. 1–18.
- [11] F. Rawlins, R. Baker, and I. Martinovic, "Death by a thousand cots: Disrupting satellite communications using low earth orbit constellations," in *Workshop on the Security of Space and Satellite Systems (SpaceSec)*, 2023.
- [12] J. Pavur, D. Moser, M. Strohmeier, V. Lenders, and I. Martinovic, "A tale of sea and sky on the security of maritime vsat communications," in *IEEE Symposium on Security and Privacy (SP)*, 2020, pp. 1384–1400.
- [13] J. Willbold, M. Schloegel, R. Bisping, M. Strohmeier, T. Holz, and V. Lenders, "Vsaster: Uncovering inherent security issues in current vsat system practices," in *ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2024, pp. 288–299.
- [14] Libre Space Foundation / SatNOGS Project, "Satnogs – satellite networked open ground station," <https://www.satnogs.org/>, 2025, open-source global network of satellite ground stations. Initial release April 2014.
- [15] "How many satellites are in space?" <https://nanoavionics.com/blog/how-many-satellites-are-in-space/>, 2025, [Accessed 26-05-2025].
- [16] E. Kulu, "Nanosats database," 2025, [Accessed 30-04-2025]. [Online]. Available: <https://www.nanosats.eu/#figures>
- [17] "tinygs-webapp," <https://tinygs.com/>, [Accessed 21-07-2025].
- [18] Amazon Web Services, "Freertos - market leading rtos (real time operating system) for embedded systems with internet of things extensions," Dec. 2023. [Online]. Available: <https://www.freertos.org/index.html>
- [19] Zephyr Project, Linux Foundation, "Zephyr project – scalable real-time operating system," <https://www.zephyrproject.org/>, 2025, open-source RTOS hosted by the Linux Foundation. Release v4.1.0 (March 7, 2025); supports multi-architecture embedded systems.
- [20] RTEMS Project, OAR Corporation, "Rtems – real-time executive for multiprocessor systems," <https://www.rtems.org/>, 2025, open-source real-time operating system for embedded and space applications; initial release 1993; latest stable release 6.1 on 2025-01-22.
- [21] Alén Space, "TREVO: Modular High Performance SDR Platform — Alén Space — alen.space," <https://alen.space/products/trevo/>, highly flexible Software Defined Radio (SDR) with flight heritage.
- [22] J. Rederlechner, U. Planta, and A. Abbasi, "One small patch for a file, one giant leap for ota updates," 2026.
- [23] D. McComas, J. Wilmot, and A. Cudmore, "The core flight system (cfs) community: Providing low cost solutions for small spacecraft," in *AIAA/USU Conference on Small Satellites*. Logan, UT: Utah State University, University Libraries, 2016.
- [24] R. Bocchino, T. Canham, G. Watney, L. Reder, and J. Levison, "F prime: An open-source framework for small-scale flight software systems," in *AIAA/USU Conference on Small Satellites*, Logan, UT, 2018. [Online]. Available: <https://digitalcommons.usu.edu/smallsat/2018/all2018/328/>
- [25] P. W. José Manual Diez, Fabian Krech, "Raccoon os," <https://gitlab.com/rccn>, [Accessed 20-01-2025].
- [26] L. J. Kamm, "Magnetorquer-a satellite orientation device," *ARS Journal*, vol. 31, pp. 813–815, 1961. [Online]. Available: <https://api.semanticscholar.org/CorpusID:122456319>
- [27] European Space Agency, "Telemetry & Telecommand – Onboard Computers & Data Handling," https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Onboard_Computers_and_Data_Handling/Telemetry_Telecommand, accessed: 2025-07-15.
- [28] P. Tedeschi, S. Sciancalepore, and R. Di Pietro, "Satellite-based communications security: A survey of threats, solutions, and research challenges," *Computer Networks*, vol. 216, p. 109246, 2022.
- [29] O. Driouch, S. Bah, and Z. Guennoun, "Distributed intrusion detection system for cubesats, based on deep learning packets classification model," in *2024 Security for Space Systems (3S)*, 2024, pp. 1–8.
- [30] D. He, X. Li, S. Chan, J. Gao, and M. Guizani, "Security analysis of a space-based wireless network," *IEEE Network*, vol. 33, no. 1, pp. 36–43, 2019.
- [31] S. Jero, J. Furgala, M. A. Heller, B. Nahill, S. Mergendahl, and R. Skowyra, "Securing the satellite software stack," in *Workshop on the Security of Space and Satellite Systems (SpaceSec)*, 2024.
- [32] D. Maurice-Michel, "How I hacked an ESA's experimental satellite — deadf00d.com," <https://www.deadf00d.com/post/how-to-hack-an-esa-experimental-satellite.html>, [Accessed 09-07-2025].
- [33] J. Smailes, S. Köhler, S. Birnbach, M. Strohmeier, and I. Martinovic, "Watch this space: Securing satellite communication through resilient transmitter fingerprinting," in *ACM Conference on Computer and Communications Security (CCS)*, ser. CCS '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 608–621. [Online]. Available: <https://doi.org/10.1145/3576915.3623135>
- [34] J. Wigchert, S. Sciancalepore, and G. Oligeri, "Detection of aerial spoofing attacks to leo satellite systems via deep learning," *Computer Networks*, p. 111408, 2025.
- [35] L. Masson, M. Bonjour, L. Thoeny, and S. Willy, "Developing a ccscs compliant platform to reliably secure current and future space data links," in *2024 Security for Space Systems (3S)*, 2024, pp. 1–8.
- [36] J. Willbold, F. Sciberras, M. Strohmeier, and V. Lenders, "Satellite cybersecurity reconnaissance: Strategies and their real-world evaluation," in *IEEE Aerospace Conference*. IEEE, 2024, pp. 1–13.
- [37] J. Pavur, D. Moser, V. Lenders, and I. Martinovic, "Secrets in the sky: on privacy and infrastructure security in dvb-s satellite broadband," in *ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2019, pp. 277–284.
- [38] J. Pavur, M. Strohmeier, V. Lenders, and I. Martinovic, "Qpep: An actionable approach to secure and performant broadband from geostationary orbit," *Proceedings 2021 Network and Distributed System Security Symposium*, 2021. [Online]. Available: <https://api.semanticscholar.org/CorpusID:231879176>
- [39] "Yamcs v5.8.6 Vulnerability Assessment — visionspace.com," <https://visionspace.com/yamcs-v5-8-6-vulnerability-assessment/>, [Accessed 28-07-2025].
- [40] E. López-Morales, U. Planta, G. Marra, C. González, J. Hopkins, M. Garosi, E. Obreque, C. Rubio-Medrano, and A. Abbasi, "Honeysat: A network-based satellite honeypot framework," in *Proceedings of the NDSS Symposium*, ser. NDSS '26. The Internet Society, 2026. [Online]. Available: <https://www.ndss-symposium.org/ndss-paper/honeysat-a-network-based-satellite-honeypot-framework/>
- [41] E. Salkield, S. Köhler, S. Birnbach, R. Baker, M. Strohmeier, and I. Martinovic, "Firefly: Spoofing earth observation satellite data through radio overshadowing," in *Workshop on the Security of Space and Satellite Systems (SpaceSec)*, 2023.
- [42] R. Bisping, J. Willbold, M. Strohmeier, and V. Lenders, "Wireless signal injection attacks on VSAT satellite modems," in *USENIX Security Symposium*. Philadelphia, PA: USENIX Association, Aug. 2024, pp. 6075–6091. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity24/presentation/bisping>
- [43] T. Saeidi and S. Karamzadeh, "Enhancing cubesat communication through beam-steering antennas: A review of technologies and challenges," *Electronics*, vol. 14, no. 4, p. 754, 2025.
- [44] AAC Clyde Space, "Pulsar-utrux uhf/uhf transceiver – high-performance cubesat telemetry and command radio," https://www.aac-clyde.space/wp-content/uploads/2021/10/AAC_DataSheet_Pulsar_UTRX-4.pdf, 2023, compact PC/104-form UHF telemetry and command radio, 9600 bps GMSK / 1200 bps AFSK, 27–33 dBm TX power, SEU-immune flash FPGA; release date June 6 2023.

- [45] C. S. N. Aeronautics and S. Administration, “SPACE DATA LINK SECURITY PROTOCOL,” <https://ccsds.org/Pubs/355x0b2.pdf>, [Accessed 23-06-2025].
- [46] U. Planta, J. Rederlechner, G. Marra, and A. Abbasi, “Let me do it for you: On the feasibility of inter-satellite friendly jamming,” in *2024 Security for Space Systems (35)*. IEEE, 2024, pp. 1–6.
- [47] PolySat Team, “Earth Station – PolySat,” <https://www.polysat.org/earth-station>, 2025, accessed: 2025-07-03.
- [48] “BRO (Breizh Reconnaissance Orbiter) / Unseenlabs - eoPortal — eoportal.org,” <https://www.eoportal.org/satellite-missions/unseenlabs#references>, [Accessed 22-07-2025].
- [49] “Kleos Commercial RF Monitoring CubeSat Constellation - eoPortal — eoportal.org,” <https://www.eoportal.org/satellite-missions/kleos#mission-status>, [Accessed 22-07-2025].
- [50] “SpooQy-1 CubeSat Mission - eoPortal — eoportal.org,” <https://www.eoportal.org/satellite-missions/spooqy-1#spooqy-1-cubesat-mission-with-qkd-quantum-key-distribution>, [Accessed 22-07-2025].
- [51] E. Kulu, “ERNST - Nanosats Database — nanosats.eu,” <https://www.nanosats.eu/sat/ernst>, [Accessed 22-07-2025].
- [52] “ION-SCV space.skyrocket.de,” https://space.skyrocket.de/doc_sdat/ion-scv-2.htm, [Accessed 22-07-2025].
- [53] “CubeSat Companies - Top Companies List of CubeSat Industry — marketsandmarkets.com,” <https://www.marketsandmarkets.com/ResearchInsight/cubesat-market.asp#:~:text=EnduroSat%20,solar%20panels%2C%20and%20custom%20modules>, [Accessed 12-11-2025].
- [54] “CubeSat Market Size, Share & Trends — Industry Report, 2033 — grandviewresearch.com,” <https://www.grandviewresearch.com/industry-analysis/cubesat-market-report#:~:text=Key%20companies%20profiled>, [Accessed 12-11-2025].
- [55] Lucintel, “CubeSat Market Report: Trends, Forecast and Competitive Analysis to 2030 — lucintel.com,” <https://www.lucintel.com/cubesat-market.aspx>, [Accessed 13-11-2025].
- [56] F. B. Insights, “CubeSat Market Size, Share — Global Growth Report [2032] — fortunebusinessinsights.com,” <https://www.fortunebusinessinsights.com/cubesat-market-113707>, 2025, [Accessed 13-11-2025].
- [57] I. Group, “CubeSat Market Size, Analysis, Trends & Forecast, 2033 — imarcgroup.com,” <https://www.imarcgroup.com/cubesat-market>, 2024, [Accessed 13-11-2025].
- [58] EnduroSat, “EnduroSat – Satellites & Space Missions,” <https://www.endurosat.com/>, accessed: 2025-07-15.
- [59] “SharedS space.skyrocket.de,” https://space.skyrocket.de/doc_sdat/spartan.htm, [Accessed 22-07-2025].
- [60] “curiumsat · GitLab — gitlab.com,” <https://gitlab.com/curiumsat>, [Accessed 22-07-2025].
- [61] “FOSDEM 2025 - SatNOGS-COMMS: An Open-Source Communication Subsystem for CubeSats — fosdem.org,” <https://fosdem.org/2025/schedule/event/fosdem-2025-6024-satnogs-comms-an-open-source-communication-subsystem-for-cubesats/>, [Accessed 28-07-2025].
- [62] “FreeRTOS Release History - FreeRTOS — freertos.org,” <https://freertos.org/Documentation/04-Roadmap-and-release-note/02-Release-notes/00-Release-history>, [Accessed 22-07-2025].
- [63] Y. Shoji, “libcsp/libcsp,” Feb. 2024. [Online]. Available: <https://github.com/libcsp/libcsp>
- [64] “MAC comparison leaks timing data · Issue #44 · libcsp/libcsp — github.com,” <https://github.com/libcsp/libcsp/issues/44>, [Accessed 22-07-2025].
- [65] G. Camurati, S. Poeplau, M. Muench, T. Hayes, and A. Francillon, “Screaming channels: When electromagnetic side channels meet radio transceivers,” in *ACM Conference on Computer and Communications Security (CCS)*, 2018, pp. 163–177.
- [66] “GOMspace — NanoCom Link SX — gomspace.com,” <https://gomspace.com/shop/subsystems/communication-systems/nanocom-link-sx.aspx>, [Accessed 22-07-2025].
- [67] Libre Space Foundation / SatNOGS Project, “Satnogs comms protobuf messaging specification,” <https://gitlab.com/librespacefoundation/satnogs-comms/satnogs-comms-protobuf>, 2023, open-source protobuf definition for SatNOGS communications subsystem; created June 19, 2023.
- [68] B. Palmer, V. Schulz, M. Jahnke, E. Stoll, and U. Kulau, “Wireless intra-satellite lifi dual can bus networks for redundancy and throughput,” in *Deutsche Luft-und Raumfahrtkongress 2025*, 2025.
- [69] Y. He, Z. Zou, K. Sun, Z. Liu, K. Xu, Q. Wang, C. Shen, Z. Wang, and Q. Li, “RapidPatch: Firmware hotpatching for Real-Time embedded devices,” in *USENIX Security Symposium*. Boston, MA: USENIX Association, Aug. 2022, pp. 2225–2242. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity22/presentation/he-yi>

Appendix A. SatNOGS Survey Details

As mission operators rarely disclose details about their use of HMACs or encryption, we approximate this information based on Telemetry data recorded and released by the SatNOGS project [14]. Our approach filters the SatNOGS database for missions using our target transmitter. For each selected mission, we downloaded up to 10,000 telemetry samples. This limit was chosen to balance between obtaining sufficient data for meaningful analysis and minimizing the load on the SatNOGS API, especially since some missions have accumulated several million samples. The downloaded data samples have undergone demodulation but not yet decoding. Consequently, we performed decoding ourselves based on the CSPv2 standard, which is used by the relevant transmitter.

Given our interest regarding the usage of HMACs within a mission, the most straightforward approach for processing these samples would be to examine the flags that have been set. However, our investigations have indicated that these flags are not always utilized correctly. Therefore, we have decided to examine the actual HMACs to obtain results that are more aligned with the truth.

The processing pipeline is as follows:

- 1) **Filtering Duplicate Samples:** Identical telemetry samples do not yield new insights about HMAC behavior, as their HMACs (if present) would remain unchanged. These duplicates are removed.
- 2) **Filtering Incomplete Samples:** We remove samples that are smaller than the CSP header size or have payloads too short to contain a truncated HMAC.
- 3) **Filtering Encrypted Samples:** We exclude all samples with the encryption flag set, regardless of the HMAC flag status, since encrypted payloads prevent meaningful inspection of the HMAC.
- 4) **Entropy Estimation via Hamming Distance:** Direct entropy analysis on 4-byte segments is infeasible due to the small size, so we approximate entropy using the Hamming distance between samples. While this method cannot distinguish whether a high Hamming distance is caused by high variance in the data or by

the presence of an HMAC, it is reasonable to assume that the rarity of a large variance in the data transmitted will yield a limited number of outliers. Conversely, the implementation of an HMAC will result in consistent, substantially large Hamming distances in comparison.

We then categorized missions based on the following criteria:

- 1) **Not enough Data:** Missions with fewer than 20 usable samples are marked as having insufficient data for analysis.
- 2) **Encryption Only:** If all but fewer than 5 samples have the encryption flag set, including the samples filtered for encryption earlier, we assume the dataset available is limited to only containing encrypted samples, with the exceptions potentially caused by bit flips or noise.
- 3) **No HMAC:** A mission is tagged as not using HMACs if fewer than 5 usable samples have the HMAC flag set and the average Hamming distance across the presumed HMAC bytes is below 14. This threshold is chosen because the expected Hamming distance for truly random 4-byte sequences is 16; a value significantly below that suggests low entropy consistent with the absence of an HMAC.

For all remaining missions that likely use HMACs, we identify three possible scenarios:

- 1) **Consistent and Plausible HMAC usage:** The HMAC flag frequency aligns with the frequency of high Hamming distances in the presumed HMAC region, suggesting correct and consistent use of HMACs. For these missions, we further analyze whether they excursively use HMACs, or if a significant number of samples don't use HMACs. The latter suggests that the mission switches between using an HMAC and not using an HMAC.
- 2) **Underused HMAC Flag:** If the HMAC flag count is significantly lower than the observed entropy, this may imply inconsistent flag usage or a large variation in the payload data that comes close to mimicking the randomness of an HMAC.
- 3) **Overused HMAC Flag:** If the HMAC flag count is significantly higher than what the entropy suggests, this may indicate that the flag is used incorrectly or repurposed.

For the latter two scenarios, we cannot confidently determine the true HMAC usage.

Appendix B. Additional Data/Attack Flows

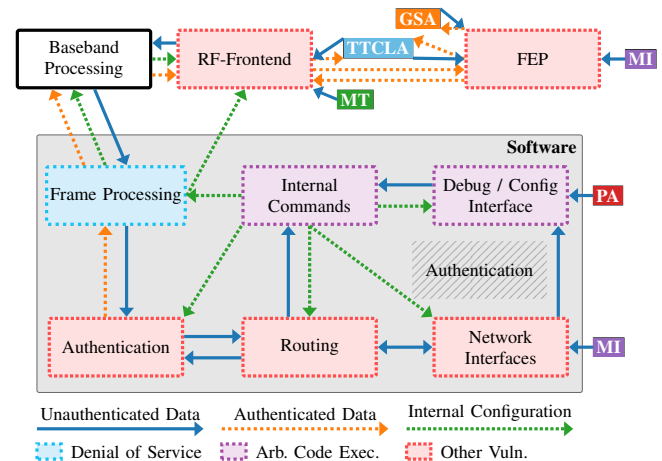
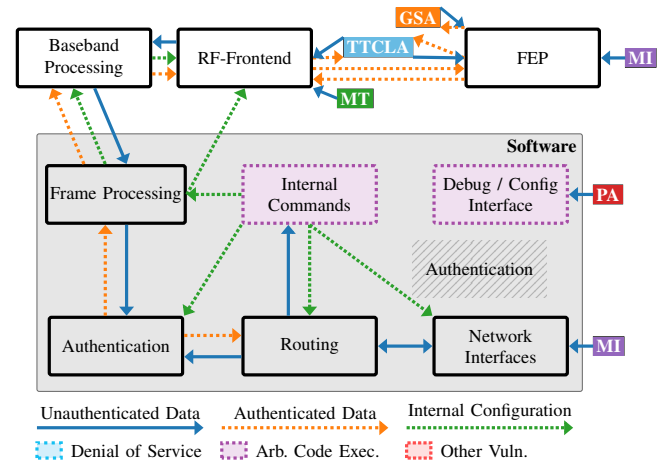


Figure 6. Sub-component based view of data/attack flow for the GomSpace NanoCom AX100.



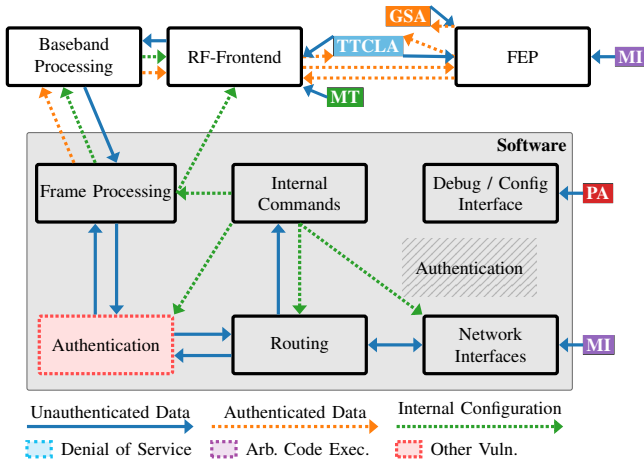


Figure 7. Sub-component based view of data/attack flow for the Libre Space Foundation SatNOGS-COMMS. Note that the MCU debug interface being open is at digression of the mission owners and thus cannot be counted as a flaw of the COM product

Appendix C. RF-Measurements of AX100

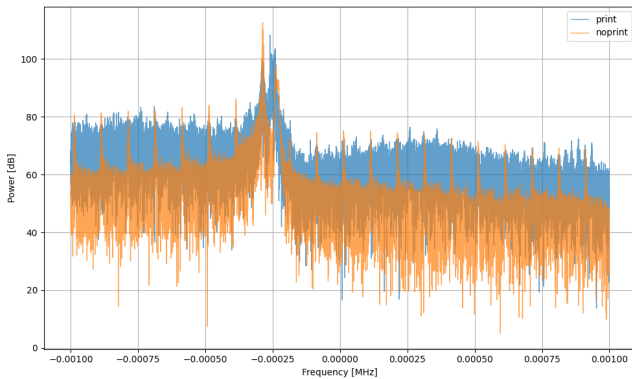


Figure 8. Measured RF spectrum of the transceiver while the MCU executes different workloads.

Appendix D. Meta-Review

The following meta-review was prepared by the program committee for the 2026 IEEE Symposium on Security and Privacy (S&P) as part of the review process as detailed in the call for papers.

D.1. Summary

This paper analyzes the security of commercial off-the-shelf (COTS) satellite communication (COM) subsystems used in small satellites. The authors examine representative COM modules from several vendors using reverse engineering and testing techniques, identify vulnerabilities in firmware and system design, and discuss potential implications for deployed satellite missions. The work aims to connect weaknesses in real-world implementations with practical attack scenarios against satellite communication systems.

D.2. Scientific Contributions

- Identifies an Impactful Vulnerability.
- Provides a Valuable Step Forward in an Established Field.
- Establishes a New Research Direction.

D.3. Reasons for Acceptance

- 1) The paper presents an empirical security analysis of widely used COTS satellite communication modules and identifies vulnerabilities in real implementations.
- 2) The work highlights satellite COM modules as an important security boundary and reveals weaknesses in widely reused components that may affect multiple deployments.