VSAsTer: Uncovering Inherent Security Issues in Current VSAT System Practices

Johannes Willbold
Ruhr University Bochum
Bochum, Germany
johannes.willbold@rub.de

Moritz Schloegel
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany
moritz.schloegel@cispa.de

Robin Bisping
ETH Zürich
Zurich, Switzerland
bispingr@student.ethz.ch

Martin Strohmeier
Cyber-Defence Campus, armasuisse Science and Technology
Thun, Switzerland
martin.strohmeier@armasuisse.ch

Thorsten Holz
CISPA Helmholtz Center for Information Security
Saarbrücken, Germany
holz@cispa.de

Vincent Lenders
Cyber-Defence Campus, armasuisse Science and Technology
Thun, Switzerland
vincent.lenders@armasuisse.ch

ABSTRACT

Recent geopolitical events have exposed our critical dependence on the wireless infrastructure used to facilitate worldwide communication. State-sponsored groups are actively attacking and exploiting space-based communication networks, causing outages and serious economic damage. Despite initial research findings pointing out a lack of security, such networks enjoy growing adoption and are still placed at the heart of today's communication infrastructure, ranging form the transportation sector over oil rigs to consumer internet. Worryingly, the command and control networks that support this satellite-based communication have received little attention from the security community so far.

This paper addresses this research gap and conducts a systematic security assessment of the *Very Small Aperture Terminal* (VSAT) ecosystem. More specifically, we investigate the attack surface of the underlying command and control networks and analyze the systems currently used by industry-leading vendors. Through systematic reverse engineering, we uncover a number of wide-reaching vulnerabilities that illustrate the perilous position of the satellite industry. We then systematically formulate a phase-based threat model to categorize these issues and uncover several inherently insecure design practices.

CCS CONCEPTS

• **Security and privacy → Systems security**; *Domain-specific security and privacy architectures.*

KEYWORDS

vsat, satellites, service networks, security analysis, vulnerabilities

ACM Reference Format:
Johannes Willbold, Moritz Schloegel, Robin Bisping, Martin Strohmeier, Thorsten Holz, and Vincent Lenders. 2024. VSAsTer: Uncovering Inherent Security Issues in Current VSAT System Practices. In *Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '24), May 27–30, 2024, Seoul, Republic of Korea.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3643833.3656139

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
WiSec '24, May 27–30, 2024, Seoul, Republic of Korea
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0582-3/24/05.
https://doi.org/10.1145/3643833.3656139

1 INTRODUCTION

Today's world is hard to imagine without satellites. They provide a number of crucial functions, ranging from global navigation and positioning systems to phone connections, imaging data, and general-purpose data links. Recent conflicts, such as the Russian invasion of Ukraine, have further substantiated the value of satellite images for military purposes [47], and space-based communication complemented or replaced terrestrial systems [29]. One crucial part of this critical satellite infrastructure are Very Small Aperture Terminal (VSAT) systems, which are two-way communication systems whose ground stations use dishes smaller than 3.8 meters. VSAT systems transmit voice, data, and video over satellites in geostationary orbit. As a single satellite can cover large areas of the Earth, VSAT systems are mainly used in long-distance transportation, i. e., shipping and aviation, as well as very remote places. This makes them attractive targets for attackers, in particular nation-state actors targeting critical infrastructure.

Recently, two high-profile cases of such attacks have illustrated the impact in practice: The KA-SAT incident [49], also referred to as *ViaSat* incident, and the Dozor-Teleport incident. Both have taken place in the context of Russia's war against Ukraine. On the eve of the Russian invasion, 45,000 endpoints connected to Viasat's KA-SAT network were rendered inoperable by the *AcidRain* malware, not only in Ukraine but across Europe [27, 42, 49]. In June 2023, the Russian satellite communication provider Dozor-Teleport, who provides services to the Russian state and military, was knocked off the grid for 15 hours. While details are sparse, the provider blamed a breach of their cloud infrastructure, which enabled the unknown attackers to exfiltrate data and take control of the network [25, 50].

While these high-profile attacks show the criticality of VSAT networks, few technical details of the vulnerabilities exploited are publicly known. This fact is aggravated by a general lack of security research on VSAT networks. Existing research focuses on the easy-to-analyze *payload traffic*, i. e., the internet traffic passed through the VSAT network. This type of traffic is publicly documented and

comparably easy to capture and analyze, allowing to study the confidentiality and integrity of user data [35, 36]. However, the security of this payload has little to do with the security of the actual *VSAT network* itself. In particular, payload traffic and command and control traffic are separated. The latter is crucial to maintaining the VSAT network's security properties; undermining its integrity potentially risks the entire network and, thus, critical infrastructure. The *ViaSat* and Dozor-Teleport incidents have shown that malicious actors have an interest in these systems and have successfully identified critical vulnerabilities that can be exploited. What remains unknown is how difficult or easy it was to penetrate these systems or how many other security vulnerabilities exist. Without a public security analysis or documentation, it is challenging to assess the state of security of VSAT systems. While existing security frameworks, such as SPARTA [46] or ESA's SpaceShield [15], allow for retroactively modeling specific incidents once details are known, their generic nature makes the reverse process challenging: It is difficult to derive threats specific to one particular system from their universal overview. In particular, these frameworks fail to capture the intrinsic internals of VSAT networks, such as their emphasis on recoverability due to the geographical remoteness and varying volatility and persistence of configurations.

In this work, we address this research gap by systematically studying VSAT networks, in particular their command and control traffic, and the security properties of these systems. Based on our study, we derive a VSAT-specific *threat taxonomy* that enables a systematic security assessment of potential threats. In addition, we underpin our results by an *experimental security analysis* of two VSAT terminals, including one that has been targeted in a recent real-world attack. Our analysis shows that VSAT networks suffer from inherent critical security flaws that enable attackers to fully compromise them. Studying whether the attack vectors in our threat taxonomy translate into actual vulnerabilities, we find critical flaws in both analyzed VSAT terminals, demonstrating the dire state of VSAT network security. Based on the lessons learned, we discuss three inherently insecure VSAT network design practices and sketch how they could be addressed.

Contributions. In summary, our contributions are:

- We are the first to systematically analyze VSAT networks; in particular, we include the command and control traffic in our security assessment, which previous work has not considered.
- We derive a VSAT-specific threat taxonomy that allows us to systematically assess VSAT-internal threats, taking into account recoverability and varying configurations of such networks.
- We experimentally validate our taxonomy through an experimental security analysis of two VSAT systems and uncover critical vulnerabilities in both.

2 BACKGROUND

Before presenting our threat taxonomy, we provide a brief technical background on VSAT networks in general and discuss recent VSAT security incidents.

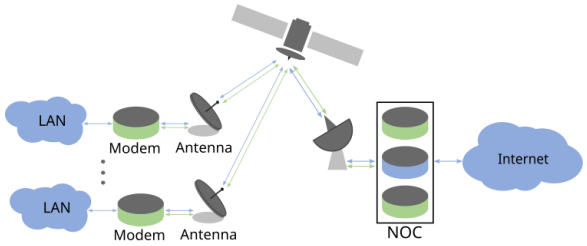


Figure 1: VSAT Network: The VSAT hub routes the user plane traffic from the individual remote networks to the internet.

2.1 VSAT Networks

VSAT networks are wireless networks using satellites to distribute network traffic over long distances into (potentially remote) areas. In these networks, an antenna sends the network signal to a satellite, which acts as a *bent pipe* and redirects the traffic to a different destination on Earth [19] (see Figure 1). Arriving at the destination, the signal is picked up by another antenna, facilitating communication between the two (or more) antennas via the satellite [8]. In a common VSAT network, a *central hub* acts as a gateway to the internet for the entire network. Each VSAT endpoint in the network communicates with the gateway via a satellite (or a larger constellation) using *uplink* (ground-to-satellite) and *downlink* (satellite-to-ground) traffic streams. The data stream from the hub to the endpoint is thereby called *forward link* and includes an uplink to the satellite followed by a downlink to the *endpoint*. In contrast, the *return link* is the traffic flow from the endpoint via an up- and downlink to the VSAT hub. We now describe VSAT network components in more detail.

VSAT Endpoint. A VSAT *endpoint* connects a Local Area Network (LAN) to the VSAT network. The endpoint uses an *antenna* with a radio transmitter and receiver. The receiver is connected via a cable to a modem device, which handles the signal processing through (de)-modulation, DVB-S de/encoding, and error correction. After the signal processing, the traffic is passed on to the network management, which performs the protocol handling as usually seen in network modems and routers. As such, this part of the VSAT endpoint is also referred to as the *VSAT modem*, which handles the network and endpoint management protocols and often acts as a router for the local network.

VSAT Hub. The *VSAT hub* connects all VSAT endpoints in the VSAT network to the internet. Like endpoints, hubs consist of an antenna and network management equipment. The dimensions are several magnitudes larger than in an endpoint, as the signal strength and traffic amount of the entire network have to be processed. Due to the scale and complexity of the radio and network equipment, there is usually dedicated staff with specialized domain knowledge to operate the hub. Additionally, the hub deploys all services required to manage and configure the network and its endpoints.

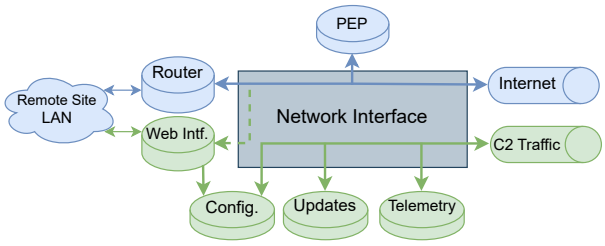


Figure 2: VSAT Endpoint Software: The network interface separates internet and C2 traffic.

2.1.1 Network Traffic. VSAT network traffic is divided into *user plane traffic*, e.g., *internet traffic* and *control plane traffic*, i.e., *command-and-control (C2) traffic*. Figure 2 highlights a VSAT endpoint's common *software components*, which are split into an internet forwarding part (upper half) and a command-and-control network part (lower half). All network traffic is physically handled by the same hardware and interacts with the same VSAT network. Thus, a network interface separates internet and C2 traffic from the same physical link.

User Plane Traffic. The *user plane traffic* is the network's main payload and is forwarded at the hub, e. g., to the internet. Due to the substantial latency of Geostationary Orbit (GEO) satellites, hubs provide a Performance Enhancing Proxy (PEP), which pre-acknowledges Transmission Control Protocol (TCP) connections to prevent timeout errors in user applications on the endpoint's LAN [18, 37]. The internet traffic is often submitted using the *DVB-S2*, a digital broadcasting standard.

Control Plane Traffic. The hub uses the control plane, which forms the *C2 network*, to maintain, monitor, and configure the endpoints remotely. This network is separated from the user plane, which can either be implemented as logical separation using, e.g., a Virtual Private Network (VPN), or using an OSI layer 2 separation that utilizes separate DVB message structures. Additionally, satellite telemetry, tracking, and command (TT&C) also utilizes a control plane, which is again entirely separate from the VSAT control plane and not the subject of this paper. The control plane is used to deploy (persistent) changes, such as software updates, key exchanges, and configurations. Finally, each endpoint provides a telemetry service that supplies the *hub* with Quality of Service (QoS) and status information.

2.2 VSAT Security Incidents

Recently, three security incidents have illustrated grave security issues within VSAT systems. We briefly review these incidents to motivate the need for a threat taxonomy.

KA-SAT Incident. The most high-profile VSAT incident was allegedly conducted to support the Russian invasion of Ukraine on February 24, 2022. The attackers targeted the US satellite ISP ViaSat, concretely their KA-SAT network, which supports critical infrastructures and military applications [27].

The attack left the affected endpoints incapable of accessing the network. Although the exact number of affected devices was not disclosed, ViaSat reported that around 30,000 replacement endpoints had been shipped to distributors, and the European Union Agency for Cybersecurity (ENISA) estimated that the attack impacted at least 27,000 devices. Collateral damage included the outage of remote monitoring and control for over 5,800 wind turbines in Germany, which remained offline for several weeks [10].

ViaSat's own incident report only confirmed the attack and that the attackers executed "legitimate, targeted management commands on numerous residential modems simultaneously", enabling them to download the *AcidRain* wiper malware [49].

Further open-source investigations argued that a known Fortinet vulnerability played a decisive role, as VPN appliances by the company were used by Gateway Earth Stations, control centers, and the affected endpoints [6, 42]. From here, the attackers could move laterally via the satellite network to target the vulnerable endpoints and use the built-in update mechanism to deploy the malware.

Dozor-Teleport Incident. The second attack on satellite systems to receive global attention in the context of the war against Ukraine was an attack on the Russian satellite ISP Dozor-Teleport [28]. The details remain vague even several months later. However, it is established that the website and the Dozor-Teleport network went down around 02:00 on June 29, 2023, for about 15 hours. Full normal operations were only established over a week later, on July 7, 2023. The ISP has significant upstream connections and serves power lines, oil fields, Russian military units, Northern Fleet ships, a nuclear power plant, and the Russian Federal Security Service, making a targeted attack likely [50]. Dozor-Teleport cited cloud infrastructure as a potential attack vector, which caused the ISP's satellite terminals to fail and enabled the attackers to exfiltrate internal data [25].

Starlink DoS Attacks. As reported widely, the communication capabilities of the Starlink constellation have been playing a crucial role during the Ukraine war, enabling the Ukrainian army to communicate flexibly and effectively during front-line operations [13, 22]. This makes Starlink an obvious target, and Russia has reportedly been trying to disable it or at least reduce its reliability by jamming the Ukrainian Starlink terminals. The signal structure of the Starlink downlink has been reverse-engineered publicly [20].

Starlink has several properties that help defend against jamming and denial of service compared to legacy constellations. Besides a more effective software update capability, inherent system and hardware features such as the large number of satellites, the highly directional, comparatively small spot beams, and the ability to choose between the available satellites provide significant resilience and redundancy [13, 22, 39].

While the alleged jamming attacks fall on the side of traditional electronic warfare and are a practical cat-and-mouse game, they illustrate the exposed position of VSAT systems for both GEO and LEO constellations today.

3 VSAT THREAT TAXONOMY

Our goal is to systematically capture all software security threats, both those observed in recent incidents and potential ones relevant to VSAT systems and enumerate them in a taxonomy. To this end, we first discuss the goals an attacker may have w.r.t. VSAT systems, introduce the security goals, and then present a threat model that

accounts for the unique lifecycle phases of these systems and their attack surface.

3.1 Attacker Goals

In the first step, we identify three realistic attacker goals based on previous incidents and related work.

3.1.1 Denial of Service. Many recent real-world VSAT security incidents, all with geopolitical significance, had the goal of Denial of Service (DoS).

Endpoint. The *ViaSat incident* targeted the endpoints in the network: After compromising the *VSAT hub*, the attackers deployed malware to the endpoints that overwrote the flash memories and made it a *persistent DoS*. The overwritten flashes and broken recovery prompted on-site intervention, making the DoS so costly.

Hub. The service outage during the *Dozor-Teleport incident* indicates that attackers compromised at least parts of the *VSAT hub*. This shows that the VSAT hub, like in the *endpoint* DoS vector, is a potential target. Unlike the *endpoint* vector, hub services can generally be restored from a centralized place with technical experts already present. Since such attacks are usually only temporary, we classify *hub DoS attacks* as *temporary DoS*.

Link. As seen in the Starlink DoS attempts in Ukraine, attackers target the physical link of VSAT networks to disrupt operations.

3.1.2 Attacker-in-the-Middle. Pavur et al.'s research on maritime VSAT internet traffic [36] has proposed an attack where malicious actors would complete a pending TCP handshake before the legitimate hub could do so, hijacking a VSAT-established TCP connection as Attacker-in-the-Middle (AitM). In Section 4.2.3, we experimentally verify a similar VSAT *link hijack*.

3.1.3 Eavesdropping. Given that eavesdropping attacks can be purely passive, it is nearly impossible to verify that one has occurred. Regardless, research by Pavur et al. showed that large portions of VSAT internet traffic are unencrypted [35], making eavesdropping relatively simple from a technical standpoint. In addition, documents leaked by Snowden indicated that intelligence agencies have identified VSAT traffic as an interesting target and carry out related operations [30].

3.2 Security Goals

Based on the attacker goals and previous incidents, we formulate four primary security goals.

Recoverability. So far, whenever attackers achieve persistence on endpoints, the incident prompts intervention from maintainers to recover the assets. However, we argue that there must be a path to recover the compromised parts. For hubs, the available specialized personnel can carry out this task, but this is not the case for endpoints. Due to their (potentially very) remote location, endpoints should be able to recover from every software fault fully autonomously. Even if an endpoint's software is entirely wiped, there must be a procedure to re-establish the broken image and reconnect to the network *without* the physical intervention of a human (operator). Hence, we consider *recoverability* as a security goal, primarily with endpoints in mind. This represents a strong requirement; however, we think it is crucial due to the nature of a VSAT network and the remote location of endpoints.

Availability. Since the VSAT network is usually the only point of connection for remote installations, it is paramount that the network service is *available* at all times.

Integrity. In cases where attackers establish themselves as *AitM* on the *link*, e. g., as shown by Pavur et al. [35] and our experimental security analysis (cf. Section 4), it is crucial to mitigate network traffic tampering.

Confidentiality. Confidentiality is of special relevance for VSAT networks since VSAT traffic is often broadcast over large geographical areas, which allows attackers to intercept traffic without being located close to the target.

3.3 VSAT Lifecycle Phases

We now introduce a model to describe the *different lifecycle phases* of VSAT systems with a strong focus on *recoverability* to account for the system's remoteness and inaccessibility. The model allows us to describe different types of data persistence and volatility and how each data type can be restored after an incident. Therefore, we divide the operational time frames into five phases, where *phase* refers to a time frame from an endpoint's point of view. Initially, the endpoint is *(i) commissioned*. Then, its regular operation cycle begins: Upon every restart, the endpoint is first *(ii) initialized* before entering the *(iii) operational* state. At times, it may be subject to *(iv) maintenance*. During an incident, the endpoint may be compromised and require *(v) recovery*.

3.3.1 Commissioning Phase. Endpoints are first introduced into a VSAT network using a commissioning process to generate information required for a first connection. There, so-called *beamtables* are generated, which contain information on the satellite beam, frequency, and pointing. The endpoint also receives a certificate that uniquely identifies it. For example, this certificate is used in commercial networks to verify paying customers. The commissioning phase introduces configuration that usually never changes.

3.3.2 Initialization Phase. The initialization phase supplies the endpoint with *volatile and temporary configurations* that change somewhat frequently and can be re-requested by the endpoint, e. g., after a restart. This information includes shared keys, which can be re-exchanged, network addresses, and layout information supplied through a protocol such as Dynamic Host Configuration Protocol (DHCP). The hub also checks if an endpoint is still eligible, e. g., if a customer is still a valid client. The phase is mostly characterized by supplying information needed for regular operation.

3.3.3 Operational Phase. An endpoint spends the overwhelming majority of its lifecycle in the operational phase, where it carries out its designed duties. This phase features two different types of traffic belonging to *service operations* and *service control*.

Service Operations. Ultimately, endpoints operate to receive network traffic, such as internet traffic transmitted using DVB-S2. This network traffic is part of the network's service offered to the customers of the VSAT endpoint and what most research papers so far exclusively focused on [5, 35, 36, 45].

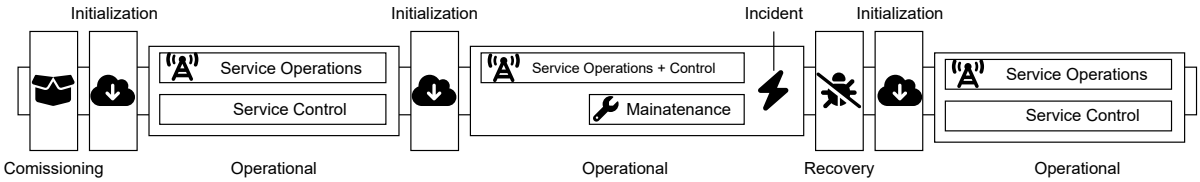


Figure 3: Exemplary VSAT Phases Timeline: The endpoint phases can be imagined on a timeline.

Service Control. To continuously provide service operations, the hub periodically sends service control information to the endpoints. Such information includes QoS monitoring, Adaptive Coding and Modulation (ACM), which adapts coding and modulation (e. g., to account for weather conditions), highly precise time synchronization, and multistream control information, used to divide service traffic into multiple traffic streams for different applications. This information is sent every few seconds or even several times per second often times in multi-casts.

3.3.4 Maintenance Phase. Software and firmware updates, critical service signaling, and persistent configuration are managed during maintenance. The phase usually performs lasting and persistent actions on the endpoint that can fundamentally change the operation of the endpoint and can only be changed by another maintenance phase or recovery phase.

3.3.5 Recovery Phase. The recovery phase is triggered automatically if an endpoint enters an invalid or non-connectable state, as shown in Figure 4. The phase should restore an endpoint to a connectable state *without requiring physical intervention*. This assumes that (1) the faulty phase identifies it is currently in a non-connectable state, (2) the faulty phase successfully transitions into the recovery phase, and (3) the recovery phase successfully recovers the endpoint. The recovery phase should be able to recover an endpoint from an attack, even if it has corrupted the software image used for regular operations or has affected the endpoint's ability to connect. During our analysis, we found that current endpoint implementations fail to recover from security incidents targeting the endpoint's recoverability (cf. Section 4).

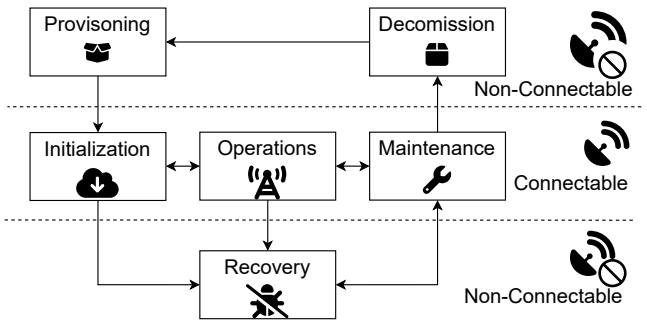


Figure 4: VSAT Endpoint Phases: Certain phases can move to specific other phases

3.4 Phase-based Threat Model

With knowledge of the different phases, we can now systematically capture and categorize phase-specific threats that undermine one of the four security goals. Figure 5 shows all threats identified during the following discussion, as well as the interfaces we later use to model attackers and phase transitions that describe which endpoint lifecycle phase can transition into which other.

3.4.1 Commissioning. The commissioning service adds new endpoints to the VSAT network. Since, at this point, the endpoint is non-connectable, there can also be no transition to the recovery phase, making *recoverability* not applicable. The commissioning phase's **availability** can be crucial, e. g., during an ongoing incident to bring a backup endpoint device online, or, considering a longer timescale, to replace broken endpoints. If attackers compromise recoverability, then new terminal commissioning is the only path to reconnect remote sites to the network again. Therefore, denying commissioning is referred to as *endpoint installation suppression*. Maliciously tampering with connection-related information **integrity**, such as *beamtables*, serves the purpose of establishing a *link AitM* attacker. Another attack vector targets the endpoint's identity information, such as a certificate, to replace a network identity. We refer to them as *network parameter replacement* and *endpoint impersonification*, respectively. Attackers may compromise **confidentiality** by identifying the new network user and their personal information that is required to issue an identifying certificate, resulting in a *network user identification* threat.

3.4.2 Initialization. This phase retrieves volatile and temporary network information that can be recovered by re-executing this phase. The **availability** of the initialization phase is critical to supply endpoints after a restart with volatile configuration, such as keys, and network addresses, i. e., through DHCP. This is critical, as during incident response, an updated software image might prompt a terminal restart. The Threat against this phase's availability impacts the re-attachment of an endpoint, resulting in *endpoint attachment denial*. Endpoint must identify faults and transition into the **recovery** phase.We refer to threats that inhibit this process and thus prevent the recoverability process as *recovery denial*. This threat is not specific to the initialization phase, but applies to the operations and maintenance phases as well. Threats against this phase's **integrity** attempt to interfere with a key exchange, network address signalling, or the endpoint authorization process. Since the integrity (but not availability) is threatened, an attacker might attempt to establish themselves as AitM by hijacking the mentioned key exchanges or by maliciously influencing the network addresses

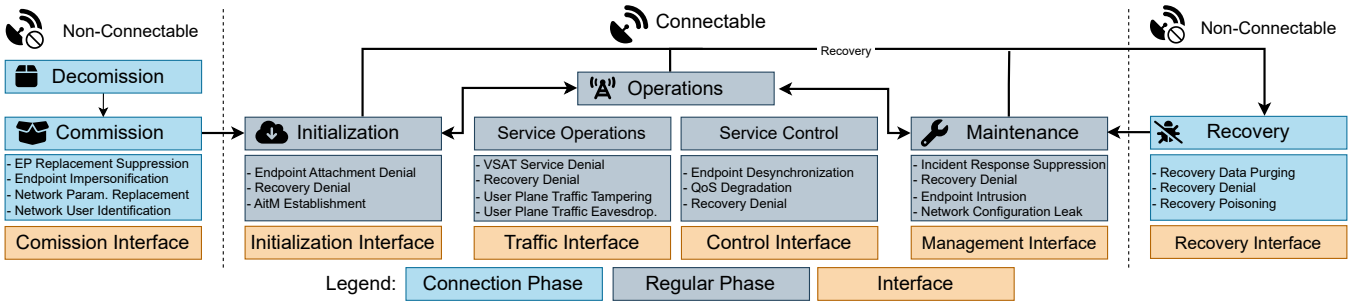


Figure 5: VSAT Phase Threats: Each phase has associated threats and an interface to model attackers

communicated. We refer to this category as *AitM establishment*. Information retrieved in this phase, such as the endpoint's IP address, is usually not **confidential**. Exchanged keys must be confidential; however, we omit these considerations since all commonly used key exchanges assume an eavesdropping attacker.

3.4.3 Service Operations. In the service operation phase, internet traffic is routed to the endpoint, usually using *DVB-S2*. The **availability** of the network service is paramount and the core goal of all surrounding security implications, which almost all aim to ensure the uninterrupted availability of transferring network payload traffic. We summarize these availability threats as *VSAT service denial*. Attackers that aim to inject malicious information in legitimate user plane traffic manipulate **integrity**, resulting in *user plane traffic tampering*. **Confidentiality** of network traffic is paramount due to the ease of eavesdropping. *user plane traffic eavesdropping* deserves special attention due to the difficulties in securing TCP-based traffic for GEO VSAT systems described by Pavur et al. [37]. In essence, the long distance for GEO-based systems imposes prolonged round-trip times of 600 ms. This long delay in connection with TCP's three-way handshake results in slow and sluggish connections. Vendors compensate this with PEP that pre-acknowledge TCP connections to shorten round-trip times for the initial handshake to the local endpoint device. However, this requires introspection of TCP connections, making many VPN solutions, e.g., IPSec-based solutions, incompatible as they do not expose the necessary TCP headers. Instead, SATCOM vendors rely on custom TCP header exposing solutions, which are often proprietary with few, if any, public insights into their security.

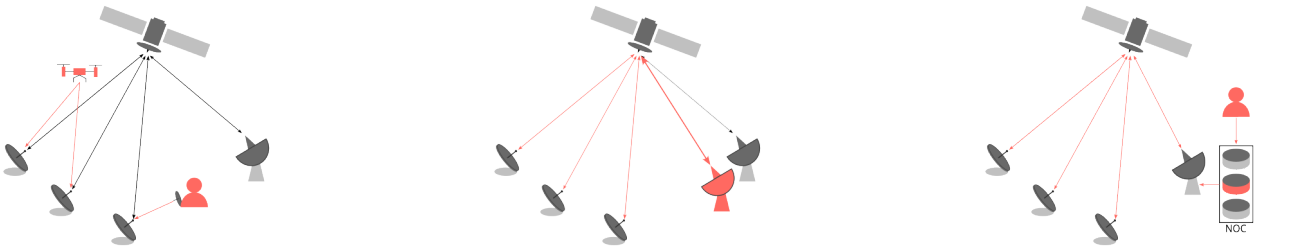
3.4.4 Service Control. The service control manages information that is required for service operations, such as time synchronization, adaptive coding updates, and channel declarations, which are typically updated every few seconds if not multiple times a second. Service control information ensures the QoS of the *service operations* and keeps it operational. An attacker breaks this phase's **integrity** by either crafting slightly wrong messages to degrade the QoS or crafting entirely wrong packets to desynchronize the hub and endpoint, thus resulting in degraded **availability** through *QoS degradation* or *endpoint desynchronization*. Since information for QoS, time synchronization, and similar services do not reveal meaningful insights, we omit **confidentiality** considerations.

3.4.5 Maintenance Phase. The maintenance phase aims to make *persisting changes* on the endpoint, such as software updates or configuration changes that alter the general operations of the network. The **availability** of the *maintenance* phase is especially critical during an ongoing security incident to patch vulnerabilities or change endpoint configurations.Since all maintenance during an active incident would be related to incident response, we consider threats to the maintenance phase's availability as *incident response suppression*. Link AitM attackers (cf. Section 3.1.2) might compromise **integrity** to tamper with software updates or configuration, either to achieve *persistent DoS* or to escalate the attack to an endpoint-side AitM, both of which require endpoint-side software or configuration changes. As such, we refer to them as *endpoint intrusion*. Attackers may comprise *confidentiality* by leverage a *network configuration leak* if they are not part of the VSAT network or if the distributed configuration differs between endpoints to gain network insights.

3.4.6 Recovery Phase. The recovery phase aims to restore information that allows an endpoint to return to a state where it is connectable to the VSAT network. Considering that the majority of recent incidents aimed to perform a *persistent DoS* (cf. Section 3.1.1), the security of this phase is crucial. In our model, this phase represents the recovery plan, such that we do not discuss *recoverability* here. The **availability** of the recovery phase can either be impeded by removing the data used as a recovery source or by denying the routine that recovers this data. Hence, we refer to *recovery data purging* and *recovery denial*, respectively. Tampering with the recovery data aims to break recovery data **integrity** and to restore malicious instead of intended data to the device, resulting in *recovery poisoning*. The recovery phase can only restore data from one of the other phases. Hence, we omit specific **confidentiality** considerations, as each phase's confidentiality considerations apply respectively.

3.5 Phase Interfaces

To model an attacker's access to individual phases, we introduce *interfaces*, shown in Figure 5 as orange boxes. Each phase has one corresponding interface, either in a *protected* or *open* state. An interface is considered *open* if traffic from the phase is not integrity protected, constituting a *trusted downlink* vulnerability. Interfaces with *trusted downlink* can be accessed by any attacker. In contrast,



(a) An Injection Attacker can inject traffic into antennas

(b) A Rogue VSAT Hub can inject and receive traffic from the VSAT network

(c) A (Semi-)Privileged Attacker compromises a *hub* service

Figure 6: Our attacker models vary in their ability to interact with the network and increase in strength from (a) to (c)

the *protected* interface requires a specific attacker model; for example, the initialization interface requires a *semi-privileged* attacker with access to the services emitting initialization traffic. Further, our model accounts for multi-stage attacks, where an attacker first compromises a phase, which *opens* access to another phase. For example, a vulnerability compromising the initialization phase might open the *management interface* for attackers, since an attacker-controlled maintenance service location has been specified. To illustrate this, an arrow leading from a phase block to an interface means that whatever vulnerability was identified in that phase, *opens* an interface to an attacker. We will later demonstrate this in our experimental analysis.

3.6 Attacker Models

The previously described *interfaces* allow us to model attackers with varying levels of privileged access. Based on the point of attack (i. e., the link or the hub), we identify four different attackers. Notably, we disregard attackers from the endpoint's LAN.

3.6.1 Link Attacker. A *link attacker* injects arbitrary traffic either directly to an endpoint's antenna or via a satellite that relays the signal (cf. Figures 6a and 6b). We distinguish the *link attackers* based on their capability to receive both the endpoint's *return link* and *forward link*, or only the latter. *link attackers* can only interfere with interfaces vulnerable to *trusted downlink*.

One-Way Traffic Injector. A *one-way traffic injector*, as shown in Figure 6a, might stand next to the victim endpoints or utilize a drone. In any case, the attacker can use a Software-Defined Radio (SDR) to emit arbitrary malicious signals and network packets into the antenna as if they were coming from the real VSAT network.

Rogue VSAT Hub. Extending the *one-way traffic injector*, an attacker could impersonate a VSAT hub (cf. Figure 6b) resembling a *rogue base station* known from mobile network security topics [3, 9]. This is significantly more complex, as an attacker must either intercept *and* inject traffic in the beam between endpoint and satellite or place a full ground station near the real VSAT hub.

3.6.2 Hub Attacker. We consider attacks that have compromised parts or all of the VSAT hub, letting us model incidents such as the *ViaSat* attacker. To avoid always assuming an omnipotent attacker, we also consider a *semi-privileged attacker*.

Semi-Privileged Attacker. We consider a *semi-privileged attacker* that compromised parts of a hub, e. g., from the internet, through a conventional cyberattack. During this process, the attacker gained control over a *hub*'s services (cf. Figure 6c) that do not distribute *persistent configuration*. Hence, this *semi-privileged* attacker can influence traffic of the *initialization* and *service control* phase, as the configuration for these phases is by definition non-persistent and restored through a *reboot*. However, such an attacker may escalate privileges by exploiting vulnerabilities in other phases, as we will show in Section 4.3.

Privileged Attacker. A *privileged attacker* has access to all *hub* services, all cryptographic material, and all technical details available about the network. In our model, such an attacker may interact with all phases but the *recovery* phase. Hence, even though this attacker has every ability to push configuration and software updates, a well-implemented *recovery* phase should protect even against such powerful attackers.

The introduction of attacker models completes our threat taxonomy: We have systematically studied attacker, derived security goals suited to thwart these attacks, and surveyed how these relate to the individual lifecycle phases of a VSAT system. This way, we systematically identified all relevant threats to VSAT networks.

4 EXPERIMENTAL ANALYSIS

We now conduct an experimental security analysis of two VSAT systems based on the taxonomy previously defined.

Responsible Disclosure. Following best practices, we have responsibly disclosed our findings to the affected providers. We contacted *iDirect* as Newtec merged with *iDirect* in 2020. Following the merger, Newtec no longer appears to have an independent operational presence. We disclosed the vulnerabilities to *iDirect* ourselves and with the help of the Swiss National Cyber Security Centre (NCSC). *iDirect* recently confirmed that they received the report, and we are collaborating to provide all necessary technical details. *ViaSat* confirmed having received the report, but—as far as we know—has not taken any further action.

4.1 Analysis Method

We perform an experimental security assessment of two VSAT systems to explore the attack surface. We work bottom up: First,

we dump and extract the endpoint's software. Then, we start by manually reverse engineering applications related to VSAT's network handling. After reverse engineering the applications related to wireless protocol handling, we verify our understanding using an experimental test setup and injecting traffic. To avoid traffic radio transmission, we patch the endpoint to directly receive the traffic via the LAN port. With a solid understanding of how the application communicates, we focus on the higher level and study how the application interfaces with the network to initially register to the network, receive updates, or receive configurations. Finally, we reverse engineer communication protocols (where needed) and uncover vulnerabilities in the protocol parsing logic.

4.2 iDirect MDM-Series

In our first analysis, we study the *MDM-series* from *iDirect*, which deploys the *Newtec Dialog* VSAT network [31]. *iDirect* operates through resellers that buy an *iDirect* VSAT hub and *iDirect* endpoints, which are distributed to customers. *iDirect*'s systems hold a 56% market share in commercial planes and private jets [21] as well as over 50% market share in maritime applications [11]. This dominant market position makes *iDirect* an interesting target, with any found issues potentially impacting a significant portion of the worldwide VSAT installations. However, it should be mentioned that the underlying Sat3Play technology in our case study certainly has a far lower market share as *iDirect* is also offering other solutions such as their *iDirect Velocity* and *Evolution*.

In the following, we describe our experimental setup, followed by a brief technical analysis, a security analysis, and two proof-of-concept attacks we tested on a real device.

Experimental Setup. We conducted our experiments on a standard live setup with the *iDirect MDM 2200 (NTC 22.99)* endpoint connected to a commercial satellite antenna. The endpoint runs the software version *2.2.6.19* from October 2014 — we ensured that the most current updates provide the same version. The endpoint had been replaced by the ISP and freshly installed in March 2021 to replace our older terminal, showing the longevity of VSAT endpoints. In total, our setup costs $500 upfront for endpoint and installation, as well as $70 monthly for active internet service over the endpoint. We further validated our analysis on a previous endpoint version (*iDirect NTC 22.18*), which runs the exact same software version but compiled for PowerPC, including all the same vulnerabilities. We extracted the endpoint's software by gaining initial remote code execution via a web interface command injection vulnerability. Notably, we only used this vulnerability to extract the software, not for further exploitation.

4.2.1 Technical Description. The endpoint uses a private RSA key generated during *commissioning* to connect to the network as part of the *initialization* phase. With this key, it can decrypt two session keys, one used for internet traffic and the other for C2 traffic. Interestingly, the endpoint works even without a private RSA key and uses the airmac address (equivalent to a MAC address) as the key for the internet traffic, while the C2 key is not set.

The endpoint deploys the *Enhanced TCP (ETCP)* protocol during the *service operations* phase, which relies on a Performance Enhancing Proxy (PEP) that pre-acknowledges TCP traffic. PEP

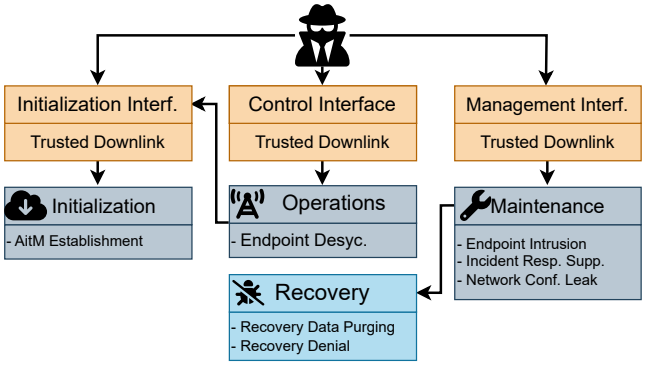


Figure 7: MDM2200 Vulnerabilities: Overview of all vulnerabilities per phase and their interactions

was analyzed by Pavur et al. [37]. This internet traffic is encrypted using the respective key.

The *service operations* encompass timing packets to synchronize an exact time between hub and endpoints, packets to adapt coding and modulation, and multi-input stream identifiers that determine how an endpoint should distribute traffic across multiple channels. All of this information has to be provided continuously for uninterrupted operations.

The endpoint's *maintenance* phase consists of software updates and persistent endpoint configuration. To this end, the hub continuously sends *update signalization* packets that specify a port and multicast address on which endpoints must listen to receive the currently up-to-date software image. All software images are permanently and repeatedly broadcast, which is also referred to internally as *lifeline*. It provides a last chance to receive a non-corrupted software image and is therefore categorized as as part of the *recovery* phase. Persistent endpoint configuration is performed over a custom protocol using Google's *protobuf* serialization format. There is a total of 15 configuration messages, including a *session key message*, an *endpoint certification message*, and a *network configuration message*, which sets the network addresses for the *endpoint*, a *name server*, and the *internet gateway*. The custom configuration messages have a field determining if it is encrypted.

The endpoint's *recovery* phase internally detects a corrupted software image by calculating a checksum at boot time and switching to a redundant flash chip. From there, it then attempts to retrieve a fresh image via the lifeline. However, all aspects, including the internal boot arguments, can be modified with *root* access.

4.2.2 Security Analysis. We conducted a security analysis and identified five issues.

Trusted Downlink & Configuration Leak. Even when the endpoint receives a valid key for C2 traffic, it generally does not encrypt or authenticate C2 traffic. Consequently, any application sending or receiving C2 traffic must implement cryptographic protection individually. Yet, only the application responsible for configuration messages implements encryption and uses the encryption indication field in the custom *protobuf*-based protocol. On the contrary, software update packets are not protected, allowing attackers to broadcast arbitrary software updates through the *trusted downlink*

on the *maintenance interface*, leading to an *endpoint intrusion* vulnerability. Additionally, all messages from the *service control* phase are unprotected, allowing attackers to send arbitrary information to the *control interface*. Further, since many messages related to ACM and input stream identifiers are not protected through encryption, this leads to a *network configuration leak*.

Encryption Bypass. While the configuration message protocol is encrypted, this encryption can be bypassed. The modem handles each packet in plain text *if* a specific header field is set to zero. Subsequent processing steps process this packet the same as other packets that arrive encrypted. From our reverse engineering efforts, we determined this header field to be a packet counter, where the first packet (with counter zero) is unencrypted. While this poses no inherent problem on its own (and may even be required for scenarios where the first packet of a key exchange cannot be encrypted), the subsequent program logic contains a bug that allows misuse of this behavior. More precisely, the packet counter field is not validated but taken as specified in the packet (i.e., it is attacker-controlled), allowing the sending of arbitrary unencrypted packets. This causes an *AitM establishment* vulnerability.

Weak Cryptography. Configuration messages, use *Blowfish* [43] in Electronic Codebook (ECB) mode, allowing attackers to re-order, add, remove, and replay blocks arbitrarily. Attackers can be reasonably assumed to have plain-text knowledge of such messages as they receive similar messages. This allows attackers to replay old configurations, leading to an *incident response suppression*, even if encryption was enforced.

Memory Corruption. We noticed a lack of modern software defenses, such as Address Space Layout Randomization (ASLR) or stack cookies, which mitigate consequences of memory corruption vulnerabilities. Further, discouraged C functions (e. g., `strcpy` or `sprintf`) are widely used throughout the software. After an initial analysis, we found two memory corruption vulnerabilities in the update signalization process and a tool that writes software images to the flash memory. These vulnerabilities provide attackers with *root* privileges. Recall that the recovery code is not separated and can be modified by a privileged user, rendering the *MDM-series* vulnerable to *recovery data purging*.

4.2.3 Proof-of-Concept Attacks. We experimentally verify the feasibility of two proof-of-concept attacks using vulnerabilities shown in Figure 7 to ensure that our previous analysis did not miss any countermeasures. For both PoCs, we use our weakest attacker possible, a *one-way traffic injector* (cf. Section 3.6). We tested both attacks with a full wireless setup, for which we implemented Newtec's S3P implementation of the DVB-RCS protocol. The protocol utilizes multiple forward carrier channels manually configured at each endpoint. We transmit our exploit signal on each carrier channel to target endpoints regardless of their configuration. Finally, the antenna expects signals to be transmitted on the K_u-band, which are then down-converted to L-band using a low-noise block (LNB). Since regular SDR are usually limited to around 6 GHz, we utilize a block upconverter (BUC), specifically a UMT-TV BUC-Ku002-10.6 v2.0. The total cost of our setup is about 1000$.

Privileged RCE via Signal Injection. The endpoint accepts maintenance traffic at any time, without protection via a trusted downlink on the management interface. There, we can use a single malicious packet and leverage a memory corruption vulnerability in the software update mechanism that allows attackers to gain remote code execution on any endpoint using a single *update signalization* packet, thereby exploiting the *endpoint intrusion*. The application parses a string of an arbitrary length in a stack buffer of limited size using the `sscanf` function. The exploitation process is only hindered through a *non-executable* stack. As the program runs with *root* privileges, we are able to write to both redundant flash images, allowing us to wipe the recovery routine, successfully exploiting the *recovery data purging* vulnerability.

Moreover, a *Rogue VSAT Hub attacker* can send this malicious packet to all endpoints via the broadcast address over the satellite, since they are continuously listening on the lifeline. We note that an injection attacker with a sufficiently strong antenna could also break all endpoints by relaying traffic via the satellite to all endpoints. Due to ethical and legal reasons, we can obviously not test the vulnerability on the entire VSAT network. However, we verified this attack in a lab environment, targeting only our endpoint by using the broadcast address instead of the individual address.

VSAT Session Takeover. An endpoint requires a continuous stream of time synchronization packets as part of the *service control* phase. Sending broken packets or jamming these packets for five seconds causes the endpoint to lose synchronization with the hub, resulting in an *endpoint desynchronization*. The endpoint then attempts to restore the network connection by returning to the *initialization phase*, which accepts the following traffic without protection, resulting in a *trusted downlink* on the *initialization interface*. From there, the attacker can answer the renewed synchronization attempt before the legitimate hub can (e. g., through physical proximity), allowing the attacker to set parameters that send the traffic to the attacker. The attacker can answer the initial request from the hub and thus perform an *AitM establishment* from malicious parameters sent to the endpoint. Again, we experimentally verified this attack using our wireless test setup.

In summary, even the weakest attacker can take over an endpoint; a rogue hub can even take over the entire network.

4.3 ViaSat Surfbeam

We evaluate the *Surfbeam 2* system from *ViaSat*, which was attacked in Ukraine during the *ViaSat* incident (cf. Section 2.2). *ViaSat*, a significant player in the industry, delivers satellite systems to governments, including tactical products to the US and other militaries [48]. In addition, *ViaSat* offers its service in many categories, such as maritime applications, in-flight connectivity (1,500 aircraft in 2021), and consumer applications, with around 600,000 subscribers in the US [48].

Experimental Setup. We conducted our analysis by purchasing a *Surfbeam 2 (RM4100)* on *eBay* for $60 after the *ViaSat* incident. The system used software version `3.7.3.10.9` from 2017 (according to timestamps of files and information included in license files), which is the firmware version involved in the incident [42]. We obtained root access via a UART port and dumped the software [24].

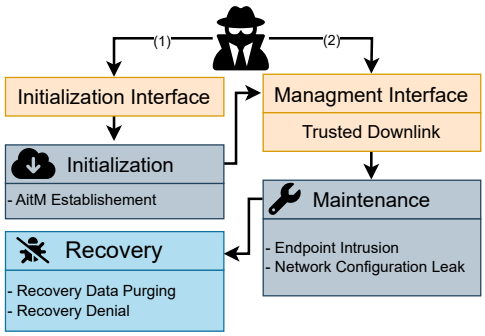


Figure 8: Surfbeam Vulnerabilities: Overview of all vulnerabilities per phase and their interactions

4.3.1 Technical Analysis. The system uses DVB-S2 for the physical layer forward link and DOCSIS Media Access Control (MAC) layers for the data link layer. The MAC layer thereby consists of sublayers, including a security layer that provides an encrypted and authenticated link. The *initialization* phase of the endpoint includes the necessary DOCSIS messages that register the endpoint for the current session with the network, set up the public key management, receive DHCP messages, and conduct further setup. Importantly, these DHCP messages use vendor-specific options (cf. RFC 2132 [12]) to hold unconventional configuration values that can modify the URL of configuration servers. The *service operations* phase exchanges internet traffic with the hub and is secured by encryption and authentication via the DOCSIS MAC security layer. The *service control* layer exchanges messages for synchronizing timing, adjusting frequency offsets and the power level, amongst other things. The *maintenance* phase uses CPE WAN Management Protocol (CWMP) configuration messages and performs software updates. The updates can be downloaded via at least four different methods, through File Transfer Protocol (FTP), a script using `wget`, and two approaches through the CWMP client. The downloaded image is then unwrapped and decrypted using a secret key stored on the endpoint before it is installed. The *recovery* phase consists of a separate image download mechanism, the *lifeline*. The *lifeline* works akin to the process described for *iDirect* via multicast.

4.3.2 Security Analysis. Figure 8 details all the vulnerabilities we found via our threat model and the potential exploitation path.

Service Location Tampering. During the *initialization* phase, a *semi-privileged attacker* can send DHCP messages used for temporary configuration and during the endpoint's boot-up phase. Figure 8 (left) models this attacker, taking the path through the *Initialization Interface* **(1)**. Notably, DHCP allows for the specification of service addresses through vendor-specific options, such as the URL of the CWMP or FTP server. Thus, an attacker can use a single DHCP message to set the URL of the FTP/CWMP server to one they control, which is an *AitM establishment* vulnerability.

Missing Service Authentication. Neither FTP nor CWMP clients verify that the target URL is in the VSAT hub or network range. Further, there is no server authentication towards the client, which allows an attacker to set up a rogue FTP or CWMP server and point any endpoint to it. As the URL's traffic is trusted, this is a *trusted downlink* into the *maintenance phase*, see path **(2)** in Figure 8. Hence, the attacker can send arbitrary packets during the maintenance phase, making them fully privileged. Further, through the CWMP client, an attacker can extract the configuration stored on the endpoint, resulting in a *network configuration leak*.

Update Decryption Bypass. By default, decryption of updates is only attempted for a single system vendor; even then, it can be overruled by uploading an empty, specifically named file. For other vendors, the endpoint checks for another file to enforce update decryption, overruling the default non-encryption. Removing that file essentially disables encryption. Crucially, the CWMP client on the endpoint allows the CWMP server to add and remove these files, thus bypassing all update decryption and signature requirements.

Shared Recovery Resources. Since the recovery phase relies on the same binary on the same operating system that an attacker would compromise as part of the *endpoint intrusion*, an attacker can arbitrarily break the *lifeline* recovery procedure, resulting in *recovery data purging* and *recovery denial*.

4.3.3 Proof-of-Concept Attack. We verify the exploitability of our findings. For simplicity reasons, we tweaked the device's traffic application to accept traffic from the LAN as if it were coming from the antenna. Note that this change has no effect on exploitability but simplifies implementation.

For our PoC exploit, we assume a *semi-privileged* attacker that sends a malicious DHCP packet to the endpoint after putting it into the DHCP accepting mode using the respective DOCSIS dynamic service addition flow. The DHCP packet sets the address of the CWMP server using the vendor-specific DHCP options. The endpoint then restarts the CWMP client with the new address. Since the new address is not restricted, the client sends a request to the new server address, and due to missing service authentication, the client accepts the malicious CWMP server address and starts the CWMP communication. After connecting, the server instructs the client to download a new software image by providing a download URL, which is again not restricted or authenticated. Hence, the client starts downloading the image and then attempts to decrypt it if necessary, which was not the case. However, we verified that the vendor-based decryption enforcement could be bypassed by uploading the corresponding configuration file before the update. Further, since software updates are not signed, the client cannot verify if the update is legitimate. Hence, the endpoint installs the attacker-controlled image and reboots. We verified this until the step of rebooting, which we omitted to avoid any chance of breaking our terminal; however, we verified that the tampered software image was stored in the correct boot location.

Our experimental analysis of two endpoints reveals a dire state of security, allowing an attacker to take control of an endpoint and even break the recovery method in both cases.

5 INSECURE VSAT DESIGN PRACTICES

We discuss three inherently insecure VSAT network design practices we discovered during the development of our threat taxonomy and the experimental analyses.

Problematic Trust Hierarchies. The *ViaSat* experiment reveals a problem that is inherently difficult to address. While the clients on the endpoint trust the service address to be valid, this issue can be solved through certificates. More worryingly, even if such server authentication is enforced, an attacker who compromises the maintenance service can still break all endpoints in the network by distributing malicious updates, prompting on-site personnel. This aspect is captured in our taxonomy, as the *management interface* access immediately leads to an *endpoint intrusion*. This shows a *trust hierarchy* where all endpoints must fully trust the hub. We believe this to be an inherent weakness. A solution could be decentralized approaches [16] such as *witnessing* [32, 44], where several endpoints would have to find software and configuration updates to be valid before *co-signing* them.

Inherently Broken Recovery. Our threat taxonomy underlines the importance of the recovery phase, even under adversarial circumstances. Ideally, a mechanism would test if a given software update or configuration allows for network connection and otherwise rolls back a software image or piece of configuration. This should be feasible in general, assuming that, during a DoS attack, hubs are only compromised for a limited time before dedicated staff recovers them from the incident. Such a system would require *non-shared resources* that an attacker cannot access with a compromised update. This would require a dedicated routine to recover from a malicious software image. Crucially, this routine must have a different *root of trust* on the endpoint. While existing research explores techniques to identify faulty software updates, they do not account for malicious updates [7, 38].

Unintuitive Network Designs. Our threat taxonomy reveals another problem: In our *ViaSat* experiment, an attacker could use a single DHCP packet to gain the ability to distribute malicious software updates. This issue arises because the packet mixes temporary configuration with typically persistent configuration, such as the URL of the FTP and CWMP server. This mix of configuration types leads to *AitM establishment*. While it appears trivial that an attacker who can distribute such configurations may set these URLs, this might not be obvious to someone configuring the network. A network administrator setting VPN and firewall rules knows DHCP but might be unaware of this obscure and unintuitive extension. At first glance, this problem is not VSAT-specific. However, in the VSAT industry, it is common practice that one company builds the endpoints and hubs and sells them to another company hosting the network. Due to this practice, such counterintuitive information is easily lost or buried in manuals.

Based on our results, we believe that in order to secure VSAT networks, it is at least necessary to break the aforementioned *trust hierarchy* and to enable endpoints to reliably perform *recovery*.

6 RELATED WORK

Adelsbach and Greveler first pointed out the significant attack surface of the unencrypted DVB-S ecosystem [2]. Later, presentations at hacker conferences picked up the threat with further proof-of-concept attacks [14, 26]. In recent years, there has been renewed interest in the topic, fuelled by the explosive growth of satellite infrastructure. In the wake of these developments, Giuliari et al. have discussed attacks on LEO-based internet communication [17]. Pavur et al. have revisited the topic of VSAT and DVB-S security, proving that the same issues concerning integrity and confidentiality still exist but that impact (e. g., on maritime and aviation customers) and ease of exploitation have grown [4, 35, 36]. This is also evidenced by recent surveys in the sector: Pavur and Martinovic have outlined the history of space incidents and the need for renewed space security research efforts [34]. Tedeschi et al. investigate link-layer security in satellite communications beyond navigation satellites [45]. Finally, Yue et al. survey the literature with a focus on LEO satellite security and reliability [51].

Increased concerns over the wireless spoofing of non-authenticated satellites to unsuspecting ground users have been analyzed recently by Salkied et al. [40, 41] Countermeasures to such threats have also been addressed recently. Oligeri et al. [33] and Jedermann et al. [23] propose transparent defense mechanisms based on physical-layer properties. Abdelsalam et al. [1] survey open problems in transparent physical layer security for satellites.

7 CONCLUSION

In this paper, we introduced a threat taxonomy that enables accurate and multi-stage modeling of attacks against VSAT systems while accounting for network-intrinsic details. We derive attacker goals from recent VSAT incidents and distill them into security goals. We emphasize the *recoverability* security goal, which is required to secure remote sites without possible physical intervention. Next, we divide VSAT network operations into six phases and formulate threats against each phase based on the security goals. We evaluate the practicality of our threat model using two real-world VSAT systems, one of which was involved in a recent large-scale incident. Finally, we discuss the vulnerabilities inherent in current VSAT systems designs.

ACKNOWLEDGMENTS

We thank Knut Eckstein from the European Space Agency for his helpful feedback. The work was partially supported by the MKW-NRW research training group SecHuman.

REFERENCES

[1] Nora Abdelsalam, Saif Al-Kuwari, and Aiman Erbad. 2023. Physical Layer Security in Satellite Communication: State-of-the-art and Open Problems. *arXiv preprint arXiv:2301.03672* (2023).
[2] André Adelsbach and Ulrich Greveler. 2005. Satellite Communication without Privacy–Attacker's Paradise. In *Sicherheit 2005, Schutz und Zuverlässigkeit*. Gesellschaft für Informatik eV, 257–268.
[3] Michel Barbeau and Jean-Marc Robert. 2006. Rogue-base Station Detection in WiMax/802.16 Wireless Access Networks. *Annales des Télécommunications* 61 (2006), 1300–1313.
[4] Georg Baselt, Martin Strohmeier, James Pavur, Vincent Lenders, and Ivan Martinovic. 2022. Security and Privacy Issues of Satellite Communication in the Aviation Domain. In *International Conference on Cyber Conflict*.
[5] Przemysław Bibik, Stanisław Gradolewski, Wojciech Zawiślak, Jacek Zbudniewek, Radoslav Darakchiev, Jerzy Krçżel, Mateusz Michalski, and Krzysztof Strzelczyk. 2012. Problems of Detecting Unauthorized Satellite Transmissions from the VSAT Terminals. In *2012 Military Communications and Information Systems Conference (MCC)*.
[6] Nicolò Boschetti, Nathaniel G Gordon, and Gregory Falco. 2022. Space Cybersecurity Lessons Learned from The ViaSat Cyberattack. In *AIAA ASCEND*.
[7] Stephen Brown and Cormac J Sreenan. 2009. Software Update Recovery for Wireless Sensor Networks. In *International Conference on Sensor Applications, Experimentation and Logistics*.
[8] D.M. Chitre and J.S. McCoskey. 1988. VSAT Networks: Architectures, Protocols, and Management. *IEEE Communications Magazine* 26 (1988), 28–38.

[9] Merlin Chlosta, David Rupprecht, Thorsten Holz, and Christina Pöpper. 2019. LTE Security Disabled: Misconfiguration in Commercial Networks. In *ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*.

[10] International cyber law: interactive toolkit. 2022. Viasat KA-SAT Attack (2022) — International Cyber Law: Interactive Toolkit. https://cyberlaw.ccdcoe.org/w/index.php?title=Viasat_KA-SAT_attack_(2022)&oldid=3408.

[11] Digital Ship. 2020. Marlink Remains Largest Retail VSAT Service Provider in 2019. https://www.thedigitalship.com/news/maritime-satellite-communications/item/6826-marlink-remains-largest-retail-vsat-service-provider-in-2019.

[12] Ralph Droms and Steve Alexander. 1997. DHCP Options and BOOTP Vendor Extensions. RFC 2132. https://doi.org/10.17487/RFC2132 https://www.rfc-editor.org/info/rfc2132.

[13] Kate Duffy. 2022. Elon Musk says Russia has stepped up efforts to jam SpaceX's Starlink in Ukraine. https://www.businessinsider.com/elon-musk-spacex-russia-ramps-up-efforts-jam-starlink-ukraine-2022-5.

[14] Leonardo Egea. 2010. Playing in a Satellite Environment 1.2. http://www.blackhat.com/presentations/bh-dc-10/Nve_Leonardo/BlackHat-DC-2010-Nve-Playing-with-SAT-1.2-wp.pdf.

[15] European Space Agency. 2023. Space Attacks and Countermeasures Engineering Shield (SPACE-SHIELD). https://spaceshield.esa.int/.

[16] Tiago M Fernández-Caramés and Paula Fraga-Lamas. 2018. A Review on the Use of Blockchain for the Internet of Things. *IEEE Access* 6 (2018), 32979–33001.

[17] Giacomo Giuliari, Tommaso Ciussani, Adrian Perrig, and Ankit Singla. 2021. ICARUS: Attacking Low Earth Orbit Satellite Networks. In *USENIX Annual Technical Conference (ATC)*.

[18] Se Gi Hong and Chi-Jiun Su. 2015. ASAP: Fast, Controllable, and Deployable Multiple Networking System for Satellite Networks. In *IEEE Global Communications Conference (GLOBECOM)*.

[19] Yurong Hu and V.O.K. Li. 2001. Satellite-based Internet: A Tutorial. *IEEE Communications Magazine* 39 (2001), 154 – 162.

[20] Todd E Humphreys, Peter A Iannucci, Zacharias M Komodromos, and Andrew M Graff. 2023. Signal Structure of the Starlink Ku-Band Downlink. *IEEE Trans. Aerospace Electron. Systems* PP (2023), 1–16.

[21] iDirect. 2020. iDirect-Corporate-Fact-Sheet. https://www.idirect.net/wp-content/uploads/2020/01/2020-STE-iDirect-Corporate-Fact-Sheet-US-1.pdf.

[22] Valerie Insinna. 2022. SpaceX Beating Russian Jamming Attack was 'Eyewatering': DoD Official. *Breaking Defense* (2022). https://breakingdefense.com/2022/04/spacex-beating-russian-jamming-attack-was-eyewatering-dod-official/.

[23] Eric Jedermann, Martin Strohmeier, Matthias Schäfer, Jens Schmitt, and Vincent Lenders. 2021. Orbit-based Authentication using TDOA Signatures in Satellite Networks. In *ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*.

[24] Eric Johnston. 2022. KA-SAT Technical System: My Guess as to How it Works. https://www.satsig.net/tooway/ka-sat-system-technical.htm.

[25] Kratos. 2023. Threat Briefing: Russian Satellite Service Provider Dozor-Teleport Targeted by Cyberattack. https://www.kratosdefense.com/constellations/articles/russian-satellite-service-provider-dozor-teleport-targeted-by-cyberattack.

[26] Adam Laurie. 2009. $atellite Hacking for Fun & Pr0fit! Blackhat.

[27] Katrina Manson. 2023. The Satellite Hack Everyone is Finally Talking About. *Bloomberg* (2023). https://www.bloomberg.com/features/2023-russia-viasat-hack-ukraine/#xj4y7vzkg.

[28] Joseph Menn. 2023. Cyberattack Knocks out Satellite Communications for Russian Military. *Washington Post* (2023). https://www.washingtonpost.com/technology/2023/06/30/satellite-hacked-russian-military/.

[29] Christopher Miller, Mark Scott, and Bryan Bender. 2022. UkraineX: How Elon Musk's Space Satellites Changed the War on the Ground. https://www.politico.eu/article/elon-musk-ukraine-starlink/.

[30] Glyn Mood. 2016. New Snowden Leaks Reveal "Collect it All" Surveillance was Born in the UK. https://arstechnica.com/tech-policy/2016/09/snowden-leaks-collect-all-signals-surveillance-born-in-uk/.

[31] newtec. 2016. MDM2200 IP Satellite Modem. http://nposp.ru/wp-content/uploads/2016/11/newtec-mdm2200-on-the-newtec-dialog-platform.pdf.

[32] Kirill Nikitin, Eleftherios Kokoris-Kogias, Philipp Jovanovic, Nicolas Gailly, Linus Gasser, Ismail Khoffi, Justin Cappos, and Bryan Ford. 2017. CHAINIAC: Proactive Software-Update Transparency via Collectively Signed Skipchains and Verified Builds. In *USENIX Security Symposium*.

[33] Gabriele Oligeri, Savio Sciancalepore, and Roberto Di Pietro. 2020. GNSS Spoofing Detection via Opportunistic IRIDIUM Signals. In *ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*.

[34] James Pavur and Ivan Martinovic. 2022. Building a Launchpad for Satellite Cybersecurity Research: Lessons from 60 Years of Spaceflight. *Journal of Cybersecurity* (2022), tyac008.

[35] James Pavur, Daniel Moser, Vincent Lenders, and Ivan Martinovic. 2019. 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)*.

[36] James Pavur, Daniel Moser, Martin Strohmeier, Vincent Lenders, and Ivan Martinovic. 2020. A Tale of Sea and Sky: On the Security of Maritime VSAT Communications. In *IEEE Symposium on Security and Privacy (S&P)*.

[37] JC Pavur, Martin Strohmeier, Vincent Lenders, and Ivan Martinovic. 2021. QPEP: An Actionable Approach to Secure and Performant Broadband from Geostationary Orbit. In *Symposium on Network and Distributed System Security (NDSS)*.

[38] Alexandru Radovici, Ioana Culic, Daniel Rosner, and Flavia Oprea. 2020. A Model for the Remote Deployment, Update, and Safe Recovery for Commercial Sensor-based IoT Systems. *Sensors* 20 (2020), 4393.

[39] Bingyin Ren, Hailong Ge, Guangfei Xu, and Yongxin Zhang. 2023. Anti-Jamming Analysis and Application of Starlink System. In *International Conference on Networking, Informatics and Computing (ICNETIC)*.

[40] Edd Salkield, Simon Birnbach, Sebastian Kohler, Richard Baker, Martin Strohmeier, and Ivan Martinovic. 2023. Firefly: Spoofing Earth Observation Satellite Data through Radio Overshadowing. In *Workshop on the Security of Space and Satellite Systems (SpaceSec)*.

[41] Edd Salkield, Marcell Szakály, Joshua Smailes, Sebastian Köhler, Simon Birnbach, Martin Strohmeier, and Ivan Martinovic. 2023. Satellite Spoofing from A to Z: On the Requirements of Satellite Downlink Overshadowing Attacks. In *ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*.

[42] Ruben Santamarta. 2022. VIASAT Incident: From Speculation to Technical Details. https://www.reversemode.com/2022/03/viasat-incident-from-speculation-to.html.

[43] Bruce Schneier. 1993. Description of a new variable-length key, 64-bit block cipher (Blowfish). In *International Workshop on Fast Software Encryption*.

[44] Ewa Syta, Iulia Tamas, Dylan Visher, David Isaac Wolinsky, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ismail Khoffi, and Bryan Ford. 2016. Keeping Authorities "Honest or Bust" with Decentralized Witness Cosigning. In *IEEE Symposium on Security and Privacy (S&P)*.

[45] Pietro Tedeschi, Savio Sciancalepore, and Roberto Di Pietro. 2022. Satellite-based Communications Security: A Survey of Threats, Solutions, and Research Challenges. *Computer Networks* 216 (2022), 109246.

[46] The Aerospace Corporation. 2023. Space Attack Research & Tactic Analysis (SPARTA). https://sparta.aerospace.org/.

[47] Patrick Tucker. 2022. As Satellite Images Reshape Conflict, Worries Mount About Keeping Them Safe. https://www.defenseone.com/technology/2022/04/satellite-images-reshape-conflict-worries-mount-about-keeping-them-safe/366265/.

[48] ViaSat. 2021. Q4 FY21, Shareholder Letter. https://investors.viasat.com/static-files/393791ed-ba16-4116-a556-cebf19ae5eb1.

[49] Viasat Corporate. 2022. KA-SAT Network Cyber Attack Overview. https://www.viasat.com/about/newsroom/blog/ka-sat-network-cyber-attack-overview/.

[50] AJ Vicens. 2023. Russian Telecom Confirms Hack after Group Backing Wagner Boasted about an Attack. *Cyberscopp* (2023). https://cyberscoop.com/russia-satellite-hack-wagner/.

[51] Pingyue Yue, Jianping An, Jiankang Zhang, Jia Ye, Gaofeng Pan, Shuai Wang, Pei Xiao, and Lajos Hanzo. 2023. Low Earth Orbit Satellite Security and Reliability: Issues, Solutions, and the Road Ahead. *IEEE Communications Surveys & Tutorials* 25 (2023).