assail

KEYNOTE · BLACK HAT USA · AUGUST 2026

From Ground Control to Star Wars: Hacking Satellite Constellations Using Autonomous AI

How an autonomous offensive security platform hacks the satellite ground segment through its APIs, and what it takes to defend it.



ALISSA KNIGHT · CEO, ASSAIL, INC.

ASSAIL, INC



YOUR PRESENTER

Alissa Knight

Chief Executive Officer, Assail

Recovering hacker of 26 years of breaking into banks, hospitals, passenger vehicles, drones, and satellites through APIs. Sold three cybersecurity companies. Laptop and hacking gear curated on display at the Mob Museum.

- 26 YRS OFFENSIVE SECURITY
- PUBLISHED AUTHOR
- RECOVERING HACKER
- SATELLITE AND DRONE HACKER

BRIEFING ORDER

Agenda

01	The Basics	Why annual testing can no longer keep pace
02	The Exposure Gap in Hacking Satellites	Autonomous agents, orchestrated by Polemos
03	The Kill Chain	One flaw is a finding. Chained together, they become command authority over a constellation.
04	The Demo	Live hack of ground control system of meridian satellite constellation
05	The Defense	Defense-in-Depth and how to protect against it

SECTION 01

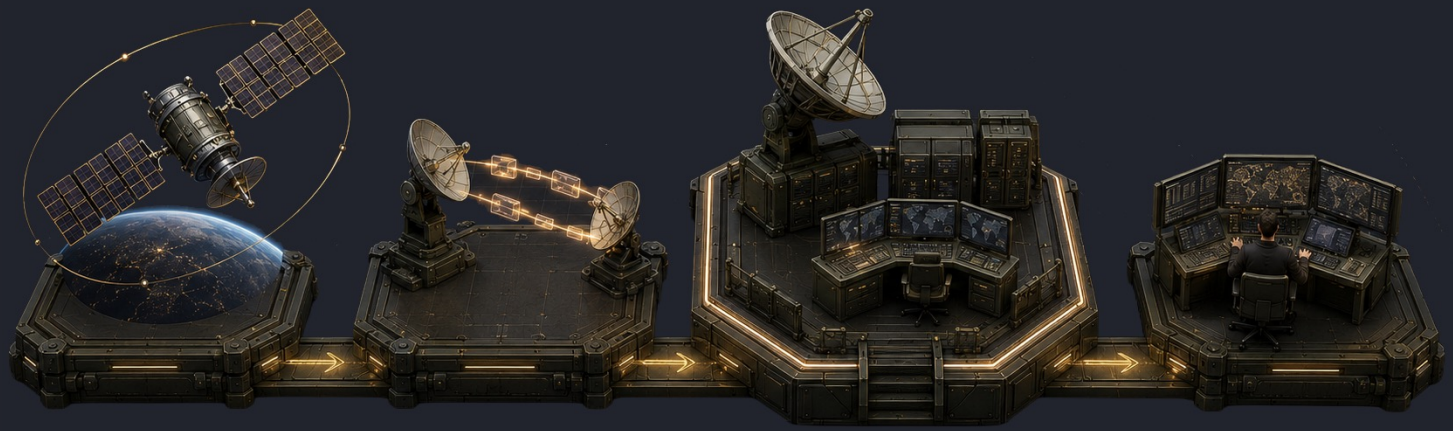
The Foundation

Demystifying ground segments, satellites, and their attack surfaces



SECTION 01 · FOUNDATIONS

How a satellite system actually works



Space

The satellite: bus, subsystems, and payload in orbit.

Link

RF uplink and downlink. Framed binary, CCSDS. Hard to touch.

Ground

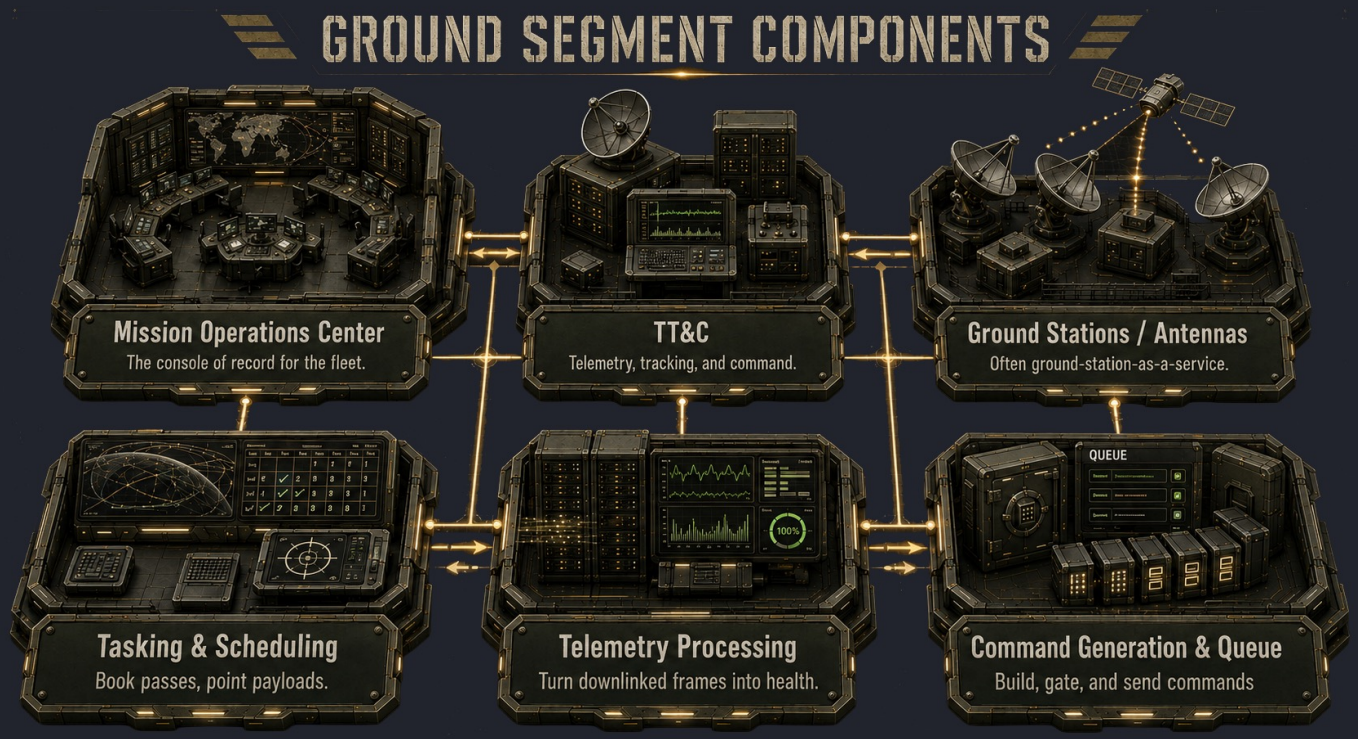
Mission ops, TT&C, antennas, and software. Command lives here.

User

Tasking customers and data consumers who drive demand.

SECTION 01 · FOUNDATIONS

Inside the ground segment



SECTION 01 · FOUNDATIONS

Why APIs are now everywhere in space

Every one of these is a request. Break the request, and you have the bird.





SECTION 02

The exposure gap in hacking satellites

Modern satellite operations run through the ground segment. The ground segment runs on APIs. That is where the door is, and that is where we knocked.

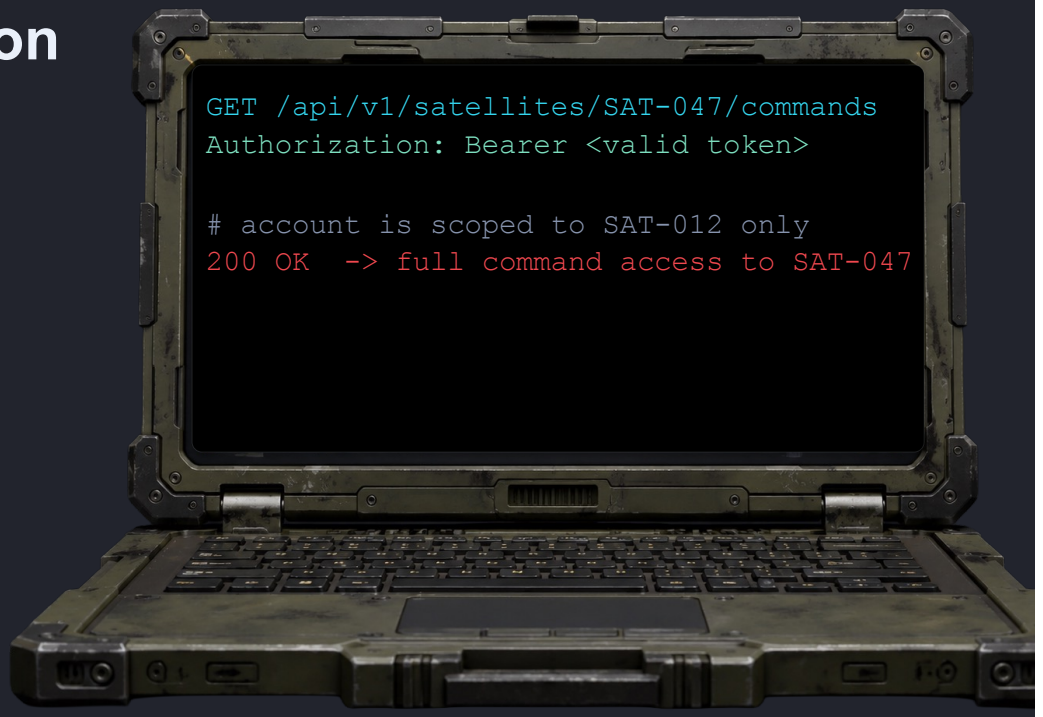


SECTION 02 · THE EXPOSURE GAP

Broken Object Level Authorization

The system checks that you are authenticated. It never checks that this satellite is yours. Change the identifier, and you command someone else's bird.

- Traffic is authenticated
- Traffic is not authorized
- The most common attack vector

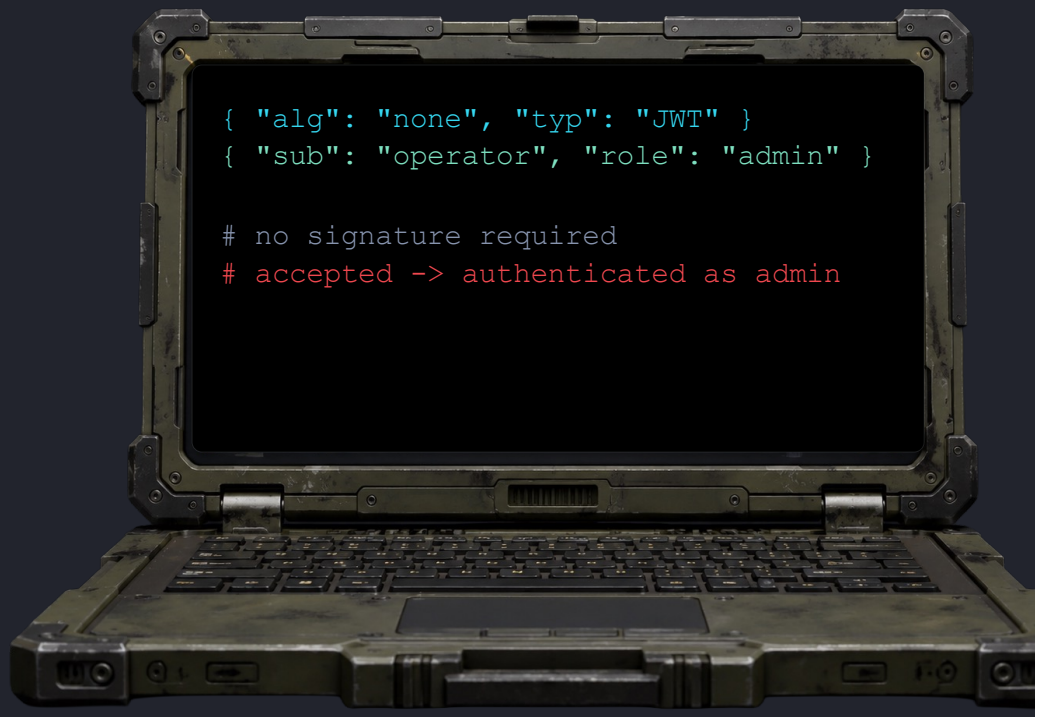


SECTION 02 · THE EXPOSURE GAP

Broken Authentication

Weak or forgeable tokens. Signatures that accept "none." Tokens that never expire. If identity can be faked, none of the other controls matter.

- Weak or guessable credentials: default operator logins, shared API keys, and no brute-force or rate limiting on the auth endpoint, so the front door opens to a wordlist.
- Broken signature validation: tokens accepted with alg: none or a weak, leaked signing secret, letting an attacker forge a valid identity and set their own role.
- No lifecycle controls: tokens that never expire and are not revoked or rotated, so one captured credential grants standing access long after it should have died.

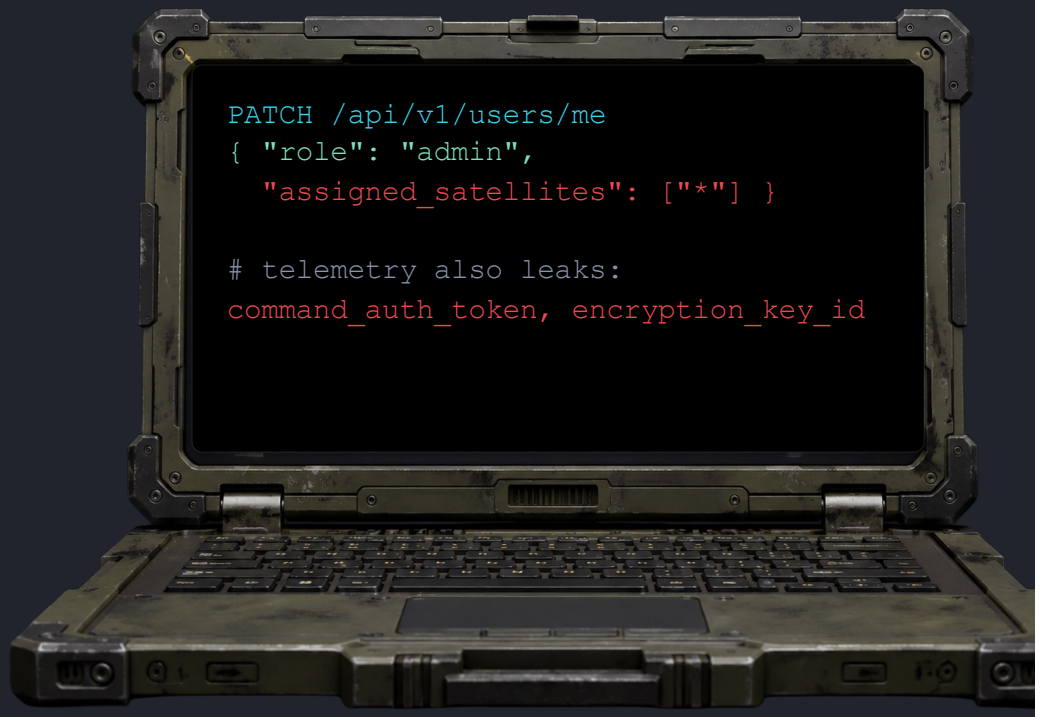


SECTION 02 · THE EXPOSURE GAP

Broken Object Property Level Authorization

Two failures in one. Send an extra property and grant yourself privilege. Read a response that returns properties it never should.

- Authorization is checked at the object level, never per property, so the API trusts every field you send and returns every field it holds.
- Mass assignment: inject a field like role or assigned_satellites and you write yourself privilege the interface never offered.
- Excessive data exposure: the same flaw in reverse, where telemetry responses leak command tokens, key IDs, and station credentials that should never leave the server.

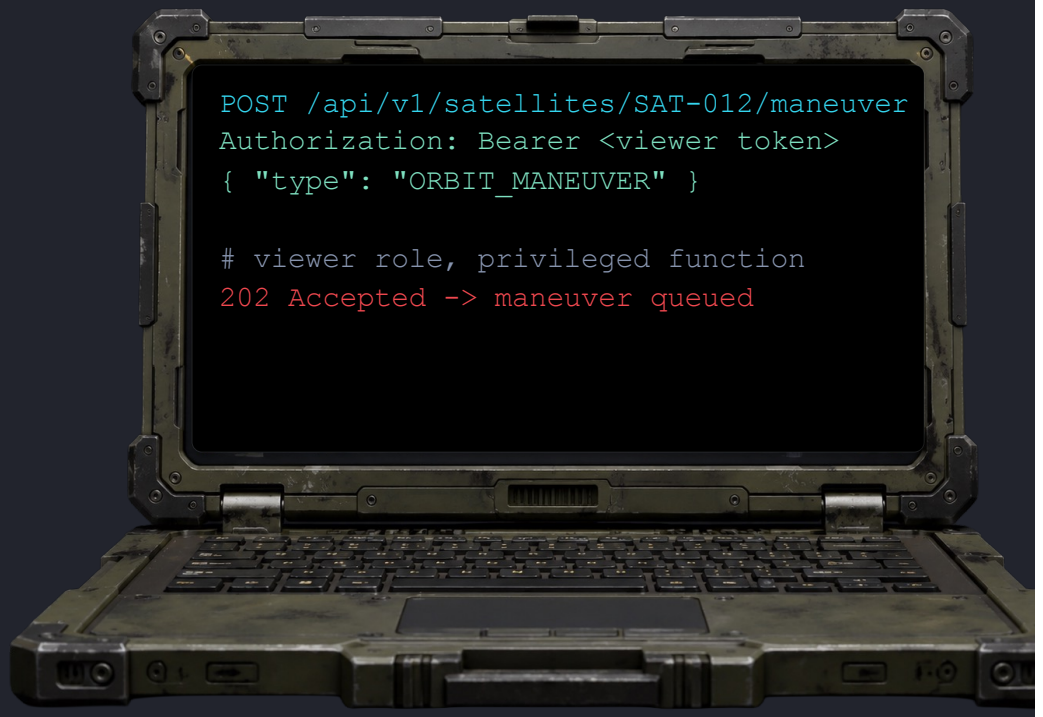


SECTION 02 · THE EXPOSURE GAP

Broken Function Level Authorization

A read-only account calls a privileged function. Execute a maneuver. Push firmware. The endpoint checks that you are authenticated, never that you are allowed.

- The endpoint verifies that you are authenticated but never that your role is allowed to call this function, so any valid account can reach privileged operations.
- Vertical privilege escalation: a viewer or read-only token invokes a high-privilege route it was never shown, often just by guessing the path or method (POST /maneuver, PUT /firmware).
- In orbit that means a low-privilege account can execute a maneuver, push firmware, or take a satellite out of safe mode, actions that should be gated to flight operators alone.

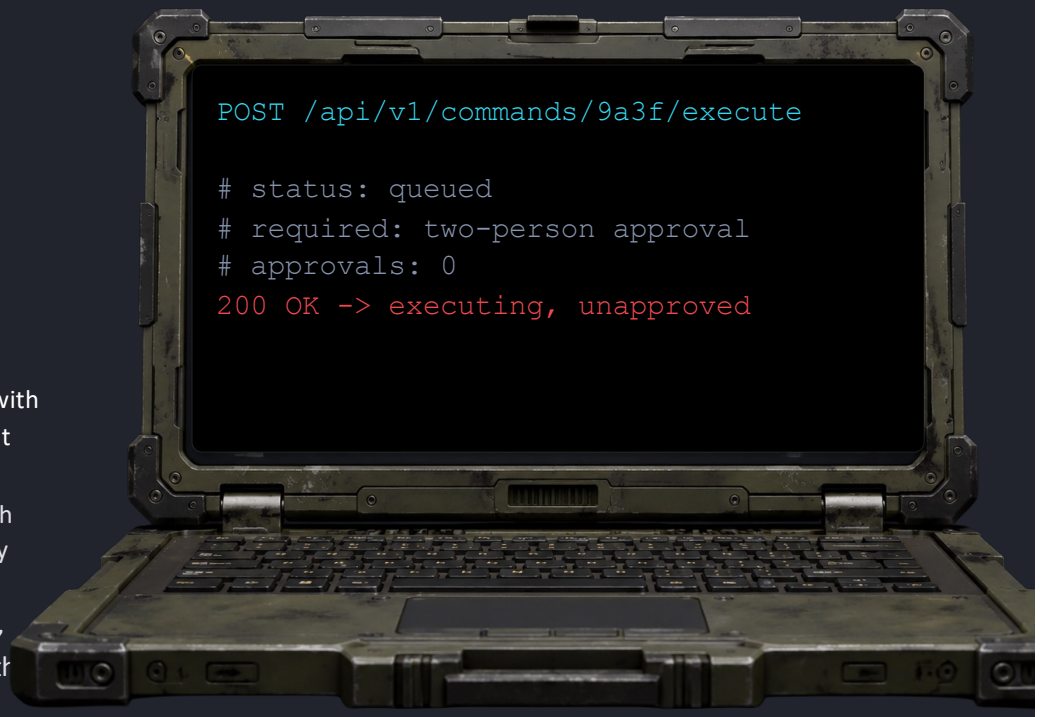


SECTION 02 · THE EXPOSURE GAP

Unrestricted Access to Sensitive Business Flows

Destructive commands are meant to require two-person integrity: queue, a second approver, then execute. The API lets you self-approve, or skip straight to execute.

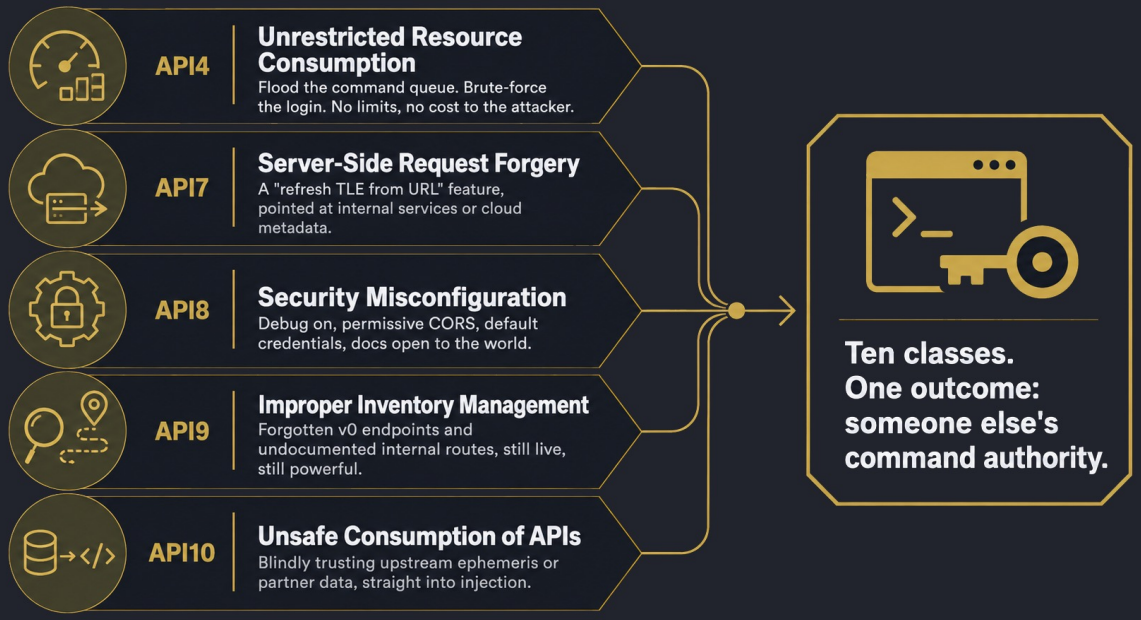
- The flaw lives in the workflow, not a single endpoint. A sensitive multi-step process, queue then second approval then execute, is exposed step by step with nothing on the server enforcing that the steps happen in order or by different people.
- **Skip, reorder, or replay the steps:** self-approve your own command, approve with the same identity that submitted it, or call POST /commands/{id}/execute directly and bypass approval entirely.
- Two-person integrity is the last human guardrail on destructive commands, and this collapses it, so a maneuver, deorbit, or payload shutdown fires with zero second-party review.



SECTION 02 · THE EXPOSURE GAP

The rest of the surface

Every one of these is a request. Break the request, and you have the bird.



SECTION 03

The Kill Chain

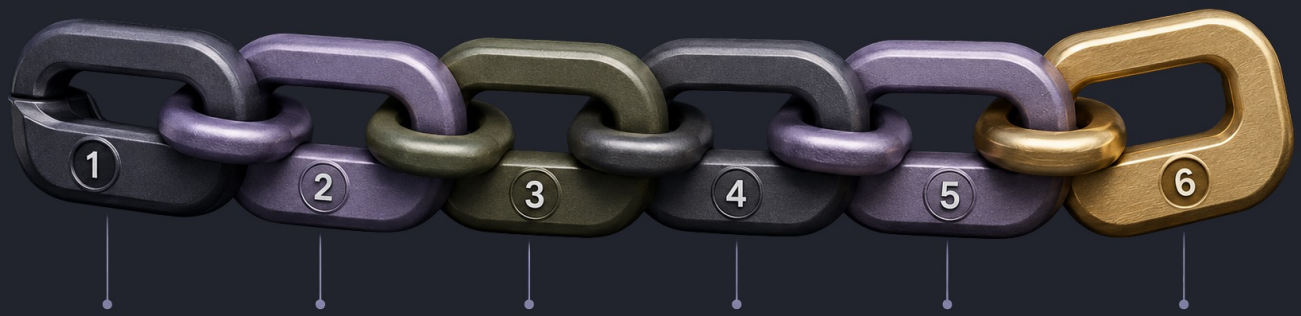
One flaw is a finding. Chained together, they become command authority over a constellation.



SECTION 03 · THE KILL CHAIN

The rest of the surface

One endpoint to the whole constellation



1	2	3	4	5	6
Recon	**Bypass Auth**	**Enumerate**	**Escalate**	**Break Integrity**	**Execute**
API9 • API8	API2	API1 • API3	API3 • API5	API6	
Find shadow and undocumented APIs	Forge or weaken the token	BOLA the fleet, harvest secrets	Grant scope, call privileged functions	Skip two-person approval	Command a bird you never owned

SECTION 03 · THE KILL CHAIN

The rest of the surface

What command authority actually costs



SECTION 04 – THE DEFENSE

Live Demonstration: Hacking Satellite Constellations

https://ares.assailai.com



SECTION 05

Defense in Depth

Modern satellite operations run through the ground segment. The ground segment runs on APIs. That is where the door is, and that is where we knocked.



THE DEFENSE

Fix the fundamentals first



Authorize every object and function

Check ownership and role on every request. Kills BOLA and BFLA.

Strong, short-lived, signed tokens

Proper signing, real expiry, mTLS. Kills broken authentication.

Least privilege, no mass assignment

Bind only allowed fields. Minimize telemetry. Kills BOPLA.

Two-person integrity, server-side

Never trust the client to say "approved." Kills business-flow abuse.

Inventory and retire shadow APIs

Know every endpoint. Turn off v0 and internal routes.

Limits, egress control, hard config

Rate limits, SSRF egress rules, no debug, no defaults.

THE DEFENSE

Then defend the architecture

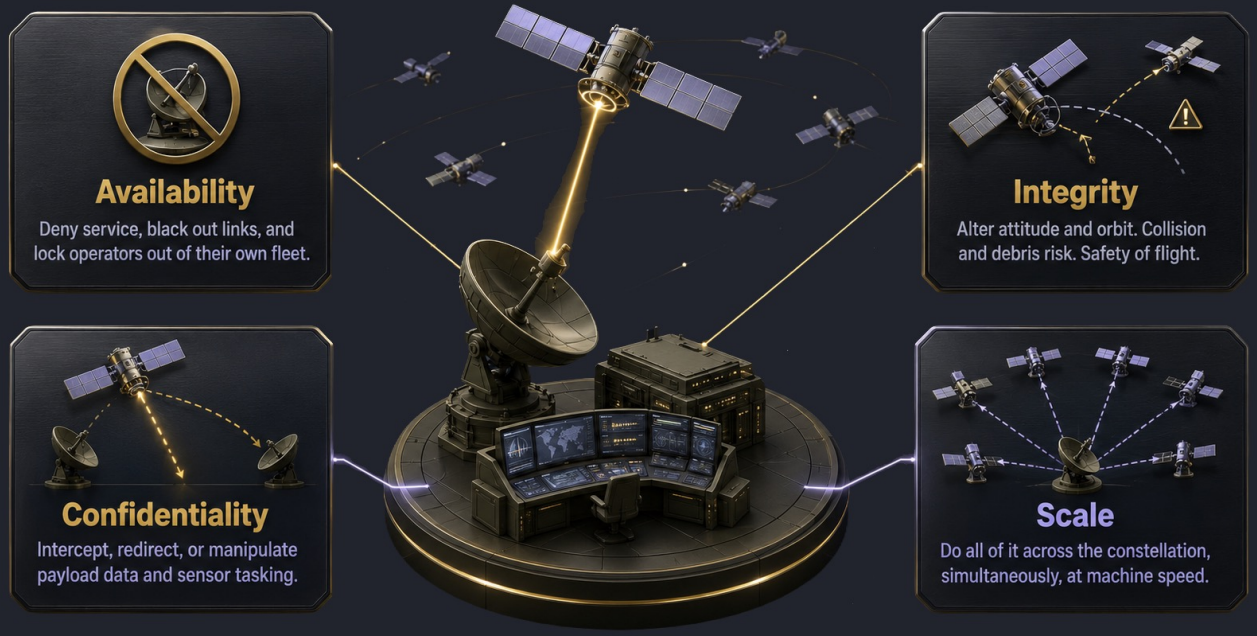


1. Segment and tenant isolation
2. Authenticate and encrypt commands at the bus, not just at the API
3. A safe mode that cannot be disabled remotely
4. Out-of-band authority for destructive commands
5. Anomaly detection on the command stream itself

SECTION 05 – THE DEFENSE

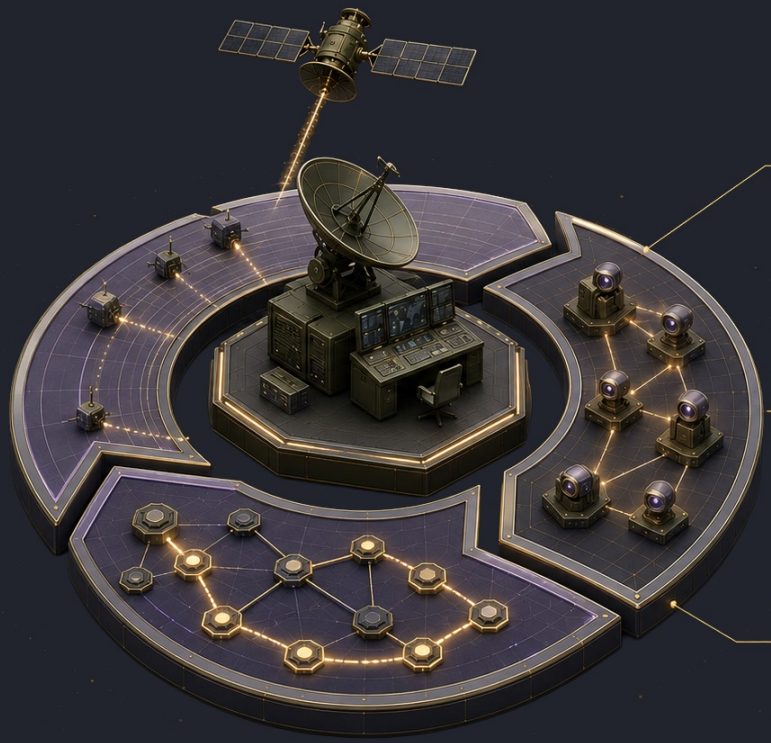
The rest of the surface

What command authority actually costs



SECTION 05 – THE DEFENSE

The rest of the surface



Validate continuously

Run autonomous offensive testing against your own ground segment, not once a year, but always.

Assume breach

Instrument for automated behavior and full chains, not just isolated human patterns.

Red-team like an AI will

Hunt for chains, not single findings. That is how the attacker will come.

SECTION 05 – THE DEFENSE

Summary

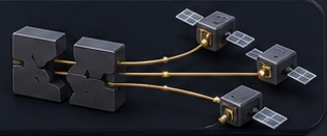
1. You do not hack the bird. You hack the ground-segment API.



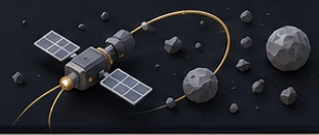
2. The harness carries the capability. The model is swappable.



3. BOLA plus a broken business flow equals fleet-wide command authority.



4. In space, the consequences are physical and often irreversible.



5. Defend at machine speed, or do not bother.



assail

Setup a Meeting

Put the swarm on your targets — and see what it proves in the first 24 hours.

Book a live attack demo

ALISSA.KNIGHT@ASSAILAI.COM WWW.ASSAILAI.COM