Cloudflare API Shield

Schema Validation + Endpoint Protection

Enterprise Sales Training • Brandon Crowe

What Is API Shield?

- Protects your server-side APIs by validating every incoming request against a schema you define
- Discovers API endpoints automatically using ML-based traffic analysis
- Schema validation ensures only properly formatted requests reach your application
- BOLA detection catches authorization abuse — someone changing an ID in the URL to access another user's data
- Sequence analytics monitors API call patterns to detect unusual behavior
- Works alongside WAF, Rate Limiting, and mTLS as part of Cloudflare's layered security model

Why API Shield Matters (The Business Case)

- Most application-layer traffic now comes through APIs, not web pages — APIs are the #1 attack surface
- Traditional WAFs catch known attack signatures (SQLi, XSS) but can't validate your specific API structure
- Schema validation is like a bouncer checking IDs against a guest list format, not just checking if someone showed up
- API discovery finds endpoints you didn't know existed — shadow APIs are a top security risk
- One dashboard to manage endpoints, schemas, sequences, and security events

How It Works — The Full Lifecycle

1. You build your APIs — Workers, Express, FastAPI, Django, or any framework that handles HTTP requests

2. Cloudflare discovers them — ML watches your traffic and identifies endpoints automatically

3. You write an OpenAPI spec — a YAML file that defines what each endpoint accepts (methods, fields, types, limits)

4. You upload the schema — API Shield parses it and maps rules to each endpoint

5. Cloudflare enforces it — Log mode records violations, Block mode rejects them at the edge

6. Ongoing monitoring — Security Events shows every non-compliant request with full details

Schema Validation — In Plain Terms

Your /api/guestbook endpoint expects a JSON body with a name and message field. Schema validation tells Cloudflare:

> *"This endpoint only accepts POST requests with a JSON body that has a name field (string, max 100 chars) and a message field (string, max 1000 chars). Nothing else."*

- Someone sends {"admin": true, "deleteAll": true} — BLOCKED. Extra fields not in the schema.
- Someone sends a 50MB binary file — BLOCKED. Wrong content type.
- Someone sends {"name": "Jane", "message": "Hello"} — ALLOWED. Matches the schema exactly.
- additionalProperties: false in the spec means ONLY the fields you define are accepted

BOLA Detection — Broken Object Level Authorization

BOLA is the #1 API vulnerability on the OWASP API Security Top 10. Here's how it works:

- Your API has /api/users/123/records that returns user 123's data
- An attacker changes the URL to /api/users/456/records
- If the API returns user 456's data without checking authorization — that's a BOLA vulnerability
- Cloudflare watches for this pattern: one user cycling through different IDs to access other people's data
- Flags it in Security Events before data leaks

Real-World Use Cases

Healthcare — Hospital patient portal APIs handling medical records, appointment scheduling, prescription data. Schema validation ensures only properly structured requests hit the API. BOLA detection prevents patients from accessing other patients' records by changing an ID.

Financial Services — Banking APIs for account balances, transfers, transaction history. Schema validation blocks malformed requests. BOLA detection prevents account enumeration attacks.

Government / Corrections — Inmate management systems, visitor scheduling, commissary account APIs. All handling extremely sensitive PII. BOLA protection prevents someone from changing an inmate ID in the URL and pulling another person's records.

E-Commerce / Retail — Product APIs, checkout flows, user account endpoints. Schema validation prevents injection attacks on search and cart APIs. Rate limiting + API Shield together prevent inventory scraping.

SaaS Platforms — Multi-tenant APIs where one customer should never access another's data. BOLA detection is critical for tenant isolation.

What I Built — API Shield on saltwaterbrc.com

- Live at saltwaterbrc.com — 3 Worker APIs protected by API Shield
- AI Agent (POST /agents/salt-water-agent/{sessionId}) — multi-turn chat with tool calling
- Guestbook (POST /api/guestbook) — accepts name, company, message via D1
- Visitor Counter (GET /?action=stats) — page analytics via Durable Objects
- Wrote openapi.yaml defining all 3 endpoints with field types, limits, and additionalProperties: false
- Uploaded schema to API Shield — Cloudflare created 18 endpoint entries across 3 origins
- Set to Log mode for monitoring, one click to switch to Block
- Fired good and bad requests from browser console to test — both logged correctly

5-Minute Live Demo

1. Open saltwaterbrc.com — navigate to any page

2. Open browser console — Cmd+Option+J on Mac, F12 on Windows

3. Fire a good request:

```
fetch('/api/guestbook', { method: 'POST', headers: { 'Content-Type': 'application/json' }, body: JSON.stringify({ name: 'Demo', message: 'Valid request' }) }).then(r => r.json()).then(console.log);
```

Result: { success: true, id: 6 } — 201 Created

4. Fire a bad request:

```
fetch('/api/guestbook', { method: 'POST', headers: { 'Content-Type': 'application/json' }, body: JSON.stringify({ admin: true, deleteAll: true, escalatePrivileges: 'root' }) }).then(r => r.json()).then(console.log);
```

Result: { error: "name and message are required" } — 400 Error

5. Open Cloudflare Dashboard — Security > Analytics > Events shows the flagged request. Security > Web Assets > Schema Validation shows non-compliant count.

The Layered Defense Model

DDoS Protection > WAF Managed Rules > API Shield > Rate Limiting > Your Code

Layer	What It Catches	Example
DDoS Protection	Volumetric attacks	10M requests/sec flood
WAF Managed Rules	Known attack signatures	/.env probes, SQLi, XSS

API Shield	Schema violations, BOLA	Extra fields, ID manipulation
Rate Limiting	Brute force / abuse	1000 requests/min from one IP
Your Worker Code	App-level validation	Missing required fields

Sales Talking Points

- "Most app traffic is now API traffic. If you're only protecting web pages, you're missing the majority of your attack surface."
- "API Shield discovers endpoints you didn't know existed. Shadow APIs are the #1 risk."
- "Schema validation is the bouncer at the door. Only properly formatted requests get through."
- "We set this up on a live site in 20 minutes. I can show you it working right now."
- "WAF catches the known attacks. API Shield catches the application-specific ones. Together, that's layered defense."