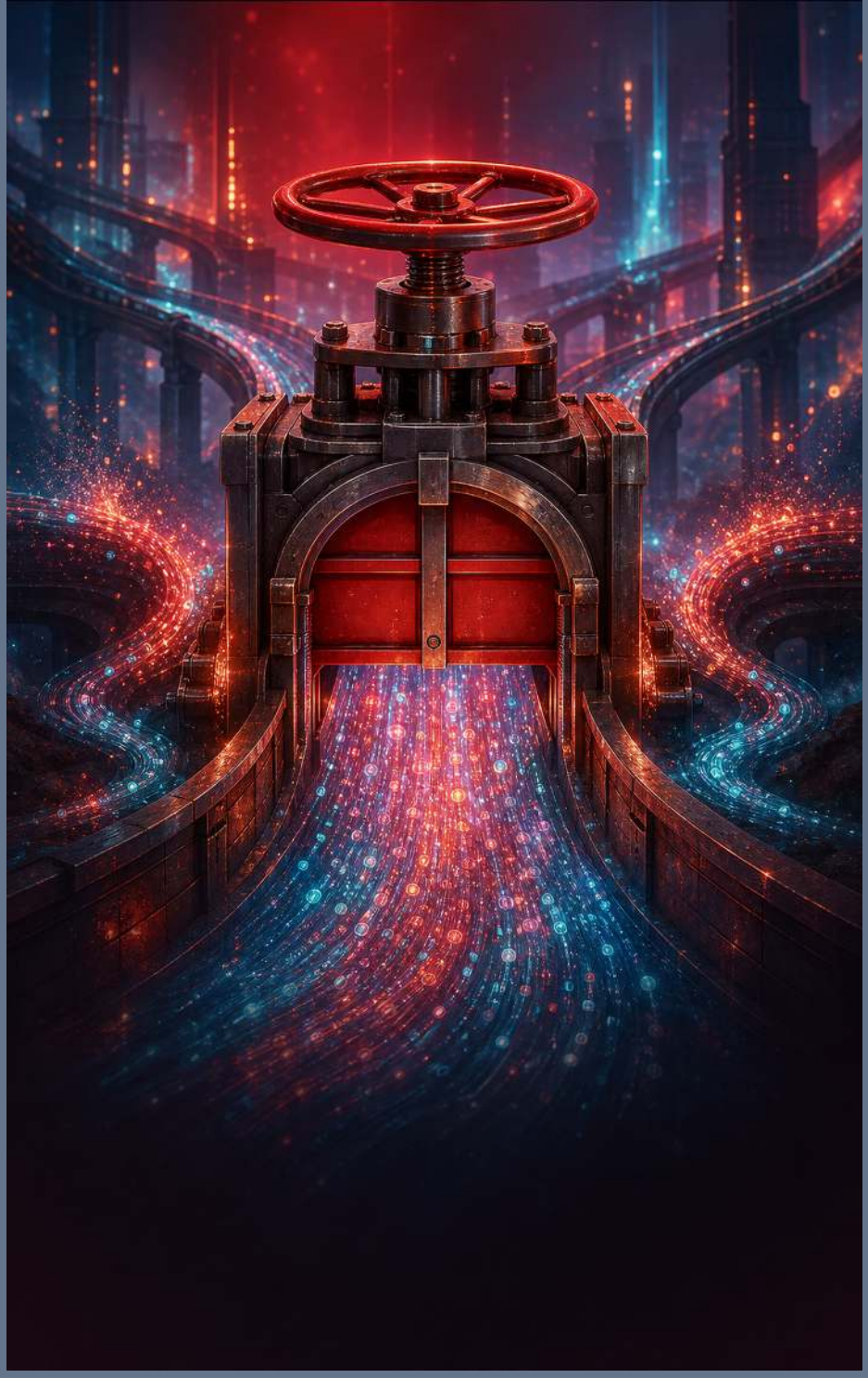


Rate Limiting & Abuse Prevention

For AI API Platforms

Pragma.Vision

Rate Limiting & Abuse Prevention

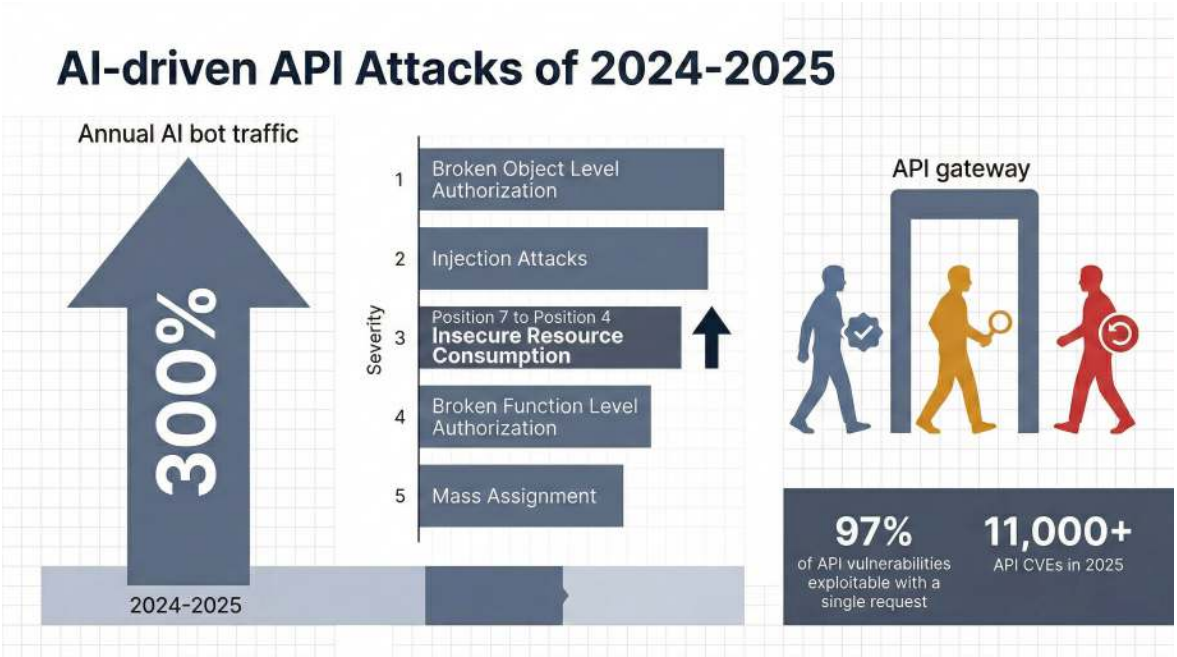
Published by Pragma.Vision

First edition, 2026.

Contents

1

AI Agents Are the New Attack Surface



A digital fortress surrounded by waves of automated traffic — some legitimate agent requests rendered in blue, others malicious in red, converging on API gateway endpoints

Your API is not being attacked by humans anymore. It is being attacked by other AI agents.

In 2025, AI-powered bot traffic increased by 300% in a single year, and 97% of documented API vulnerabilities could be exploited with a single request. The attackers are not teenagers with scripts — they are autonomous agents that rotate IPs, mimic legitimate traffic patterns, and operate at machine speed around the clock. Your rate limiter from 2022 was designed for human-speed abuse. It is now facing machine-speed adversaries, and it is losing.

300%

increase in AI-powered bot traffic in 2025 alone[1]

The traditional rate limiting playbook — count requests per IP, block after a threshold — was adequate when abuse came from predictable sources at predictable speeds. That era is over. AI agents create new categories of abuse that traditional rate limiters cannot detect: agents that stay just under your limits while scraping your entire catalog, agents that replay captured authentication tokens across thousands of IPs, agents that inject prompts through your API to extract internal system information.

This book is a practical engineering guide. Every chapter contains production TypeScript code you can deploy to Cloudflare Workers today. By the end, you will have a complete rate limiting and abuse prevention system that handles per-user limits, per-IP limits, per-agent limits, brute force protection, replay attack prevention, and agent-specific abuse detection — all running at the edge with sub-millisecond overhead.

[1] Imperva, "Bad Bot Report," 2025; see also Akamai and Cloudflare threat reports.

1.1 The Scale of the Problem

The numbers tell a clear story. According to the 2026 API ThreatStats Report, APIs accounted for 17% of all published security bulletins in 2025 — over 11,000 vulnerabilities. Of those, 98% were classified as easy or trivial to exploit, and 59% required no authentication at all. Insecure Resource Consumption — the category that includes rate limiting failures — rose from seventh place in 2024 to fourth in 2025, driven by automated scraping, enumeration, and denial-of-service attacks.

Key Insight

The most dangerous API abuse is not the dramatic DDoS attack that takes your service offline. It is the low-and-slow scraping agent that stays just under your rate limits for weeks, extracting your product catalog, pricing data, or user information one request at a time. Rate limiting is not just about blocking floods — it is about detecting and throttling persistent, intelligent adversaries.

For AI API platforms specifically, the attack surface is uniquely large. Every endpoint that accepts natural language input is a potential prompt injection vector. Every endpoint that returns structured data is a scraping target. Every authentication endpoint is a credential stuffing target. And because your legitimate users are themselves AI agents, distinguishing good bots from bad bots is fundamentally harder than distinguishing humans from bots.

1.2 About Pragma.Vision

This book is drawn from the production security architecture of the Pragma.Vision ecosystem — an AI-native commerce platform where nine interconnected platforms share infrastructure, identity, and payment systems through the soft.house developer portal. Every rate limiting pattern in this book runs in production on Cloudflare Workers, protecting real API endpoints that process real transactions. The specific limits

described here — 100 requests per minute per user, 5 login attempts per 15 minutes per IP, 10 mandate creations per hour per user — are the actual limits enforced across the ecosystem.

Pragma.Vision uses a three-layer protocol architecture: identity verification (Visa TAP), user authorization (Google AP2 with cryptographic mandates), and payment execution (Stripe ACP). Rate limiting operates across all three layers, with different strategies for each. The nonce-based replay protection described in Chapter 6 integrates directly with the mandate authorization system that secures every financial transaction.

1.3 What You Will Build

By the end of this book, you will have:

1. A **sliding window rate limiter** using Cloudflare KV and Durable Objects, with configurable windows from seconds to hours (Chapter 2).
2. A **multi-dimensional limit system** that enforces per-user, per-IP, and per-agent limits simultaneously with different thresholds for each (Chapter 3).
3. A **brute force protection layer** with progressive penalties, account lockout, and credential stuffing detection (Chapter 4).
4. An **agent-specific abuse detection system** that identifies scraping, replay attacks, and prompt injection attempts through API endpoints (Chapter 5).
5. A **nonce-based replay protection service** with server-side nonce tracking and time-bounded validity windows (Chapter 6).
6. A **complete production rate limiter** that integrates all components into a single Cloudflare Worker middleware (Chapter 7).

7. A **monitoring and operational playbook system** with structured telemetry, alert rules, incident response procedures, and continuous calibration processes (Chapter 8).

Every component is TypeScript, deployable to Cloudflare Workers, and designed to operate within free-tier limits (100,000 requests per day, 100,000 KV reads per day). You do not need to spend money to deploy production-grade rate limiting.

DEMO

This is a free preview of the full edition.

Get the complete book at:

https://pragmavision.lemonsqueezy.com/