

Rate Limiting & Abuse Prevention

For AI API Platforms

Pragma.Vision

Rate Limiting & Abuse Prevention

© 2026 Pragma.Vision. All rights reserved.

Trademark Notice

Google, Google Pay, Google Cloud, and Android are trademarks of Google LLC. Stripe is a trademark of Stripe, Inc. Cloudflare and Cloudflare Workers are trademarks of Cloudflare, Inc. Supabase is a trademark of Supabase, Inc. OpenAI and ChatGPT are trademarks of OpenAI, Inc. Claude is a trademark of Anthropic, PBC. W3C is a trademark of the World Wide Web Consortium. Visa is a trademark of Visa International Service Association. OWASP is a trademark of the OWASP Foundation. Midjourney is a trademark of Midjourney, Inc. Canva is a trademark of Canva Pty Ltd. Etsy is a trademark of Etsy, Inc. Amazon is a trademark of Amazon.com, Inc. All other trademarks are the property of their respective owners.

No Affiliation

This book is an independent publication. It is not authorized, sponsored, or endorsed by any of the companies or organizations whose products or services are mentioned herein.

No Professional Advice

The information in this book is provided for educational purposes only. It does not constitute legal, financial, investment, tax, or other professional advice. Readers should consult qualified professionals for guidance specific to their situation.

Code Examples

Code examples in this book are provided for illustration only. They may not be suitable for production use without additional validation, error handling, and security review.

Published by Pragma.Vision

First edition, 2026.

Contents

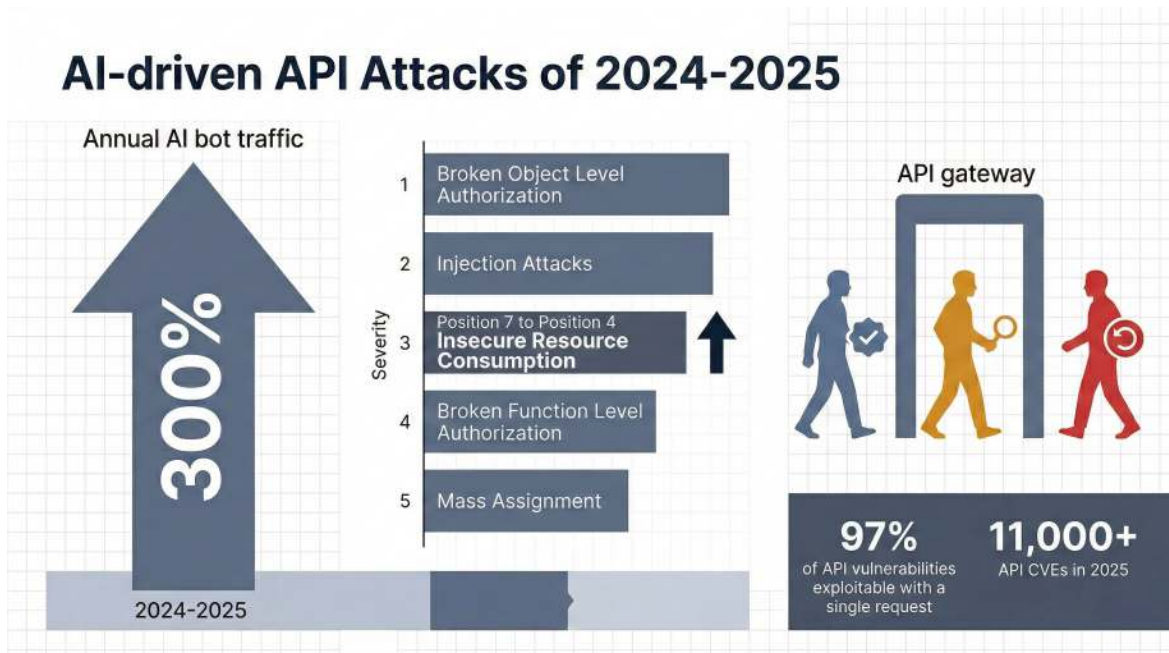
1	AI Agents Are the New Attack Surface	6
1.1	The Scale of the Problem	8
1.2	About Pragma.Vision	8
1.3	What You Will Build	9
2	Sliding Window Rate Limiting with Cloudflare KV	11
2.1	Why Sliding Window Beats Fixed Window	12
2.2	The Sliding Window Counter Algorithm	13
2.3	Implementing on Cloudflare KV	15
2.4	Handling KV Eventual Consistency	18
2.5	Durable Objects for Strong Consistency	18
2.6	Standard Rate Limit Headers	21
3	Per-User, Per-IP, Per-Agent Limit Design	24
3.1	The Three Dimensions	25
3.2	Production Limit Configuration	25
3.3	Multi-Dimensional Rate Limit Middleware	26
3.4	Operation Classification	30
3.5	Adaptive Limits for Trusted Agents	31
3.6	Rate Limit Response Design	34
4	Brute Force and Credential Stuffing Prevention	36
4.1	Progressive Penalty Architecture	38
4.2	Credential Stuffing Detection	40
4.3	IP Reputation and Block Lists	43

4.4	Account Lockout with Self-Service Recovery	45
5	Agent-Specific Abuse Patterns	47
5.1	Pattern 1: Intelligent Catalog Scraping	48
5.2	Pattern 2: Replay Attacks via Captured Tokens	50
5.3	Pattern 3: Prompt Injection via API	52
5.4	Pattern 4: API Key Enumeration	55
5.5	Pattern 5: Resource Exhaustion Through Expensive Queries	56
6	Nonce-Based Replay Protection	58
6.1	What Is a Nonce and Why It Matters	59
6.2	The Nonce Verification Flow	59
6.3	Nonce Storage Strategy	63
6.4	Integrating Nonces with Mandate Authorization	65
6.5	Clock Synchronization Considerations	67
7	Production Rate Limiter: Complete Implementation	69
7.1	Architecture Overview	70
7.2	The Complete Middleware	71
7.3	Wrangler Configuration	77
7.4	Observability and Monitoring	78
7.5	Testing Your Rate Limiter	80
7.6	Deployment Checklist	82
7.7	Closing Perspective	83
8	Monitoring, Alerting, and Operational Playbooks	85
8.1	The Four Pillars of Rate Limit Observability	86
8.2	Implementing Structured Telemetry	87
8.3	Alert Conditions and Escalation	89
8.4	Operational Playbooks	91
8.4.1	Playbook: Credential Stuffing Attack	91
8.4.2	Playbook: DDoS via API Flood	92

8.4.3	Playbook: Nonce Replay Surge	92
8.5	Capacity Planning for Rate Limit Infrastructure	93
8.6	Post-Incident Analysis Template	95
8.7	Continuous Calibration	97
8.8	Closing: Rate Limiting as a Living System	98
	What's Next	99
	About Pragma.Vision	101

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 alone¹

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.

¹Imperva, “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/>