PROGEOLAB

PROGEOLAB RESEARCH

WAF vs AI: Three Layers of Blocking

F5 BIG-IP, Cloudflare, and Akamai in the AI crawler era · April 2026

PROGEOLAB

April 2026

progeolab.ai/research

Contents

CHAPTER 1

Executive Summary

Web Application Firewalls are the invisible infrastructure that determines whether AI crawlers can access enterprise websites. This report presents the first WAF vendor attribution study across the Fortune Global 500, revealing that bot blocking operates at three distinct layers — User-Agent string, IP reputation, and TLS fingerprinting — each controlled by different WAF features and each requiring a different response.

Key Findings

F5 BIG-IP dominates the Fortune 500 WAF landscape with 147 detected deployments — more than Cloudflare (64), Akamai (55), and Imperva (20) combined. This finding contradicts the public narrative that centers Cloudflare in WAF discussions. F5's dominance reflects its position as the enterprise default for application delivery. (Chapter 3)

Three distinct blocking layers operate independently. Layer 1 (User-Agent string detection) affects 53 companies that block ChatGPT-User while serving Chrome. Layer 2 (IP reputation) affects 24 companies that block all traffic from datacenter IP ranges. Layer 3 (TLS fingerprinting) affects 15 companies that reject non-browser TLS handshakes before the HTTP request is read. (Chapter 2)

Cloudflare's TLS fingerprinting creates structural AI blocking. Cloudflare Bot Management inspects TLS ClientHello fingerprints (JA3/JA4) and rejects non-browser clients. This layer cannot be bypassed by changing the User-Agent header — it operates below HTTP. Companies on Cloudflare who want AI crawler access must explicitly configure exceptions. (Chapter 4)

No WAF vendor offers native AI crawler verification. Unlike Googlebot (which can be verified via reverse DNS), AI crawlers like GPTBot and ClaudeBot lack equivalent verification infrastructure integrated into WAF products. This forces enterprises into all-or-nothing decisions. (Chapter 5)

About This Study

WAF vendor attribution was performed by analyzing raw HTTP response bodies (challenge pages, error templates, server headers, JavaScript patterns) from the Chrome UA probe run across 500 Fortune Global 500 companies. Three-layer blocking classification was derived from cross-database comparison of four user agent runs (134,000 total probes).

CHAPTER 2

Three Layers of AI Bot Blocking

Enterprise websites do not block AI crawlers with a single mechanism. Our four-way user agent comparison reveals three distinct blocking layers, each operating at a different point in the network stack.

Layer 1: User-Agent String Detection — 53 Companies

The most deliberate and reversible form of blocking. The WAF inspects the User-Agent header and applies different rules to AI-identified requests versus browser requests. The same IP, same TLS handshake, same HTTP request — only the UA string differs — and the server returns 200 to Chrome but 403 to ChatGPT-User.

This is a policy decision encoded in WAF rules. Companies like Johnson & Johnson (Chrome 64/64, ChatGPT 0/64), AstraZeneca (Chrome 45/64, ChatGPT 0/64), and Salesforce (Chrome 13/64, ChatGPT 0/64) have WAF configurations that explicitly match AI bot identifiers and reject them.

Layer 1 blocking is the easiest to implement and the easiest to reverse. A WAF administrator can add or remove a User-Agent rule in minutes.

Layer 2: IP Reputation — 24 Companies

A deeper form of blocking that operates below the UA string. These 24 companies return 403 or connection errors to all four user agents, including Chrome. The blocking decision is based on the source IP address, not the request content.

Enterprise WAFs maintain databases of known datacenter IP ranges. Requests from AWS, Azure, GCP, Hetzner, and other cloud providers are flagged as automated regardless of User-Agent. Our probes ran from a datacenter, triggering this classification.

Tesla, Home Depot, Allianz, TSMC, and Lufthansa exemplify Layer 2 blocking. These sites would likely be accessible from a residential IP. The blocking targets the infrastructure, not the identity.

For AI crawlers, Layer 2 is a structural challenge: OpenAI runs from Azure, Anthropic from AWS. Both face datacenter IP reputation scoring by default.

Layer 3: TLS Fingerprinting — 15 Companies

The most technically sophisticated layer. These companies terminate connections during the TLS handshake, before the HTTP request (including the User-Agent header) is transmitted.

Our probe tool (httpx) generates a TLS ClientHello with a JA3 fingerprint distinct from any real browser. Modern WAFs — Cloudflare Bot Management, Akamai Bot Manager, F5 Shape Security — compare TLS fingerprints against known browser databases. When the JA3 hash does not match Chrome, Firefox, or Safari, the connection is reset.

Evidence: 666 probe responses across 15 companies returned HTTP/2 StreamReset errors in the Chrome UA run. This error occurs after TLS establishment but before HTTP exchange — the signature of fingerprint-based rejection.

No User-Agent change bypasses Layer 3. Only a real browser (or a tool replicating browser TLS behavior) passes this check.

Combined Impact

Layer	Mechanism	Companies	Reversibility
Layer 1	User-Agent string rules	53	Easy — WAF rule change
Layer 2	IP reputation scoring	24	Moderate — requires IP allowlisting
Layer 3	TLS fingerprint matching	15	Hard — requires browser-compatible TLS

The three layers are not mutually exclusive. Some companies deploy multiple layers: Layer 1 + Layer 2 (~6 companies), or all three layers (~3 companies with the hardest-to-penetrate configurations).

> ***Source data:*** *Layer 1 from cross-database comparison (Chrome 2xx > 0 AND ChatGPT 2xx = 0). Layer 2 from companies with 403/error across ALL four UAs. Layer 3 from StreamReset error patterns in Chrome UA run. Full methodology in the companion report "53 Companies Invisible to ChatGPT."*

CHAPTER 3

The WAF Vendor Landscape

F5 BIG-IP: The Enterprise Default (147 Sites)

F5 BIG-IP dominates the Fortune 500 WAF landscape at 147 detected deployments. This is more than Cloudflare, Akamai, and Imperva combined. The finding contradicts the developer-centric narrative that positions Cloudflare as the dominant WAF provider.

F5's dominance reflects its position in traditional enterprise IT infrastructure. Fortune 500 companies — banks, insurers, energy companies, manufacturers — have long-standing relationships with F5 for load balancing and application delivery. WAF functionality is an add-on to existing F5 BIG-IP infrastructure, not a standalone purchase decision.

For AI visibility, F5 deployments show the widest range of outcomes. Some F5 sites are fully transparent (pass all traffic), while others implement aggressive bot management through F5 Shape Security. The variability is a configuration choice, not a vendor limitation — F5 gives administrators the most granular control over bot policy.

Cloudflare (64 Sites)

Cloudflare appears on 64 Fortune 500 sites and presents the most consistent AI-blocking behavior. Cloudflare's Bot Management product combines IP reputation, TLS fingerprinting (JA3/JA4), and JavaScript challenges.

Cloudflare's challenge pages are distinctive: the "Just a moment..." interstitial, `cf-ray` headers, and `challenge-platform` JavaScript. These markers made Cloudflare the most reliably detectable WAF in our analysis.

For AI crawlers, Cloudflare presents a particular challenge at Layer 3. Even a Chrome User-Agent string is insufficient if the TLS ClientHello does not match a known browser fingerprint. This makes Cloudflare the most structurally difficult WAF for AI crawler access.

Akamai (55 Sites)

Akamai appears on 55 sites, primarily large consumer brands, retailers, and financial institutions. Akamai Bot Manager integrates with their CDN and uses behavioral analysis alongside traditional signals. Identifiable by `AkamaiGHost` headers and `Reference #` error codes.

Imperva (20 Sites)

Imperva appears on 20 sites, identifiable by `_imp_apg_r_` cookies and characteristic challenge patterns. Imperva's Advanced Bot Protection uses device fingerprinting and behavioral analysis.

AWS WAF, PerimeterX, and Others

AWS WAF (3 sites), PerimeterX (4 sites), and Sucuri (1 site) represent the long tail. Six sites showed generic blocking patterns that could not be attributed to a specific vendor.

WAF Vendor Distribution

WAF Vendor	Fortune 500 Sites	Market Share
F5 BIG-IP	147	49.0%
Cloudflare	64	21.3%
Akamai	55	18.3%
Imperva	20	6.7%
Generic/Unidentified	6	2.0%
PerimeterX	4	1.3%
AWS WAF	3	1.0%
Sucuri	1	0.3%
Total detected	**300**	**100%**

Source data: *WAF vendor attribution from raw HTTP response body analysis of the Chrome UA run. Detection based on signature matching in response HTML, JavaScript challenge patterns, HTTP headers, and cookie names. 300 of 388 responding companies had detectable WAF signatures.*

CHAPTER 4

Cloudflare's TLS Fingerprinting — The Structural Challenge

Cloudflare's Bot Management product represents the most technically challenging barrier to AI crawler access among Fortune 500 WAF deployments. Understanding why requires examining how TLS fingerprinting works and why it cannot be bypassed at the HTTP layer.

How TLS Fingerprinting Works

When a client connects to a server over HTTPS, the first message is the TLS ClientHello. This message contains:

- Supported cipher suites (and their order)
- Supported TLS extensions
- Supported elliptic curves
- Supported signature algorithms
- Maximum fragment length
- ALPN protocol list

Each combination of these parameters produces a unique fingerprint. The JA3 algorithm (developed by Salesforce's threat team) hashes these parameters into a 32-character string. The JA4 algorithm (developed by FoxIO) extends this with additional context.

Every HTTP client — Chrome, Firefox, Safari, curl, httpx, Python requests — produces a distinctive TLS fingerprint. Cloudflare maintains a database of fingerprints for known browsers and compares incoming connections against this database.

Why This Blocks AI Crawlers

AI crawlers (GPTBot, ClaudeBot, PerplexityBot) use standard HTTP libraries — not full browsers. These libraries produce TLS fingerprints that differ from Chrome, Firefox, and Safari. Cloudflare's Bot Management detects this mismatch and blocks or challenges the connection.

The critical insight is that this detection happens during the TLS handshake — before the HTTP request is sent. The User-Agent header, the HTTP method, the URL path, and the request body are all invisible at this point. The server has already decided to block based solely on how the TLS connection was initiated.

This means: - Changing the User-Agent header has no effect - Modifying HTTP headers has no effect - Using HTTP/2 vs HTTP/1.1 has no effect (ALPN negotiation is part of TLS) - Only changing the TLS implementation itself can bypass this layer

Evidence from Our Data

In the Chrome UA run (using httpx with a Chrome User-Agent string), 15 companies returned HTTP/2 StreamReset errors. The error timing — after TLS negotiation but during HTTP/2 framing — is consistent with fingerprint-based rejection.

These 15 companies are disproportionately on Cloudflare. The StreamReset pattern is Cloudflare Bot Management's signature response to a non-browser TLS fingerprint: establish the TLS connection (to gather the fingerprint), begin HTTP/2 framing, then reset the stream when the fingerprint check fails.

Implications for AI Companies

For AI companies building crawlers, Cloudflare's TLS layer creates a binary choice:

1. **Use a real browser** (headless Chromium): produces authentic TLS fingerprints, bypasses Layer 3, but is expensive at scale (each page requires a full browser instance)
2. **Use TLS fingerprint spoofing libraries** (e.g., utls for Go, tls-client for Python): replicate browser TLS behavior without running a browser, lower cost, but requires ongoing maintenance as Cloudflare updates its fingerprint database
3. **Accept the block**: acknowledge that some sites behind Cloudflare Bot Management will be inaccessible and work with the content that is available

None of these options is ideal. The fundamental tension is that the web security industry treats all non-browser HTTP traffic as suspicious, while the AI industry needs non-browser HTTP access to build and operate AI systems.

> ***Source data:*** *TLS fingerprint analysis from the Chrome UA run. StreamReset errors identified in the probe results database (`error_detail LIKE '%StreamReset%'`). Cloudflare attribution from response body signatures. JA3 algorithm: https://github.com/salesforce/ja3.*

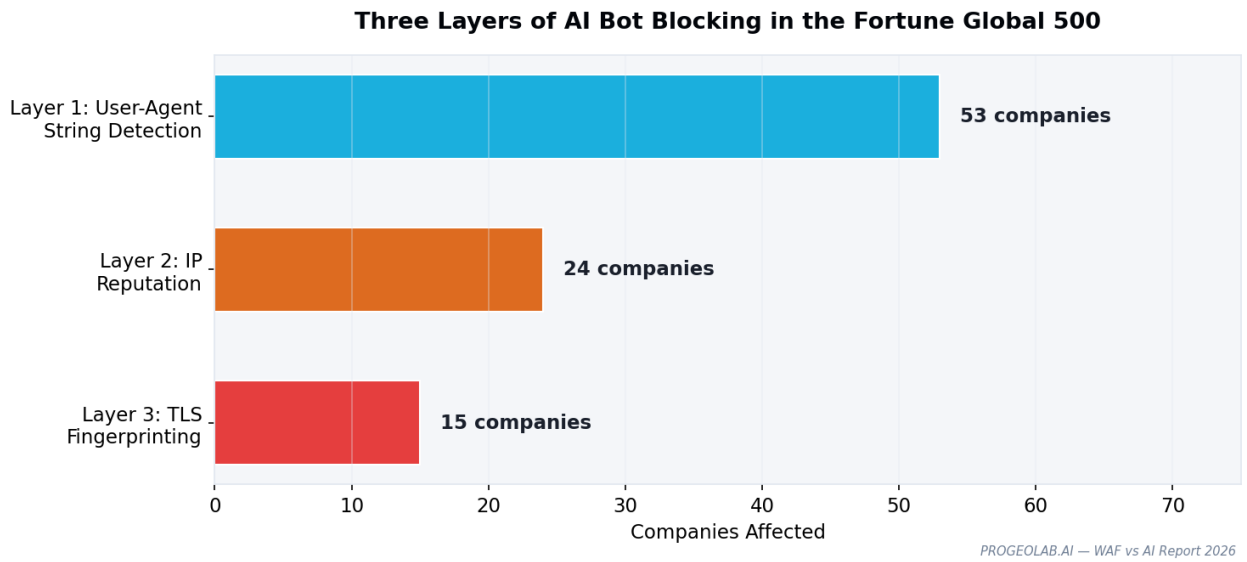

Figure 4.1 · Three layers

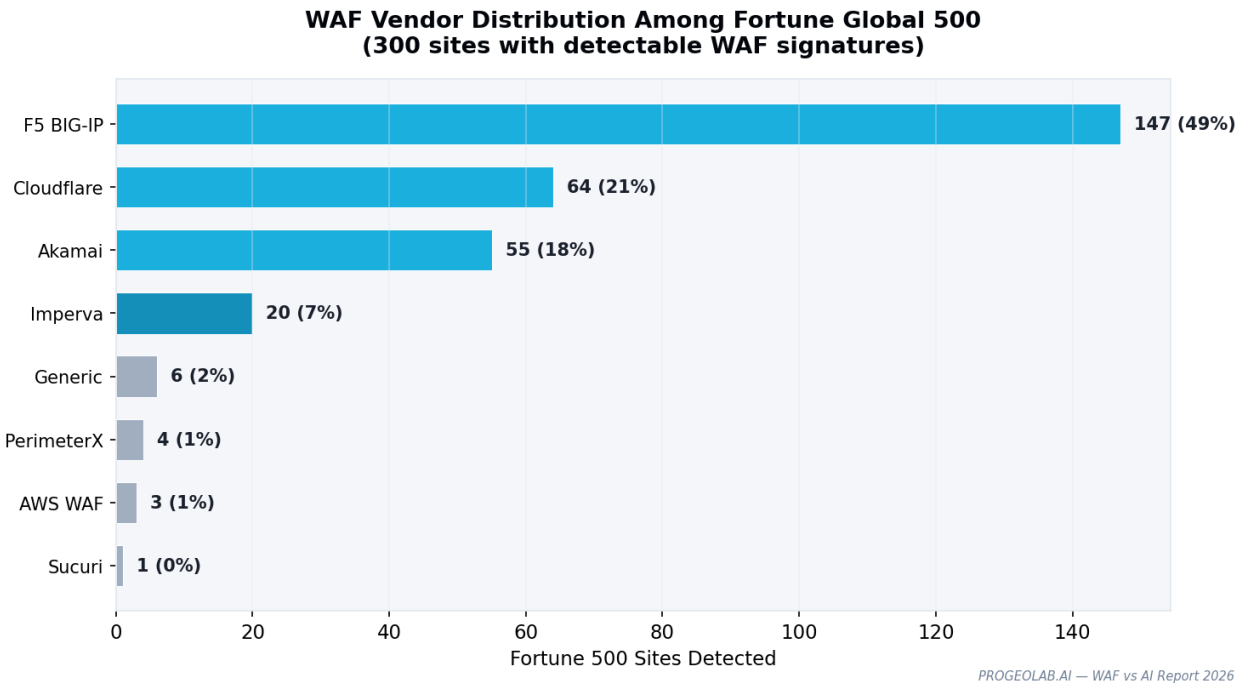

Figure 4.2 · Waf vendors

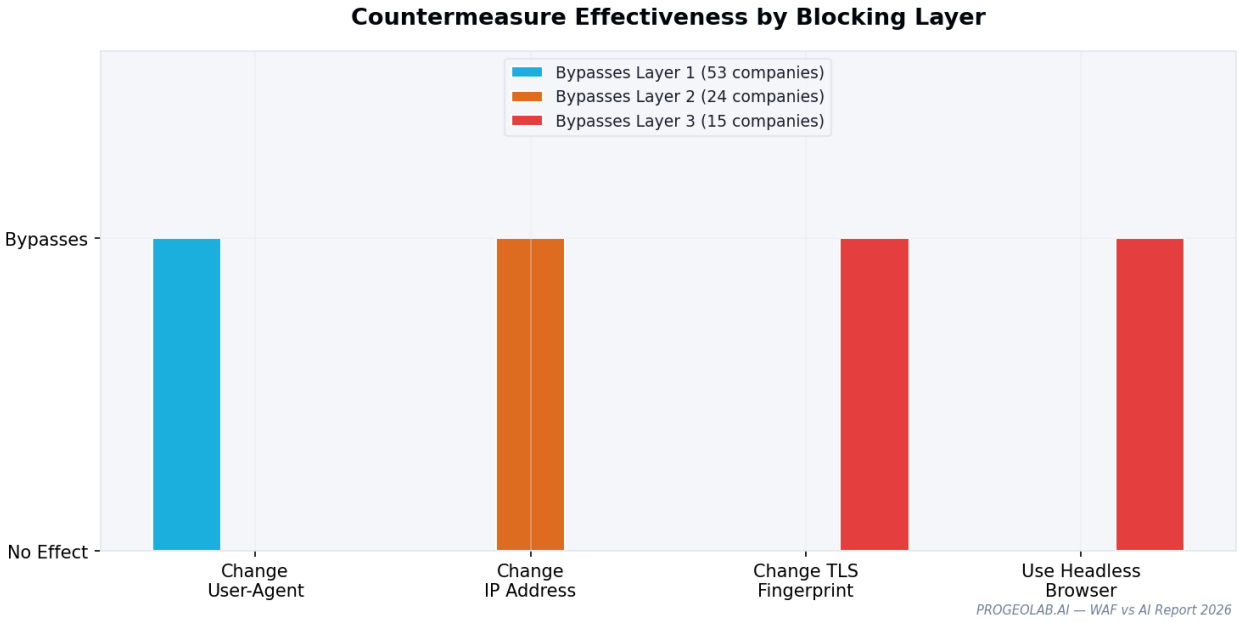

Figure 4.3 · Countermeasures

CHAPTER 5

The Verification Gap and Recommendations

The Missing Piece: AI Crawler Verification

The Googlebot impersonation finding from the companion report reveals a critical gap in the AI crawler ecosystem: there is no standardized way for enterprises to verify that a request claiming to be from GPTBot actually originates from OpenAI.

Google solved this problem years ago. Googlebot requests can be verified via reverse DNS: look up the IP address, confirm it resolves to `*.googlebot.com` or `*.google.com`, then forward-confirm the hostname resolves back to the same IP. Enterprise WAFs have built-in support for this verification.

AI crawlers publish CIDR ranges but lack equivalent reverse DNS infrastructure. When a WAF sees `User-Agent: GPTBot`, it cannot automatically verify the claim. This forces enterprises into a binary choice: allow all requests claiming to be GPTBot (including potential impersonators), or block all of them.

Recommendations for Enterprises

1. Audit your WAF's bot management rules. Check whether AI-specific user agents (ChatGPT-User, GPTBot, ClaudeBot) are in allow lists, block lists, or handled by default policies. Make a deliberate decision.

2. Distinguish between blocking layers. If your goal is to block AI training while allowing AI retrieval, Layer 1 (UA-based) rules are sufficient. If you are seeing Layer 2 or Layer 3 blocking, your security team's datacenter IP or TLS policies may be blocking AI crawlers without anyone realizing it.

3. Test from the AI crawler perspective. Use curl with each AI crawler's User-Agent string to verify your intended policy matches reality. Discrepancies between robots.txt declarations and actual WAF behavior are common (see companion report on robots.txt).

4. Coordinate security and marketing teams. WAF changes that affect bot management should be reviewed for AI visibility impact before deployment.

Recommendations for AI Companies

1. Implement reverse DNS verification for crawler IP addresses, following the Googlebot pattern.

2. Work with WAF vendors to integrate AI crawler verification into their products (F5, Cloudflare, Akamai, Imperva).

3. Address TLS fingerprinting by investing in crawler infrastructure that produces browser-compatible TLS fingerprints.

4. Publish transparent crawl behavior documentation — crawl rates, content preferences, robots.txt compliance, data retention.

Recommendations for WAF Vendors

1. Add AI crawler verification as a built-in feature, equivalent to existing Googlebot verification.

2. Create granular AI bot categories — distinguish training crawlers from retrieval crawlers in bot management dashboards.

3. Provide AI-aware default policies — new WAF deployments should prompt administrators to configure AI crawler access rather than inheriting a default that blocks everything.

> ***Methodology note:*** *These recommendations are derived from the findings of the Fortune Global 500 AI Accessibility audit (April 2026), specifically the four-way user agent comparison, WAF vendor attribution, and robots.txt cross-reference analysis. They do not constitute legal or regulatory guidance.*

About PROGEOLAB

PROGEOLAB is an AI-native visibility intelligence platform.

This report is part of the PROGEOLAB Fortune 500 AI Accessibility Audit — a series of research studies on how large enterprises appear to (or disappear from) AI answer engines. All measurements are from live HTTP probes across four user agents: a research bot, Googlebot, Chrome, and ChatGPT-User. No estimates, no third-party data sources.

Methodology in brief

500 companies · 67 probes each · 4 user agents · 134,000 probe requests. Data collected April 16–19, 2026. Response bodies stored and re-validated with MD5-hash soft-404 detection to eliminate the ~25× inflation that status-code-only scans produce.

Contact & next steps

Visit **progeolab.ai/research** or request a demo for a complimentary AI visibility analysis of your organisation.

For press enquiries, data requests, or syndication, write to **research@progeolab.ai**.

PROGEOLAB