Research Article

Report AI Crawler Traffic in Two Columns, Declared and Verified

Evgenii Arsentev[1]

1. Independent researcher

Most counts of AI crawler traffic assign a request to a crawler by its User-Agent header. The client writes that header, so the count shows what clients said about themselves, not who they were. I reanalysed fifteen days of complete access logs from one origin server behind a CDN. The logs hold 345,808 requests, and 46,155 of them carry one of 22 published crawler tokens. For 35.7% of the token-bearing requests, the client connected to the origin directly. Not one of the 11,276 direct requests under tokens with published address ranges came from those ranges. The direct traffic also behaves differently from the traffic that came through the CDN. It never asked for robots.txt or the sitemap, and per request it used about three and a half times as many distinct User-Agent strings. The CDN route is not proof either. A token that Google says it never sends as a User-Agent came through the CDN 293 times. At this origin, no single request could be confirmed by address, because the CDN-routed requests carry the CDN's address in the log. I propose that reports of crawler traffic give separate counts for declared, route-consistent, verified and unverifiable requests. They should state which verification was possible. They should never present the declared count alone as crawler activity.

Correspondence: papers@team.qeios.com — Qeios will forward to the authors

1. Why this matters

Site owners make decisions based on crawler counts. They decide whether to block AI crawlers, whether to publish files for language models, and whether to negotiate access. The simplest way to produce such a count is to take every request whose User-Agent contains a crawler's name. This is the way I used in my own measurement.

 doi.org/10.32388/SRD4TB

The report (Arsentev 2026, DOI 10.5281/zenodo.23018429)[1] asked whether crawlers fetched /llms.txt on one server. The answer was no: none of the 46,155 requests with a crawler token asked for it.

In Section 5 of that report I noted that about a third of those requests bypassed the CDN, and none of them came from the operators' published ranges. There it was a caveat, meant to show that the main zero does not depend on it.

Here I take the caveat as the subject. I ask what the same logs say about counting crawlers at all. I also ask what a report should contain so that a reader can tell the declared number from the checked one.

2. Data

The data are the complete nginx access logs of one origin server for 28 August to 11 September 2026. The server stood behind the Cloudflare CDN[2] and served three hostnames of one publisher into one log. The log has 345,808 parsed requests in the window.

A request is attributed to a crawler when its User-Agent contains, case-insensitively, one of 22 tokens published by search and AI operators. The list and its order are in the report, Section 3.2. In total, 46,155 requests carried a token, and 20 of the 22 tokens occurred.

For each request I know the connecting address as nginx logged it. Using Cloudflare's published ranges, I split the requests into those that arrived from a CDN edge address and those that reached the origin directly. For the direct requests I tested the address against the ranges that Google, OpenAI[3], Microsoft, Apple, Perplexity and DuckDuckGo publish for their crawlers. The range files were downloaded on 28 September 2026, 17 days after the window.

All new numbers in this article come from one script over the same raw logs (declared_vs_route.py). It uses the parsing, the window, the token list and the range snapshot of the published report. It stops with an error if its totals differ from the published ones. Its output holds aggregates only, without addresses or User-Agent strings.

3. What an origin behind a CDN can check

A crawler that resolves a public hostname of this site gets a Cloudflare address and connects to Cloudflare. Cloudflare then opens its own connection to the origin. In the origin log, such a request carries the address of a Cloudflare edge node, not of the crawler. So for 29,691 of the 46,155 token-bearing

requests (64.3%), the origin log has nothing that could confirm or refute the client's identity by address. The CDN itself sees the client address, but in this setup it was not passed into the log.

The other 16,464 requests (35.7%) reached the origin directly. Here the log has the client's own address, so a check is possible. Of these, 11,276 carried a token whose operator publishes address ranges (I also counted Google-Extended against Google's ranges). The result of the check is 0 of 11,276.

For the remaining direct requests, under tokens such as Claude-User, Amazonbot or Bytespider, the operators did not publish range files that I could use. These requests cannot be checked by this method at all.

The outcome is unusual, and it is the first reason for my proposal. At this origin, the number of requests verified by address is zero. The data do not show that all traffic was false. They show that the traffic which could be genuine came through the CDN, where the origin cannot check it, and that the traffic which the origin could check failed the check. A report that writes "46,155 crawler requests" hides both facts in one number.

4. Four signals that the declared count mixes different clients

4.1. The two routes behave differently

If the direct requests were the same crawlers taking another path, they would look like the CDN-routed ones. They do not. Table 1 compares the two groups over the full window.

doi.org/10.32388/SRD4TB

	Through CDN	Direct to origin
Requests with a crawler token	29,691	16,464
Response 2xx	72.5%	0.2%
Response 3xx	6.4%	57.8%
Response 4xx	21.0%	42.1%
Requests for /robots.txt	2,135	0
Requests for /sitemap.xml	1,266	0
Distinct paths requested	3,613	2,165
Distinct User-Agent strings	2,658	5,089
Requests with a User-Agent string seen only once	2,162	4,371
Distinct User-Agent strings per request	0.09	0.31

Table 1. *Token-bearing requests by route, 28 August to 11 September 2026. Source: declared vs route.py. The response rows describe how the server treated each route, not how the clients behaved (see text).*

The response codes need care. Of 16,464 direct requests, only 26 got a successful response. This is mostly the server speaking, not the client. The origin answers plain-HTTP requests with a redirect to HTTPS, and clients that connect to a bare address often arrive over plain HTTP. I checked this rule in the current server configuration. The configuration of the window itself was not archived. For this reason I do not use the response rows as evidence about the clients. The client-side signals are the next three.

More telling for me is robots.txt. Crawlers of the large operators fetch it by protocol (RFC 9309)[4]. Through the CDN, the token-bearing clients asked for it 2,135 times in fifteen days. Through the direct route they asked for it zero times. The same holds for the sitemap. The direct clients also changed their User-Agent string far more often. Per request they used 0.31 distinct strings against 0.09 through the CDN, and 4,371 direct requests used a string that appeared in no other request.

doi.org/10.32388/SRD4TB

4.2. A token that should not appear as a User-Agent

Google documents Google-Extended as a product token for robots.txt, which is not sent as a User-Agent [5]. The report discusses it in Section 5. In this log, 863 requests presented it as one, and none of them also contained Googlebot. Of these, 570 came directly and 293 came through the CDN.

This small number matters for the method. It shows that an imitation can arrive by the normal route. So the CDN route is not a verification. It only means that the route does not contradict the declared name. I therefore call these requests route-consistent, not verified.

4.3. A tidy User-Agent string proves little

One could hope that false clients give themselves away by odd strings. In this log, the largest group of direct requests contradicts that hope. Under the Googlebot token, 5,106 requests came directly, and 99.9% of them used one and the same User-Agent string, with only two distinct strings in total. None of the 5,106 came from Google's published ranges. Through the CDN, the Googlebot token appeared with 11 distinct strings. So the direct group looked more uniform than the CDN group, and a check of string format would have passed it.

Other tokens show the opposite picture. Under GPTBot, the direct requests used 551 distinct strings, and the most frequent string covered 10.6% of them. Format is a weak signal in both directions, and I would not use it as a verification step.

4.4. The share of direct requests differs by token

The direct share was not the same for all names. For YandexBot, DuckDuckBot, cohere-ai and MistralAI-User it was zero, and for bingbot it was 1 request of 2,503. For seven tokens the direct requests were the majority: Google-Extended (66.0%), Perplexity-User (64.4%), Bytespider (62.8%), Claude-User (59.5%), Googlebot (58.3%), Amazonbot (52.8%) and OAI-SearchBot (50.2%).

So any comparison between operators made on declared counts carries a bias that differs from one operator to another. If I rank the tokens by declared requests and then by CDN-routed requests only, the order changes. Bingbot moves from 8th to 5th place, ClaudeBot from 12th to 8th, Claude-User from 7th to 10th and Amazonbot from 9th to 12th. Googlebot stays first in both rankings, but 58.3% of its declared volume belongs to the group that failed the address check.

5. What exists already

The distinction between a declared and a verified crawler isn't new. Operators document how to verify their crawlers [5][6], and CDNs keep their own lists of verified bots. An IETF working group (webbotauth) is also specifying how bots can sign their requests under HTTP Message Signatures [7][8].

What is missing is on the side of the reports. A count built from an origin log doesn't inherit the CDN's classification. And a published number rarely says which of these checks stood behind it.

My proposal is about that gap. It's not a new way to detect crawlers. It's a way to report what was and was not checked.

6. The proposal

I think a crawler report should show more than one number per crawler. I propose that it gives at least four counts, each one separately, and says which of them could be computed in its setting.

Declared. These are requests whose User-Agent contains the token. Most reports give only this count today, and it should keep its name. In the text I would call it "requests presenting the token", not "requests by the crawler".

Route-consistent. These are requests whose route does not contradict the name. Behind a CDN, this means requests that came through the CDN. Without a CDN it means nothing, so the count should be marked as not applicable.

Verified. These are requests whose source was confirmed by a method the operator supports. I know three methods. One is the published address ranges, as I used here. Another is a reverse DNS lookup of the address, followed by a forward lookup of the returned name (Google and Microsoft describe this in their documentation for verifying their crawlers). The third is a cryptographic signature under HTTP Message Signatures (RFC 9421), and the IETF webbotauth group has a protocol draft for it [8]. The report should say which method was used and on what date the ranges were taken.

Unverifiable. For these requests no check was possible. The operator may publish no ranges and no signature, or the log may hold only a proxy address. I would show this group, not drop it into either of the other ones.

Two more items belong in the same table. Requests under tokens that the operator does not send as a User-Agent, such as Google-Extended, should be a separate line. The report should also state whether the

log contains the client address at all. In my case it did not for the CDN route. Cloudflare passes the original client address in a request header (CF-Connecting-IP). If the origin had logged it, the CDN-routed requests could have been checked by the same ranges. In my opinion the next measurement at this origin should log it, and I think this one change would make origin-side studies much stronger at small cost.

I also propose a rule for the wording of conclusions. A conclusion about what crawlers do should rest on the verified count. If the verified count is zero or not available, the conclusion should be about clients presenting the token, and the text should say so in those words.

7. What changes in my own measurement under this rule

I applied the rule to the llms.txt measurement to see whether the findings survive. The main finding is a zero. No request with a crawler token asked for /llms.txt or /llms-full.txt. A zero over all declared requests is also a zero over every subset of them. So it holds for the route-consistent count and for the verified count too. Here the rule changes nothing.

The comparison with robots.txt is different. The report says crawlers fetched robots.txt 698 times and the sitemap 282 times in the days after the files appeared. Under the rule, these numbers rest only on the route-consistent group. The direct group did not ask for robots.txt or the sitemap at all (report, Section 5, and Table 1 above). The numbers stay. Their correct label is “route-consistent requests presenting a crawler token”, not “crawler requests”. The per-token table of the report already carries this label (report, Table 2). I think this is the minimum any such table should carry.

The per-token ranking is the part that would mislead a reader most. Suppose somebody used my declared counts to say that GPTBot was the second most active crawler at this site. The CDN-routed count would put Claude-SearchBot in that place. Neither statement is verified. At this origin, the honest statement is that I cannot rank the operators by verified activity. The report should say that instead of printing a ranking.

8. Limitations

This is one origin server, one operator and fifteen days. The proportions in Table 1 will differ at other sites, and I do not offer them as estimates for the web. My point is about the method of counting. The data only show that the difference between declared and verified can be large.

I took the address ranges 17 days after the window closed. If an operator retired a range in that time, I would have missed it. This could only move some direct requests from failed to verified. It does not affect the CDN-routed group, which could not be checked anyway.

The log pooled three hostnames and did not record the hostname. The routes of the two groups may differ partly for this reason, for example if one hostname was reachable directly and others were not. I did not check this.

The direct group may contain genuine crawlers that I could not verify, for example those of operators without published ranges. I do not claim that every direct request was an imitation. I claim only that the declared count cannot tell which ones were, and that the checks I could run found no genuine ones.

The direct route also selects its clients. Only a client that knows the origin address can use it, for example an address scanner or a client that found the address in old DNS records. This selection may explain much of the difference in Table 1 by itself. It is one more reason not to read the direct group as crawlers on another path.

Finally, the route-consistent group may contain imitations, as the 293 Google-Extended requests show. Its size is therefore an upper bound for genuine crawler traffic at the origin, not an estimate of it.

9. Data and code

The raw logs are not released, because they contain visitor addresses. The report record (DOI 10.5281/zenodo.23018429) holds the analysis scripts, aggregate outputs, the range snapshot and SHA-256 checksums of the raw logs. The script for this article, declared_vs_route.py, uses only the Python standard library and the same inputs. It writes aggregates only. It is published as supplementary material with this article, together with its aggregate output, declared_vs_route.out.json.

Statements and Declarations

Funding

No specific funding was received for this work.

Potential Competing Interests

The author is the author of the technical report and dataset re-analysed here (reference 1). No funding was received and there are no financial competing interests.

Author Contributions

CRediT: Evgenii Arsentev — Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Writing (original draft), Writing (review and editing).

Use of Generative AI

Generative AI (Claude, Anthropic) was used to draft and edit the language of this manuscript and to write the analysis script. Every number in the text is computed by script from the logs described in Section 2 and checked against the report it re-analyses.

References

1. ^*Arsentev E (2026). "No Request With a Known Crawler User-Agent Reached llms.txt: Fifteen Days of Logs at One Origin Server." doi:10.5281/zenodo.23018429.*
2. ^*Cloudflare. "IP Ranges, Accessed 28 September 2026; Cloudflare HTTP Headers (CF-Connecting-IP), developers.cloudflare.com/fundamentals/reference/ Http-Headers/, Accessed 22 October 2026."*
3. ^*OpenAI. "Overview of OpenAI Crawlers and Published IP Range Files. Accessed 28 September 2026."*
4. ^*Koster M, Illyes G, Zeller H, Sassman L (2022). "Robots Exclusion Protocol." RFC 9309.*
5. a, b*Google Search Central. "Google's Common Crawlers and Product Tokens (Google-Extended), Accessed 28 September 2026; Verifying Googlebot and Other Google Crawlers, developers.google.com/search/docs/crawling-indexing/verifying-googlebot, Accessed 22 October 2026."*
6. ^*Microsoft Bing. "Bingbot.json IP Range File, Accessed 28 September 2026; How to Verify That Bingbot Is Bingbot, Bing Webmaster Blog, 31 August 2012."*
7. ^*Backman A, Richer J, Sporny M (2024). "HTTP Message Signatures." RFC 9421.*
8. a, b*"HTTP Message Signatures for Bots." (2026). Internet-Draft draft-ietf-webbotauth- httpsig-protocol-00, IETF webbotauth working group.*

Supplementary data: available at https://doi.org/10.32388/SRD4TB

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: The author is the author of the technical report and dataset re-analysed here (reference 1). No funding was received and there are no financial competing interests.