

A crawler on 20,000 home IPs showed up after I blocked Meta

Isaac Bythewood · 2026-10-14 · 7 min read

sysadmin

webdev

ai



I blocked Meta for ignoring my robots.txt and then my git browser started getting a request every few seconds, each one from a different home internet connection.

I run a small git browser at repos.bythewood.me with my own projects on it, and since the middle of September its robots.txt has been two lines:

```
User-agent: *  
Disallow: /
```

A git browser is about the worst thing you can let a crawler loose on. Every file at every commit is its own URL, so even a couple dozen small repos add up to more pages than anyone could ever want, and every one of them costs a few git subprocesses to render. It used to disallow only the most expensive pages but I gave up on that and disallowed the whole site.

Meta first

Meta's crawler had been poking at it a few hundred times a day through September and then at the start of October it took off. It went from a couple hundred requests a day to 80,000 in about four days, roughly 290,000 requests from 229 addresses in Meta's `2a03:2880::/32` range and 7.65GB served before I did anything. It identified itself as `meta-externalagent`, Meta's own crawler, and it fetched robots.txt 83 times along the way and kept going every time.

My per IP rate limit never kicked in either since the requests were spread across a couple hundred addresses, so on October 6th I added a check that returns a 403 for that whole range. Meta came back

for two more bursts a few days later, one of them climbing to almost 5,000 requests an hour, and all of it got a 403.

Then the other one

Once Meta was out of the way I was still seeing repos at the top of my dashboard all day long, a steady 1,100 to 1,200 requests an hour around the clock. When I dug into it there were about 22,000 requests in a day from 21,000 different IP addresses, so almost every single request came from an address it had never seen before. Most of them claimed to be Chrome on a Mac, a smaller chunk Chrome on Windows, with the version picked at random from somewhere between 110 and 150, and they were walking blobs, trees, logs and raw files pinned to specific commits across all my old archived projects, things like `yarn.lock` and `.editorconfig` from a Django project I haven't touched in years.

I ran a sample of 1,500 of the addresses through [Team Cymru's IP to ASN lookup](#) and they came back as Comcast, AT&T, T-Mobile, Starlink, Orange, BT, Saudi Telecom, Viettel, Maroc Telecom and a long tail of other home and mobile ISPs in dozens of countries. Only four of the 1,500 were anything like a hosting company. That's what a residential proxy network looks like, where companies pay for apps and browser extensions to bundle an SDK that quietly routes other people's traffic through your phone or home connection, then sell access to that pool by the gigabyte.

The user agents look real but the rest of the headers don't. This is roughly what one of their requests looked like coming through Cloudflare, with the Cloudflare headers stripped out:

```
GET /timestrap-django/blob/18cc9e4ffbd4d37e3d950fa92f516c80eccad278/yarn.lock
Accept: */*
Accept-Encoding: gzip, br
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/146.0.0.0 Safari/537.36
```

And this is my phone loading the same site in actual Chrome:

```
GET /
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/jxl,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, br
Accept-Language: en-US,en;q=0.9
Priority: u=0, i
Sec-Ch-Ua: "Chromium";v="136", "Google Chrome";v="136"
Sec-Ch-Ua-Mobile: ?1
Sec-Ch-Ua-Platform: "Android"
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Linux; Android 11; moto g power (2022)) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Mobile Safari/537.36
```

Real Chrome has sent the `Sec-Fetch` headers and `Accept-Language` on every page load for years now, so a request that says it's Chrome and has neither is a script wearing a Chrome user agent. About 83% of all the requests to repos that day looked like that. Another 700 or so came with full browser headers but `Sec-Fetch-Site: none` and no referrer or cookie on deep commit URLs, which is probably the same crawler using a headless browser for some of its requests. My analytics, which only counts visitors that run JavaScript, saw two real page views on repos that same day.

robots.txt only got 18 requests that whole day from everything put together. The crawler almost never asked for the same URL twice in a day, and it held a steady 10 to 15 requests a minute day and night. That

looks like a big crawler with a proper queue and a per site rate limit behind it, and taking every file at every commit makes me think it's someone building a code dataset to train on.

Was it there before Meta?

This is the part that bugs me. Here's a day by day count of requests to repos from Meta's range next to everything else claiming to be Chrome on a Mac or Windows, leaving out my own uptime checker:

day	meta	chrome	chrome IPs
2026-09-20	0	207	171
2026-09-28	148	941	892
2026-10-01	228	2129	2107
2026-10-02	23204	1777	1744
2026-10-04	59874	6338	6165
2026-10-06	83329	7604	7552 ← Meta blocked
2026-10-07	12710	11897	11766
2026-10-08	167	8300	7843
2026-10-10	28792	5058	5039
2026-10-12	26116	6853	6752
2026-10-13	29141	8991	8538
2026-10-14	342	19781	19313

So it was already around before Meta took off, a few hundred requests a day in September, and the two of them ramped up together over the first week of October. But its two biggest jumps came the day after I blocked Meta and right after Meta's last burst of 403s ended. That burst stopped at 7am UTC on the 13th and within a few hours everything else went from about 500 requests an hour to 900, then 1,100 by the next morning. Probably that's just coincidence and both of them found the site around the same time from the same links, but it does look a bit sus. I have no way to know who's actually paying for the residential crawler, since every request points back at some stranger's home connection, so all I can say is that the timing lines up.

Everyone else is seeing it too

I'm far from the only one. A few of the reports I found:

- git.kernel.org's operator [wrote in August](#) that only something like 2% of its requests look like real developers, crawlers use about 20% of its CPU, and about a third of them get through its Anubis proof of work challenge anyway.
- SourceHut has been fighting LLM crawlers since early 2025 when Drew DeVault [wrote about it](#), and this year [an LLM scraping botnet got past their defenses](#) from June to September and they had to turn off parts of the site to cope.
- A Gitea operator [described the same thing in August](#), a rotating pool of IPs with fake Chrome user agents where most addresses make one or two requests, and found per IP rate limiting did nothing for it.
- Google [took down a big chunk of the IPIDEA proxy network](#) in January, which ran a dozen or so residential proxy brands off SDKs in apps on people's devices.
- Cloudflare caught [Perplexity's undeclared crawler](#) in 2025 using the exact same Mac Chrome user agent while rotating IPs and ignoring robots.txt. That's the most common fake user agent there is though so it doesn't point at anybody in particular.

What I'm doing about it

Nothing yet. It's using about 0.02% CPU, 22ms a request and roughly a gigabyte a day, so it's mostly noise in my logs. If I do something it'll probably be a 403 for anything claiming to be Chrome or Firefox

without a `Sec-Fetch-Mode` header on the commit pinned pages, which would take out most of it without touching real browsers, and maybe a Cloudflare challenge on those same paths for the headless part.

I'd love to hear if anyone else saw something take over right after they blocked Meta.