1.28 Million Requests for Pages That Don't Exist

By William Tygart · November 2026

Over 55 days, the origin server behind BC ESG — my ESG and sustainability site — logged 2,008,202 requests. Of those, 1,280,233 came from a single crawler.

That's 63.75% of everything. One bot, outweighing all other traffic to the site combined — every human, every search engine, every other crawler, all of it.

And it was asking for pages that don't exist.

What it was asking for

The requests were for legacy `dictionary.phtml` paths — URLs from an old dictionary-style URL scheme, the kind of pattern an earlier generation of web publishing used to serve definition and reference entries. The site doesn't serve that scheme. It hasn't for a long time. There is no version of those pages to fetch, no content behind those URLs, nothing.

ExaSearchBot requested them 1,280,233 times anyway.

I want to be precise about what kind of traffic this is, because it doesn't fit the categories most people argue about. This wasn't an indexation crawl of live content. It wasn't a live agent fetching a page to answer somebody's question. It was a crawl queue working through a memory — a corpus of URLs assembled at some point in the past, replayed against a server that no longer speaks that language.

The 410, and why the storm continues anyway

My nginx configuration already answers those paths with HTTP 410 Gone, via a rule (`block-phtml.conf`) that was in place before I measured any of this. That rule stays. Permanently.

410 is the correct answer, and it's worth understanding why it's different from the default. A 404 says "not found" — which every crawler correctly interprets as "not found *right now*," possibly temporary, worth another try later. A 410 says "gone." Deliberately, permanently removed. It's the most definitive statement HTTP gives you: stop asking.

The storm continues anyway.

That's not a failure of the 410. That's a lesson about timescales. Crawl corpora and queues are built to be exhaustive and conservative — dropping a URL pattern on the say-so of a status code is exactly the kind of thing large-scale crawlers are slow to do, because a wrong drop loses coverage forever. So the requests keep coming, get their 410 in a few bytes, and cost the server almost nothing per hit. Almost nothing, times 1.28 million, across 55 days, is still the majority of your traffic. The queue will forget eventually. "Eventually" is measured in their time, not yours.

Anyone who tells you a status code fixes bot behavior on a human timescale hasn't watched their own origin logs long enough.

This is the background radiation of a public website

The .phtml storm is the headline because of its size, but it isn't an anomaly. It's the largest example of a category that fills every origin log I've ever read: requests for things that aren't there, from machines that will never load a page.

The same 55 days of BC ESG logs contain 20,156 probes for `/.env` files — the configuration files careless deployments leave exposed, full of credentials. All of them were blocked. Zero breaches. But the probes don't stop because any individual one failed; they're automated sweeps across the whole addressable web, and my server is one stop on the route.

Same logs: more than 3,020 requests for favicons that returned 404. Harmless, trivial — and completely invisible anywhere except here.

That's the mechanical point I keep coming back to. None of this traffic loads a page. None of it executes JavaScript, accepts a consent banner, starts a session, or becomes a "visit" in any ordinary sense of the word. It exists only in the origin server logs, because the origin server is the only thing that ever saw it. When people argue about what their traffic "is," they're usually arguing about the minority of requests that behaved politely enough to be counted by a script on a page. The majority can be something else entirely — in this case, it was one crawler's memory of a dead URL scheme.

The operator's job

So what do you do with a log that says most of your traffic is ghosts? You make decisions — deliberately, per pattern, from the evidence:

Wall off what's dead and dangerous to keep answering. The .phtml scheme is gone and stays gone. The 410 is cheap, definitive, and correct. Probes for `/.env` and its cousins get blocked, not answered. This category is most of the volume and none of the value, and the correct response is a rule, not a worry.

Redirect what carries real demand. Not every ghost is junk. Old paths sometimes hold genuine intent — people and agents following dated links to content that still exists somewhere else on the site. The test is in the logs: what's being asked for, how often, and whether what it maps to still deserves to exist. A redirect is a promise that the destination answers the question the old URL asked. Make that promise only where it's true.

Rebuild what the demand justifies. Occasionally the ghosts are telling you something a keyword tool won't: a topic your site used to cover, or never covered, that machines keep expecting to find. Persistent requests for absent pages are a demand signal with the politeness stripped off. Most of it is noise. Some of it is a content brief.

Count honestly. If you report your traffic without the log-level picture, you are reporting the residue — the requests that survived being real pages loaded by real browsers. Build your expectations, your capacity planning, and your sense of what your site is *for* on the whole record.

Legacy URL schemes never die. Anything your domain ever exposed — under you, or under whoever ran it before you — will be requested for years, by machines working from old maps. The only question is whether you've looked at the map they're using.

We do AI citation restoration. When the AI can't find you, can't read you, or keeps citing somebody else, send us the worst one you have. Your RFPs, your disasters, the problem everyone else already failed to fix — that's the work we want. We come from the restoration industry. We know what's on the floor, and cleaning it up is the job.

And if you're a developer who just needs help on the hard part — the log file nobody can read, the crawl behavior nobody can explain — get a hold of us. That's the part we're best at.