Getting Bots to Respect Boundaries

How AI Crawlers Are Straining Web Infrastructure

Audrey Hingle

January 2026

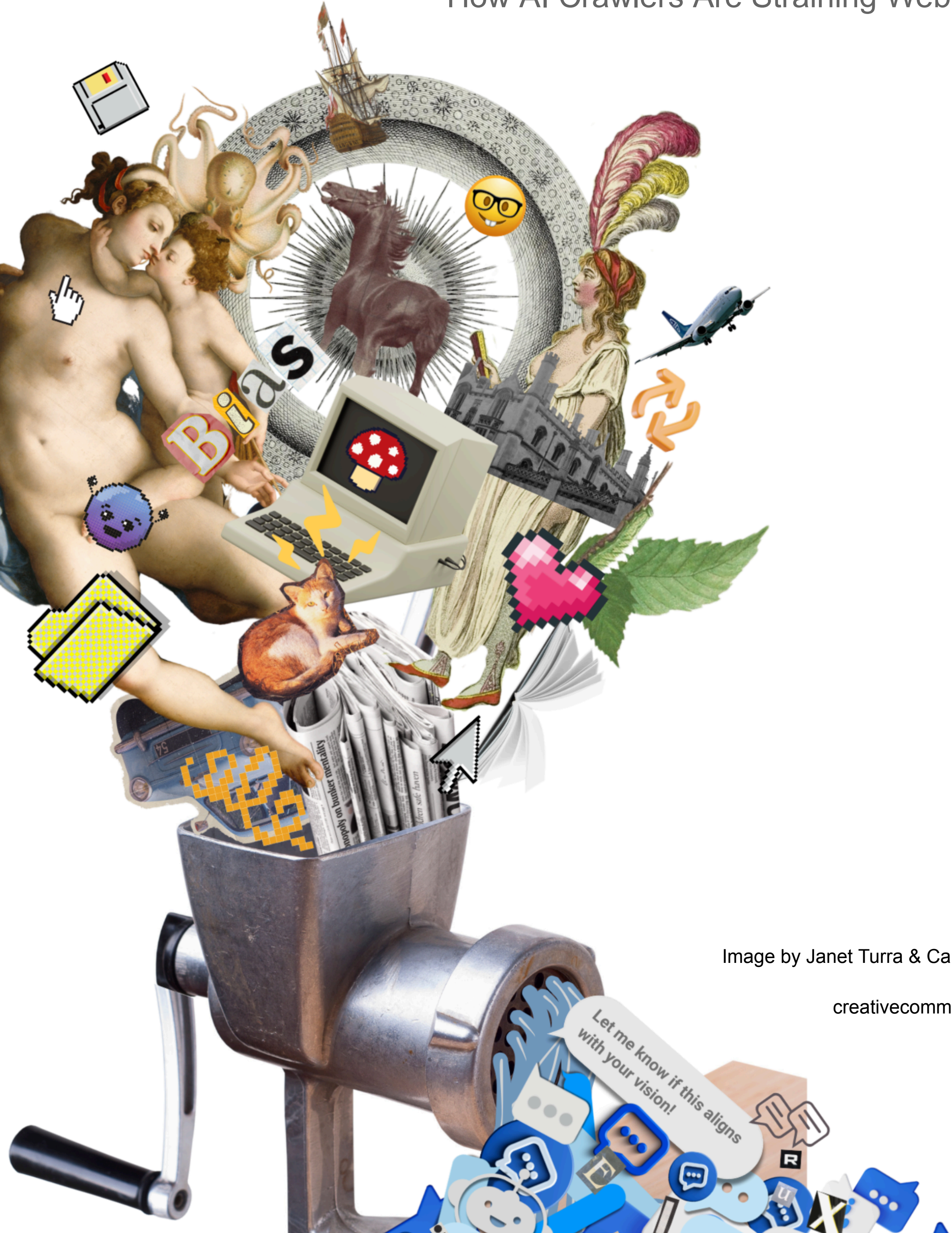
Bias
monopoly on bunker mentality
Let me know if this aligns with your vision!

Image by Janet Turra & Cambridge Diversity Fund
betterimagesofai.org

Contents

AI Scraping as an Infrastructure Problem

In April 2025, Wikimedia Foundation reported a 50% growth in bandwidth used for downloading multimedia content since January 2024. This increase, they said, was not coming from human readers, “but largely from automated programs that scrape the Wikimedia Commons image catalog of openly licensed images to feed images to AI models.” Although bots account for only around 35% of total pageviews, they generate 65% of Wikimedia’s most expensive traffic, because automated crawlers tend to bulk-download large volumes of content, repeatedly access less popular pages, and bypass regional caches, sending requests back to Wikimedia’s core datacentres. This pattern has placed sustained pressure on Wikimedia’s infrastructure, forcing Site Reliability teams to regularly intervene and block aggressive crawlers to prevent disruption for human users.[1]

That is not the only way AI is affecting Wikimedia’s sustainability. Search engines are increasingly using generative AI to answer queries directly on their own platforms, often drawing on Wikipedia content without sending users through to the site. As a result, people are reading Wikipedia-derived information without ever visiting wikipedia.org, contributing to a measurable decline in human pageviews even as demand for Wikimedia’s content continues to grow. The Wikimedia Foundation has warned that this shift risks undermining the volunteer and donor ecosystems that sustain the projects, since fewer visits mean fewer opportunities for readers to contribute, edit, or financially support the human-curated knowledge that AI systems and search platforms increasingly depend on. In other words, as AI bots put strain on Wikimedia’s infrastructure, the funding model for that infrastructure is also under threat.[2]

And Wikimedia is not alone. A similar pattern played out at the University of North Carolina at Chapel Hill in late 2024, when the University Libraries’ online catalog began intermittently failing under what initially appeared to be heavy student use during finals season. In reality, Library IT staff discovered the system was being overwhelmed by highly distributed, automated traffic generating thousands of complex search queries per minute. The requests closely mimicked legitimate research behaviour, using reputable internet service providers and combining large numbers of search filters in ways that were nearly impossible for human users to replicate.

As staff later concluded, the activity was consistent with large-scale scraping to support AI model training. While the bots were sophisticated in their evasiveness, they were inefficient, repeatedly pulling the same records through different paths and placing sustained strain on library infrastructure. Mitigating the attack required a week of intensive work by multiple specialists and the deployment of an AI-based firewall in coordination with central campus IT.

[1] "How Crawlers Impact the Operations of the Wikimedia Projects." Wikimedia Blog. Wikimedia, April 1, 2025. https://diff.wikimedia.org/2025/04/01/how-crawlers-impact-the-operations-of-the-wikimedia-projects/.

[2] "New User Trends on Wikipedia." Wikimedia Blog. Wikimedia, October 17, 2025. https://diff.wikimedia.org/2025/10/17/new-user-trends-on-wikipedia/.

Library staff noted that many peer institutions lack the resources to mount a similar response, leaving them far more vulnerable to disruption.[3]

Reddit, news publishers and cultural institutions all say they are struggling with the costs associated with bot traffic that delivers little in return, like readers or subscribers, that could generate advertising revenue, paid subscriptions, donations or other value.[4] It's no surprise, then, that Wikimedia and other publishers would like to better control traffic coming to their websites.

Together, these trends create a double challenge for publishers. Their content is being extracted at scale for AI training and AI-powered services, often without clear consent, attribution, or compensation, and the volume and behaviour of AI crawlers degrades site performance, raises operational risk, and diverts engineering resources away from serving human users. The existing tool for expressing crawling preferences, robots.txt, was created nearly 30 years ago as a voluntary protocol to signal basic access rules to search engines, researchers, and archiving projects, and was designed for a far more cooperative web—long before anyone could have imagined large-scale automated extraction for commercial AI systems.[5] Left unchecked, large-scale automated extraction risks eroding the very ecosystems that produce and maintain human-curated knowledge, turning a technical strain into a broader governance challenge for the future of the open web.

In response to these pressures, the Internet Engineering Task Force has chartered the AI Preferences (AIPREF) Working Group to develop standardised building blocks that allow authors and publishers to express preferences about how their content is collected and processed for AI model development, deployment, and use. AIPREF is working to define a common vocabulary for these preferences, along with mechanisms for attaching them to content on the web. However, such mechanisms will depend on adoption and compliance by AI developers and crawlers, and will not be sufficient on their own to block bots already ignoring existing voluntary protocols. That is why publishers are deploying technical measures to detect, limit, and manage AI-driven bot traffic. This case study examines those measures and the trade-offs involved in enforcing boundaries in the current web environment.

Understanding the Problem: The Nature of AI Crawlers

The OG Crawlers

The most familiar category of crawlers are those operated by search engines. Search engines enable the discovery of online content by querying structured indexes of the web. Web crawlers,

[3] Panitch, Judy. "Library IT Vs. the AI Bots." UNC University Libraries. June 9, 2025. https://library.unc.edu/news/library-it-vs-the-ai-bots/.
[4] "How AI Bots Are Threatening Your Favorite Websites." The Washington Post, July 1, 2025. https://www.washingtonpost.com/technology/2025/07/01/ai-crawlers-reddit-wikipedia-fight/.
[5] Hingle, Audrey, and Mallory Knodel. "Robots.Txt Is Having a Moment: Here's Why We Should Care." *Tech Policy Press*, April 3, 2025. https://www.techpolicy.press/robotstxt-is-having-a-moment-heres-why-we-should-care/.

also referred to as web robots or spiders, are a core infrastructural component of these systems, responsible for systematically discovering, retrieving, and updating web content for inclusion in search indexes. They are automated software programs that systematically browse the internet by following links from one webpage to another, downloading page content so it can be indexed, analysed, or otherwise processed.

Early search engines and crawlers emerged as a way to catalogue the rapidly expanding World Wide Web of the early 1990s. Although the web was made publicly available in 1991, its early growth was largely driven by academic and research institutions, which also confronted the first challenges of scale and navigation.

The first known web robot, the World Wide Web Wanderer,[6] built by Matthew K. Gray at MIT, launched in spring of 1993 to track the growth of the web.[7] Just a few months later, in September 1993, the first web search engine, W3Catalog, was developed by Oscar Nierstrasz at the University of Geneva.[8] By winter of the same year, JumpStation, written by Jonathan Fletcher and hosted at the University of Stirling in Scotland, combined crawling and indexing to become the first system resembling a modern search engine.[9]

The first crawlers and search tools were developed primarily to measure the size of the web, index documents, and help users locate information, rather than to generate revenue. Their concentration in universities reflected both the origins of the web itself and the fact that these institutions had the technical expertise, network access, and incentive to solve emerging navigation problems. Much like today, early web robots frequently behaved in naïve or aggressive ways, repeatedly downloading the same interlinked documents at high speed, and placing significant strain on servers that were often academic, under-resourced, and manually maintained. As crawling activity increased, concerns about server load and control led to the creation of robots.txt in 1994 by Dutch software engineer and early web developer Martijn Koster.[10] It was (and still is) a simple, voluntary protocol to help these bots behave more sensibly.

While it has been described as "the text file that runs the internet," it was never designed as a security tool or legal instrument.[11] It was a way for site owners to communicate their preferences to search engines, researchers, and archival projects about how their content should be accessed and reused, relying on shared norms and good manners.

[6] Wikipedia. 2026. "World Wide Web Wanderer." Wikimedia Foundation. Last modified October 4, 2026. https://en.wikipedia.org/wiki/World_Wide_Web_Wanderer.

[7] Gray, Matthew. "Credits and Background." Old Home Page of Matthew K. Gray. MIT, https://www.mit.edu/~mkgray/net/background.html.

[8] Wikipedia. 2025. "W3Catalog." Wikimedia Foundation. Last modified July 26, 2025. https://en.wikipedia.org/wiki/W3Catalog.

[9] Wikipedia. 2025. "JumpStation." Wikimedia Foundation. Last modified January 10, 2025. https://en.wikipedia.org/wiki/JumpStation.

[10] "Robots.txt Is 25 Years Old — Martijn Koster's Pages." n.d. https://www.greenhills.co.uk/posts/robotstxt-25/.

[11] Pierce, David. 2024. "The Text File That Runs the Internet." The Verge, February 14, 2024. https://www.theverge.com/24067997/robots-txt-ai-text-file-web-crawlers-spiders.

Commercialisation of search followed later in the 1990s with services like Infoseek (1994), Lycos (1994), Yahoo (1994), and AltaVista (1995).[12] These search engines marked a shift away from university-based projects toward venture-backed and corporate platforms aimed at mass audiences. Profitability came not from crawling itself, but from advertising and sponsorship models tied to search queries and user attention, positioning search engines as both navigation tools and commercial intermediaries within the web's emerging economic structure.

It was Google that ultimately refined the paid and organic search model that still structures the web today. With the launch of Google AdWords in 2000, Google tightly coupled search relevance with auction-based advertising, allowing paid listings to scale alongside organic results. Website owners increasingly optimised their sites for Google's crawlers because high rankings delivered free traffic, creating a tacit bargain between publishers and the search engine.[13] That bargain, however, was not without problems. Google retained unilateral control over ranking algorithms and could demote or remove sites from searches without warning, directly affecting publisher visibility and revenue.[14] Many users of search have also complained that Google search quality has declined steadily over time,[15] particularly for complex or specific queries, and studies show this isn't an anecdotal complaint.[16] Others point to Google emails released as part of the Department of Justice's antitrust case against Google as evidence that Google's declining quality was directly related to internal pressure to increase advertising revenue and query growth, even where those goals conflicted with maintaining the usefulness and integrity of search results.[17]

Search crawlers were not the only automated systems publishers optimised for. As social media platforms became major sources of referral traffic, publishers also adapted their sites to meet the technical requirements of platform-specific crawlers used to generate link previews, thumbnails, and rich embeds. This often went beyond simple metadata, shaping how pages were structured and delivered through formats such as Facebook Instant Articles, Google's Accelerated Mobile Pages (AMP), and Apple News, all of which required publishers to generate specialised versions of their content for platform ingestion. Although these crawlers served a different purpose from search indexing, they still influenced backend publishing practices. Publishers largely accepted these constraints because they offered clear performance benefits

[12] Wikipedia. 2025. "Timeline of Web Search Engines." Wikimedia Foundation. Last modified December 13, 2025. https://en.wikipedia.org/wiki/Timeline_of_web_search_engines.

[13] Wikipedia. 2025. "Google Ads." Wikimedia Foundation. Last modified December 26, 2025. https://en.wikipedia.org/wiki/Google_Ads.

[14] McCullagh, Declan. 2011. "News Sites Helped, Hurt by Google Algorithm Change." CNET, April 19, 2011. https://www.cnet.com/news/privacy/news-sites-helped-hurt-by-google-algorithm-change/.

[15] Davidson, Adam. 2024. "Google Search Has Been Getting Worse for Years." Pocket-Lint, October 4, 2024. https://www.pocket-lint.com/google-search-has-been-getting-worse-for-years/.

[16] Bevendorff, Janek, 1, Matti Wiegmann 2, Martin Potthast 1, Benno Stein 2, Leipzig University, Bauhaus-Universität Weimar, and ScaDS.AI. 2022. "Is Google Getting Worse? A Longitudinal Investigation of SEO Spam in Search Engines." Journal-article. https://downloads.webis.de/publications/papers/bevendorff_2024a.pdf?ref=404media.co.

[17] Zitron, Edward. 2024. "The Man Who Killed Google Search." Ed Zitron's Where's Your Ed At. October 21, 2024. https://www.wheresyoured.at/the-men-who-killed-google/.

and, crucially, reliably sent human readers back to the original sites.[18] That relationship between platforms and publishers has increasingly soured, as engagement on social platforms has grown while referral traffic back to publishers' own sites has steadily declined, weakening the original bargain that justified platform-specific optimisation.[19]

This concentration of power shaped an environment in which the web was extensively optimised for automated crawlers, even as search results were widely believed to be declining in quality, social media was delivering less referral traffic, and users were growing increasingly dissatisfied with traditional search. It's crucial to note that often there was no difference from a publisher's perspective between bots crawling for search engines, social media, or other purposes. That ambiguity was manageable when most automation supported discovery and referral, but it left publishers without clear ways to express consent or draw boundaries based on crawler purpose.

It was into this context that large-scale AI crawlers were introduced: an ecosystem built for machine access, marked by strained publisher–platform relations, and growing demand for new ways to ask questions and retrieve information. Until the public launch of ChatGPT in November 2022, most publishers and creatives who put their content on the web were unaware that their content was being systematically scraped for large-scale AI training, and a number later filed lawsuits reflecting their objection to the use of their content without consent or compensation.[20] In practice, publishers had little control beyond robots.txt to manage crawling, a system that was not designed to govern data extraction for generative AI systems.

AI-focused Crawlers

Search crawlers and AI-focused crawlers differ in purpose, what they extract, how the data is used, and the demand they put on servers.[21] Crawlers built to support search mapped the web so humans could navigate it. They were designed around indexing, retrieval, and referral, and typically obeyed publisher preferences expressed through tools like robots.txt. Search crawlers historically revisited pages relatively infrequently. For many small or static sites, this could mean weeks or even months between crawls. Larger, frequently updated sites might be crawled daily

[18] Penland, Jon. 2016. "Facebook Instant Articles Vs AMP Vs Apple News for WordPress." WPMU DEV. March 31, 2016. https://wpmudev.com/blog/facebook-instant-articles-vs-amp-vs-apple-news-for-wordpress/.
[19] Harmon, Grace. 2025. "Publishers Struggle to Turn Social Media Engagement Into Website Traffic." EMARKETER. June 26, 2025. https://www.emarketer.com/content/publishers-struggle-turn-social-media-engagement-website-traffic.
[20] Harmon, Grace. 2025. "Publishers Struggle to Turn Social Media Engagement Into Website Traffic." EMARKETER. June 26, 2025. https://www.emarketer.com/content/publishers-struggle-turn-social-media-engagement-website-traffic.
[21] McMurray, Morgan. 2024. "Understanding Web Crawlers: Strategies for Traditional and AI Search Bots." December 18, 2024. https://www.botify.com/blog/traditional-and-ai-search-bots

or more often.[22] It was bounded and proportional, constrained by crawl budgets and incentives to minimise server load, because overwhelming sites degraded the quality and sustainability of search itself.

AI-focused crawlers operate very differently. They are optimised not for pointing users back to pages, but for large-scale ingestion of content for training.[23] They tend to prioritise full-text extraction, revisit sites aggressively to maximise coverage, and place significantly higher and more unpredictable load on servers.[24] Eric Hellman, Executive Director of Project Gutenberg told me that "the bots we're seeing now are totally different from two years ago. Back then, bots were mostly well behaved. The real problem was script kiddies, and they were easy to block… the bots that have appeared in the past year are aggressive and badly engineered. They do stupid things and they're kind of psychotic."[25]

Unlike search engines, which store pages and link back to them, AI systems break content down and fold it into the model itself. This often weakens or removes the connection to the original source and because the system is not designed to send users back to the original website, publishers also do not receive the referral traffic they would from search. This extractive model doesn't just bypass publishers: it actively degrades humans' ability to access information by redirecting value away from the open web and into closed, corporate systems.[26] In addition, AI bots, particularly those from smaller vendors may ignore publisher preferences expressed through tools like robots.txt, rotate through IP addresses, or skip user-agent disclosure entirely.[27] These crawlers are stealthy and hard to identify.

As a result, practices that were workable for search: light-touch crawling, clear reciprocity, and slow, predictable re-indexing cycles, do not translate cleanly to AI crawling, and researchers have found that current web control tools are not adequate for protecting publishers against AI crawlers.[28] There is strong interest among publishers to block undesired crawling, but awareness, technical ability, and tool effectiveness are all limited. As Hellman put it "the pace of change is such that what worked six months ago doesn't work today."[29]

[22] Pratt, Kristine. 2025. "How Often Is Google Crawling My Website?" Boostability. May 17, 2025. https://www.boostability.com/content/how-often-is-google-crawling-my-site/.

[23] "The Crawl-to-click Gap: Cloudflare Data on AI Bots, Training, and Referrals." 2025. The Cloudflare Blog. October 15, 2025. https://blog.cloudflare.com/crawlers-click-ai-bots-training/.

[24] Birgit Mueller, Wikimedia Foundation. 2025. "How Crawlers Impact the Operations of the Wikimedia Projects." Diff. April 17, 2025. https://diff.wikimedia.org/2025/04/01/how-crawlers-impact-the-operations-of-the-wikimedia-projects/.

[25] Hellman, Eric. Interview by Audrey Hingle. January 15, 2026.

[26] Bruce Schneier & J.B. Branch, "We cannot let big tech control access to information," San Francisco Chronicle. https://pages.pagesuite.com/4/b/4b9efc32-27f8-454d-b2a1-697203b705eb/page.pdf

[27] Vastel, Antoine. 2025. "From Detection to Trust: The Evolving Challenge of AI Bot Authentication." The Castle Blog. August 6, 2025. https://blog.castle.io/from-detection-to-trust-the-evolving-challenge-of-ai-bot-authentication/.

[28] "Somesite I Used to Crawl: Awareness, Agency and Efficacy in Protecting Content Creators From AI Crawlers." n.d. https://arxiv.org/html/2411.15091v2.

[29] Hellman, Eric. Interview by Audrey Hingle. January 15, 2026.

Existing and Emerging Mitigation Strategies

Because of the nature of modern AI crawlers, and the strain they put on publishers, publishers are deploying a range of technical measures to detect, limit, and manage AI-driven bot traffic. As background to writing this section, I spoke to Jamie McClelland, director of technology services for The Progressive Technology Project and co-founder and board member of May First Movement Technology, a non-profit membership organisation that engages in building movements by advancing the strategic use and collective control of technology for local struggles, global transformation, and emancipation without borders. I also spoke to Eric Hellman, Executive Director of Project Gutenberg, about how AI-driven scraping is changing the practical realities of keeping public-interest content online and Nick Sullivan, an independent technologist and applied cryptographer, formerly Head of Research at Cloudflare and a leader in internet standards work at the IETF, about Venom and defensive data poisoning.

McClelland described two primary strategies used by smaller, independent web hosts and publishers to manage bot traffic: IP-based blocking and proof-of-work mechanisms. IP blocking, he noted, is unavoidable in practice: "We would be dead in the water if we weren't banning some IP addresses," even though it means routinely punishing users who "don't deserve it."[30] These measures are not applied in real time, but through daily analysis of activity patterns, a process that is difficult to calibrate effectively and frequently produces false positives. In some cases, entire networks are blocked.

As an alternative, May First has increasingly relied on proof-of-work systems such as Anubis, which require browsers to demonstrate they are legitimate for the duration of a session. McClelland prefers this approach because it is session-based rather than IP-based. While session-based blocking is less likely to be over-broad, he emphasised that it is also "fraught," difficult to implement without errors, and generates frequent user complaints when challenges fail or block legitimate access. McClelland described the situation, as it stands, as having "bad options" with organisations forced to choose the "least bad," which are "still pretty bad."

Hellman emphasised that most current defences are aimed at keeping services available rather than achieving clean or durable enforcement, and that even these measures are becoming less reliable as crawler behaviour escalates. He described a rapid shift in the threat landscape: traffic that was manageable a year or two ago has doubled in volume, is far more distributed, and is increasingly difficult to distinguish from human use. As a result, techniques that once worked predictably no longer hold. "The pace of change is such that what worked six months ago doesn't work today," he said, noting that it is now often impossible to reliably determine which traffic is human and which is automated. While Project Gutenberg has avoided significant network-level blocking and has only occasionally shut off particularly aggressive IP addresses, Hellman stressed that there is no guarantee these approaches will remain viable. More broadly, he pointed to selective friction as a short-term survival tactic: on other sites he manages, gating the most expensive search functions has helped deter the worst abuse, but only temporarily. In

[30] McClelland, Jamie. Interview by Audrey Hingle. January 14, 2026.

Hellman’s view, the deeper problem is that scraping incentives continue to favour escalation, even as defensive tools grow less effective over time.[31]

IP-Based and Network-Level Controls

Blocking or throttling traffic based on network identity rather than behaviour.

These approaches rely on identifying and restricting traffic at the IP or network level. They are often effective in practice but risk over-blocking and disproportionately affecting legitimate users.

Blocking Known Bots (via User-Agent filtering)

User-agents are software programs that initiate requests on the web. The most familiar of which are web browsers, which are designed to be used and controlled by human users. User-agents also include systems that are not directly controlled by people: search engine crawlers, automated scripts, mobile applications, and networked devices.[32]

User-agents not controlled by people operate automatically in the background, following pre-configured instructions. As a result, some user-agents cannot provide user prompts or warnings, and any errors or confirmations may be handled through configuration, logs, or system-level settings rather than direct interaction.[33]

Blocking known bots via User-Agent (UA) filtering is one of the oldest and most widely deployed techniques for managing automated traffic. In this approach, HTTP requests are inspected for their declared User-Agent string, and requests matching known crawler identifiers are blocked, throttled, or otherwise restricted at the web server or reverse proxy layer. [34]

This method relies on voluntary self-identification by automated clients. Many large-scale crawlers, including search engines and some AI-associated bots, publicly document their User-Agent strings and expected crawling behaviour. Web servers can therefore implement allowlists, denylists, or rate limits for these agents using standard tooling in Apache, Nginx, HAProxy, or similar infrastructure.[35]

From an operational standpoint, User-Agent filtering is attractive because it is easy to deploy and is lightweight and inexpensive to run. Rules can often be implemented with a single

[31] Hellman, Eric. Interview by Audrey Hingle. January 15, 2026.
[32] Fielding, Roy T., Mark Nottingham, and Julian Reschke. 2022. “RFC 9110: HTTP Semantics.” June 1, 2022. https://www.rfc-editor.org/rfc/rfc9110.html#name-user-agents.
[33] “Definition of User Agent - WAI UA Wiki.” n.d. https://www.w3.org/WAI/UA/work/wiki/Definition_of_User_Agent.
[34] Hubbard, Mandy. 2023. “Get Started With User Agent Filtering.” Ngrok Blog (blog). December 15, 2023. https://ngrok.com/blog/get-started-with-user-agent-filtering.
[35] Edwards, Jeff. "The Ultimate 2026 List of Web Crawlers and Good Bots: Identification, Examples, and Best Practices." Human Security. June 2, 2025. https://www.humansecurity.com/learn/blog/crawlers-list-known-bots-guide/.

configuration change and evaluated early in the request lifecycle, making this technique accessible even to small operators with limited resources.[36]

However, UA-based blocking has limits. User-Agent strings are not verified or checked for accuracy: they are simply labels that clients choose to send. The web standards treat them as informational, not as a security feature, and software is free to leave them out or change them. Because of this, UA filtering mainly affects “well-behaved” bots that honestly identify themselves, and does little to stop scrapers or bots that are trying to avoid detection.

Empirical reports from site operators indicate that a significant portion of abusive scraping traffic now originates from botnets or distributed scraping services that deliberately impersonate mainstream browsers or rotate User-Agent strings to evade detection. This limits the long-term utility of UA filtering as a standalone defence and creates a perverse incentive structure: compliant actors are constrained, while non-compliant actors face few barriers.[37]

For these reasons, User-Agent filtering works best as a first line of defence rather than a complete solution. It can help reduce traffic from well-behaved crawlers, signal what kinds of access are expected, and give operators a simple way to separate different types of traffic. On its own, though, it is not enough to deal with large-scale or deliberately evasive scraping, including much of what is associated with modern AI data collection. In practice, it needs to be paired with other approaches such as rate limiting, behaviour-based detection, or challenge-response systems.

Per-IP Rate Limiting

Per-IP rate limiting is one of the most common techniques publishers use to manage automated traffic. In its simplest form, it restricts how many requests a single IP address can make to a website within a given time period. Requests that exceed this threshold are delayed, throttled, or blocked.[38] The approach is attractive because it is easy to deploy, well supported, and relatively cheap to run. For many sites, it is the first line of defence against crawlers that place excessive load on infrastructure.

However, per-IP rate limiting was designed for a web where abusive traffic typically came from a small number of clearly identifiable sources, but AI crawlers often distribute requests across

[36] Hubbard, Mandy. 2023. “Get Started With User Agent Filtering.” Ngrok Blog (blog). December 15, 2023. https://ngrok.com/blog/get-started-with-user-agent-filtering.

[37] Thales Group. 2025. “Artificial Intelligence Drives Surge in Bot Traffic, Now Surpassing Human Activity, According to 2025 Imperva Bad Bot Report.” *Thales Cloud Security Products*, April 15, 2025. https://cpl.thalesgroup.com/about-us/newsroom/2025-imperva-bad-bot-report-ai-internet-traffic.

[38] "What Is Rate Limiting?" Cloudflare. https://www.cloudflare.com/en-gb/learning/bots/what-is-rate-limiting/.

large numbers of IP addresses, including cloud providers,[39] residential proxies, and botnets.[40] By spreading activity in this way, crawlers can stay below per-IP thresholds while still generating very high aggregate load. As a result, rate limiting can give a false sense of protection while failing to address the underlying problem. As Hellman noted “we have on occasion shut off particularly aggressive and psychotic IP addresses. I don’t know if that will continue to work at all going forward.”[41]

Per-IP limits also carry a risk of collateral damage, a concern McClelland raised at the beginning of this section. Many legitimate users and services share IP addresses, particularly in environments such as universities, libraries, mobile networks, and workplaces. Strict limits can unintentionally block or degrade access for human users, researchers, or accessibility tools, especially in regions where network address translation is common. In these cases, rate limiting shifts the burden of AI crawling onto people who have no control over how their IP address is shared.[42]

More sophisticated extensions of per-IP rate limiting are beginning to emerge. One example is *Logrip*, which analyses patterns across related IP addresses over time in order to detect distributed crawling that evades simple per-IP thresholds.[43] Approaches like this underline both the continued relevance of rate limiting as a defensive technique and the increasingly adversarial nature of automated access, in which crawlers adapt their behaviour specifically to avoid existing controls.

Behavioural Detection and Pattern-Based Controls

Identifying automated traffic based on how clients behave rather than where they connect from.

These approaches analyse request patterns over time, such as navigation paths, error rates, or asset loading behaviour, to distinguish automated activity from human use. They can be more resilient to IP rotation and distributed crawling, but depend on assumptions about “normal” behaviour and may generate false positives, particularly for specialised or non-standard clients.

Referer Analysis

A simple way websites try to tell bots from real users is by looking at the HTTP “Referer” header, the field that tells the server where a request came from. When a person clicks a link in a browser, that browser usually sends a referer showing the page they came from. Bots and

[39] “How to Identify and Stop Scrapers.” n.d. F5 Labs. https://www.f5.com/labs/articles/how-to-identify-and-stop-scrapers.
[40] Brian Krebs, “The Kimwolf Botnet Is Stalking Your Local Network,” Krebs on Security, January 2, 2026. https://krebsonsecurity.com/2026/01/the-kimwolf-botnet-is-stalking-your-local-network
[41] Hellman, Eric. Interview by Audrey Hingle. January 15, 2026.
[42] “One IP Address, Many Users: Detecting CGNAT to Reduce Collateral Effects.” 2025. The Cloudflare Blog. October 29, 2025. https://blog.cloudflare.com/detecting-cgn-to-reduce-collateral-damage/.
[43] Höetzlein, Rama K. "Protecting Small Organizations from AI Bots with Logrip: Hierarchical IP Hashing." (2025). https://arxiv.org/pdf/2508.03130.

scripts often do not send a referer by default unless it is explicitly programmed, so checking whether a referer is present can be a quick rule of thumb for spotting automated traffic.[44]

This makes referer analysis a fast and easy way to spot obvious bots. If a request lacks a referer when it should have one, a site might treat that request as suspicious and block or throttle it. Some web application firewalls and security tools even build rules around suspicious or missing referer values to cut down on unwanted traffic.

However, referer checks are easy to break. More sophisticated scrapers can simply add a fake or plausible referer to every request, making them appear to come from a real user.[45] Because the referer header is optional and can be changed freely, it cannot be trusted on its own as a signal of legitimacy. There's also a risk that legitimate requests may be blocked if they do not include the expected referer. For example, a user may arrive at a page by typing a URL directly or using a bookmark, neither of which generates a referer header. In those cases, a site that blocks requests without a referer could unintentionally deny access to normal users.

File Request Profiling (e.g., blocking clients who don't request .css, .jpg)

File request profiling is another way websites try to tell humans and bots apart. It looks at what kinds of files a visitor requests, not just how many requests they make. When people use a normal web browser, it usually downloads many different files at once, such as HTML pages, images (.jpg, .png), stylesheets (.css), fonts, and scripts. Basic scrapers often do not. They may only request the raw HTML or specific data endpoints and skip everything needed to display the page.[46]

Because of this, some sites block or limit clients that do not request supporting files like images or stylesheets. If a visitor repeatedly asks for pages but never loads .css or .jpg files, the traffic is treated as likely automated. This can be a fairly accurate way to catch simple bots that are only extracting text or data and are not pretending to be full browsers. As a result, missing or unusual asset-request patterns are commonly treated as a signal of automated activity, though not a definitive indicator on their own.

However, this approach also has limits. Not all legitimate clients behave like a full browser. Some accessibility tools, text-only browsers, performance monitors, and API clients intentionally avoid loading images or stylesheets. In these cases, file request profiling can produce false positives, blocking users who are not doing anything abusive. Security researchers note that

[44] Agrawal, Adarsh. "🧭 Understanding the Referer Header: The Silent Signal That Outsmarts Bots." Medium. June 29, 2025. https://medium.com/%40agrawal.adarsh3004/understanding-the-referer-header-the-silent-signal-that-outsmarts-bots-80253e3df680.

[45] Rethabile. 2018. "Do Bots Actually Set the Referer Url?" Stack Overflow. March 29, 2018. https://stackoverflow.com/questions/49550954/do-bots-actually-set-the-referer-url.

[46] Muwandi, Tafara. "How to Identify and Stop Scrapers." Medium. F5 Labs, May 9, 2025. https://www.f5.com/labs/articles/how-to-identify-and-stop-scrapers.

automated checks based on missing assets can wrongly classify lightweight or specialised clients as bots.[47]

Like other behavioural techniques, file request profiling also becomes less effective over time. More advanced scrapers already request images, stylesheets, and other assets to use or to blend in with normal traffic. As a result, this method mainly catches unsophisticated bots, while more capable automated systems adapt quickly.

404 Error Monitoring

404 error monitoring is another way websites try to detect automated traffic. A 404 error happens when a client asks for a page or file that does not exist. Human users usually reach pages by clicking links or using search, so they tend to generate very few 404 errors. Scrapers, on the other hand, often guess URLs, crawl entire directories, or reuse scripts across many sites. This makes them more likely to request pages that are missing or were never there.

Because of this, repeated 404 errors are often treated as a sign of automated scanning or scraping. Security teams watch for clients that trigger many 404 responses in a short period of time. When this happens, the traffic may be flagged or limited, since it looks more like probing or crawling than normal browsing.[48] In this sense, 404 monitoring works well as a proxy for intrusion detection, helping identify activity that does not follow expected user paths.

However, 404 errors are not always a sign of abuse. A 404 error simply means the server couldn't find the requested resource, and common causes include users mistyping URLs, following broken or outdated bookmarks, or requesting resources that were moved or deleted. These are normal, everyday scenarios that generate 404s from real users and tools.[49]

As a result, blocking traffic based on 404 errors alone can lead to false positives. Like other behavioural techniques, 404 monitoring works best as a supporting signal rather than a hard rule.

Client-Side Challenges and Proof-of-Work

Requiring browsers or clients to demonstrate legitimacy at the session or interaction level.

These techniques attempt to distinguish humans from automated traffic through client-side challenges or computational tasks that impose a cost on the requester, such as CAPTCHAs, JavaScript challenges, or proof-of-work puzzles. They can be effective at deterring large-scale

[47] Vastel, Antoine. 2025. "How Bot Detection Misfires on Non-mainstream Browsers and Privacy Tools." The Castle Blog. June 17, 2025. https://blog.castle.io/how-bot-detection-misfires-on-non-mainstream-browsers-and-privacy-tools/.

[48] Wimmenhoeve, Leon. 2025. "Suspected Bots Causing 404 Errors." Really Simple Security. October 10, 2025. https://really-simple-ssl.com/suspected-bots-causing-404-errors/.

[49] Wikipedia contributors. 2025. "HTTP 404." Wikipedia. December 17, 2025. https://en.wikipedia.org/wiki/HTTP_404.

automation, but often raise concerns around accessibility, privacy, and compatibility with non-browser clients, and may shift friction onto legitimate users.

JavaScript and Cookie Checks

JavaScript and cookie checks are another way websites try to control automated traffic. These checks are based on a simple idea: most people use browsers that run JavaScript and accept cookies, while many basic bots do not.[50] In practice, this often means a site requires a visitor to run a small piece of JavaScript or return a cookie before allowing access. These rules are often enforced at the proxy level, for example using HAProxy.[51]

When they work, these checks can block simple scrapers early. Bots that only download HTML and do not run JavaScript are stopped before they reach the main site. Cookie checks can also help tell the difference between one-off requests and ongoing sessions. For site operators, these tools are easy to set up and do not require much extra infrastructure.

Some newer tools extend this approach by combining JavaScript execution with proof-of-work requirements. For example, Anubis is an open-source system that presents clients with a JavaScript-based computational puzzle before granting access, with the aim of making large-scale scraping more expensive while remaining relatively unobtrusive for human visitors.[52]

However, these checks can also block real people. Not everyone can or wants to run JavaScript or accept cookies. This includes people using assistive technologies, text-only browsers, older devices, or strict privacy settings.[53] When these users are blocked, they may not understand why. This creates real accessibility and usability problems, especially for public-interest sites.

These checks are also not future-proof. More advanced bots already run JavaScript, store cookies, and behave like normal browsers.[54] Even proof-of-work systems such as Anubis rely on assumptions about cost asymmetry that may erode as automation becomes cheaper and more capable. As a result, JavaScript and cookie checks mainly stop the simplest bots, not the ones causing the most harm.

Captcha Challenges

CAPTCHA stands for “Completely Automated Public Turing test to tell Computers and Humans Apart.” It is a common way websites try to block bots by asking visitors to solve a puzzle before they can do something like create an account or post content. For example, a CAPTCHA may

[50] “W3Schools.com.” n.d. https://www.w3schools.com/js/js_cookies.asp.
[51] Technologies, HAProxy. 2025. “What Is HAProxy?” HAProxy Technologies (blog). December 1, 2025. https://www.haproxy.com/glossary/what-is-haproxy.
[52] “Anubis: Web AI Firewall Utility | Anubis.” n.d. https://anubis.techaro.lol/.
[53] Clarke, James M., Maryam Mehrnezhad, and Ehsan Toreini. 2024. “Invisible, Unreadable, and Inaudible Cookie Notices: An Evaluation of Cookie Notices for Users With Visual Impairments.” ACM Transactions on Accessible Computing 17 (1): 1–39. https://doi.org/10.1145/3641281.
[54] "Cloudflare JS Challenge: How It Works and How to Solve It." Medium. https://medium.com/@datajournal/cloudflare-js-challenge-how-to-solve-83c2f02b92e1.

ask a user to click on all the pictures with traffic lights or type letters from a distorted image. The goal is to make bots fail while letting humans through.

Many large sites, including Wikipedia, have used CAPTCHAs to defend against automated abuse such as spam, vandalism, or bot-driven account creation. Recently, the Wikimedia Foundation announced it will trial hCaptcha, a bot-detection service designed to replace its older CAPTCHA system so that it is better at stopping modern automated attacks while still being easier for real people to use.[55]

CAPTCHAs reflect a long-running cycle of escalation between website defenders and automated attackers. As bots get better at acting like people, CAPTCHA challenges are made harder in response. The result is often counter-productive: many modern AI systems can pass these tests with little trouble, while real users are left dealing with confusing puzzles, repeated failures, and unnecessary friction.[56]

Commercial and Infrastructure-Heavy Mitigation Systems

Paid, platform-mediated, or technically complex solutions often inaccessible to smaller actors.

These tools typically operate at scale through intermediaries such as reverse proxies or security platforms. While powerful, they introduce cost, dependency, and governance concerns, particularly for civil society and small organisations.

Web Application Firewalls (WAFs)

Web Application Firewalls, or WAFs, sit in front of a website and inspect incoming traffic before it reaches the server. They analyse requests based on factors such as IP address, headers, request frequency, and known attack patterns, and can block, slow, or challenge traffic that appears automated.[57] As AI scraping has increased, WAFs have become a common tool for websites trying to protect content and infrastructure from large volumes of bot traffic.

Tools such as SafeLine, open-appsec, and BunkerWeb bundle together multiple defensive techniques, including rate limiting, header inspection, and behavioural analysis. For site operators facing both identifiable crawlers and more covert AI scraping, WAFs can provide a single, centralised layer of control that is relatively easy to deploy.[58]

[55] Mill, Eric. 2025. “Better Detecting Bots and Replacing Our CAPTCHA.” Diff. September 2, 2025. https://diff.wikimedia.org/2025/09/02/better-detecting-bots-and-replacing-our-captcha/.
[56] Hendry, Andrew. "CAPTCHA in the Age of AI: Why It’s No Longer Enough." DataDome. https://datadome.co/bot-management-protection/captcha-in-the-age-of-ai-why-its-no-longer-enough/.
[57] Hendry, Andrew. "What Is a WAF? | Web Application Firewall Explained." CloudFlare. https://www.cloudflare.com/en-gb/learning/ddos/glossary/web-application-firewall-waf/.
[58] Hendry, Andrew. "7 Free WAF Solutions Compared: Which One Should You Use in 2025?" CloudFlare. July 15, 2025. https://medium.com/%40tvvzvpb186/7-free-waf-solutions-compared-which-one-should-you-use-in-2025-81260f8ee76d.

At the same time, WAFs reflect many of the trade-offs that run through current anti-scraping strategies. Most mature WAF products are closed-source or freemium, limiting transparency and making it difficult for site owners to fully understand or customise how decisions are made. Open-source options exist, but they are fewer in number and often require significant expertise to configure and maintain.[59] Like other enforcement-based tools, WAFs rely on pattern recognition and probabilities rather than shared protocol-level signals, which can lead to false positives and uneven access.

In this sense, WAFs are best understood as part of a broader, layered response rather than a complete solution. They can help reduce infrastructure strain and blunt large-scale scraping, but they do not resolve the deeper problem of how websites can express and enforce consent around AI training in a way that is transparent, fair, and interoperable. As AI-driven automation continues to grow, WAFs remain a useful but partial defence within a fragmented and contested web ecosystem.

Adversarial Response and Defensive Data Poisoning

Recent research has explored a more aggressive class of anti-scraping techniques. While most current approaches focus on blocking or rate-limiting automated access, Nick Sullivan explained that adversarial responses aim instead to change the incentive structure by increasing the cost and risk of ignoring preference signals.[60]

Some techniques focus on detection and proof. One example is the use of copyright traps: deliberately inserted, unique text sequences or tracking markers embedded in content. These traps are designed to be highly unlikely to occur naturally. If they later appear in model outputs, this can provide evidence that an AI provider bypassed stated crawling preferences and incorporated the data into a training set. Such evidence could allow a copyright holder to demonstrate unauthorised use, support contractual or legal claims, or strengthen their negotiating position with model providers.[61]

Another approach is Cloudflare's Labyrinth, which detects AI crawler behaviour and traps bots in an endless network of meaningless pages, increasing crawl costs while also generating signals to improve bot detection.[62] Another, more extreme example is the HTML bomb, which serves crawlers a page that rapidly expands in size and crashes the system processing it.[63]

Venom represents a different approach. An experimental system that sits in front of a website, Venom monitors incoming traffic for signals that a visitor may be an automated crawler rather

[59] Zenarmor. 2025. "Zenarmor Documentation." May 16, 2025. https://www.zenarmor.com/docs/network-security-tutorials/best-open-source-web-application-firewalls.
[60] Nick Sullivan, interview by Audrey Hingle, January 19, 2026.
[61] Matthieu Meeus, Igor Shilov, Manuel Faysse, and Yves-Alexandre de Montjoye, "Copyright Traps for Large Language Models," arXiv preprint arXiv:2402.09363 (2024), https://arxiv.org/abs/2402.09363
[62] "Trapping Misbehaving Bots in an AI Labyrinth." 2025. The Cloudflare Blog. December 22, 2025. https://blog.cloudflare.com/ai-labyrinth/.
[63] "A Valid HTML Zip Bomb - Ache." n.d. https://ache.one/notes/html_zip_bomb.

than a human user. It is designed to operate under uncertainty. When the system cannot confidently classify a visitor, it does not need to make a binary allow-or-block decision.

Instead, Venom can selectively alter the content served to visitors that are suspected of being crawlers. Human users continue to receive the normal site experience, while automated systems may be served modified or misleading content that remains structurally valid. This allows sites to respond to ambiguous traffic without risking false positives that could degrade the user experience.[64]

If collected at scale and incorporated into training datasets, this altered content can degrade model performance, introducing costs and risks that are difficult for operators to detect or mitigate. For large or high-profile websites, this ability to handle uncertainty is a key advantage. Blocking carries the risk of excluding legitimate users, particularly when crawlers increasingly mimic human behaviour. Returning visually identical pages to people while subtly altering what automated systems receive offers a safer middle ground.

Venom is not only about protecting infrastructure. Its goal is to raise the cost and risk associated with scraping at scale, and ignoring consent signals, making indiscriminate crawling a less attractive strategy.[65]

Community-Driven Bot and Threat Detection

One way to lower the cost of defending small sites against large-scale scraping might be through community-driven, pooled security systems that share the work of detection and response across many publishers. One example of this is CrowdSec, an open-source security tool that detects and blocks malicious or automated traffic by analysing behaviour rather than relying on static rules. It works by monitoring logs and requests for patterns that suggest abuse, such as scraping, credential stuffing, or coordinated scanning. When suspicious activity is detected, CrowdSec can trigger local enforcement through integrations with web servers, firewalls, and reverse proxies.

What makes CrowdSec distinctive is its shared threat intelligence model. Jona Azizaj, Open Source Community Manager at CrowdSec told me that “Crawlers often rely on ephemeral infrastructure, which limits the effectiveness of static blocklists. The dynamic nature of our approach allows us to continuously adapt and better keep pace with these crawlers.”[66] When many CrowdSec users observe similar malicious behaviour from the same IP addresses, those observations are pooled into a community blocklist. Other participants can then use that information to pre-emptively block traffic that has already been identified elsewhere. This shifts defence from an isolated, site-by-site effort toward a collective approach based on shared experience.

[64] Nick Sullivan, interview by Audrey Hingle, January 19, 2026.

[65] Sullivan, Nick. 2025. “Fighting Fire With Venom: Adversarial Defense Against Unauthorized Web Crawling.” USENIX. 2025. https://www.usenix.org/conference/usenixsecurity25/presentation/sullivan.

[66] Jona Azizaj, email message to Audrey Hingle, January 21, 2026.

This approach mirrors the way email spam filtering systems work. For decades, email providers have relied on shared spam clearinghouses and DNS-based blocklists, where mail servers report abusive sending behaviour and use aggregated reputation data to filter spam before it reaches users' inboxes.[67] CrowdSec applies a similar collective reputation model to web traffic, rather than email.

In the context of AI scraping, this model offers clear benefits. Large-scale crawlers often distribute requests across many sites and networks, making them hard to detect from a single vantage point. By aggregating signals across organisations, CrowdSec can surface patterns that individual publishers might miss, particularly when dealing with distributed or low-intensity scraping that stays below local thresholds.

At the same time, CrowdSec still reflects the limits of today's enforcement-based controls. Decisions about access are inferred from behaviour rather than expressed through shared, protocol-level signals. Participation also requires operational capacity and ongoing tuning, which may be out of reach for some smaller sites. Even so, CrowdSec points toward a more cooperative model of defence, one that aligns with the broader need for community-driven responses to the strain that large-scale automation places on the open web.

Monetisation Gateways

Monetisation gateways are an emerging response to large-scale AI scraping. Instead of trying to block crawlers entirely, these systems offer a way for site owners to charge AI companies for access. Cloudflare's AI Bot Marketplace is one example: it allows publishers to set terms under which AI crawlers may access their content, shifting scraping from unauthorised extraction toward licensed and compensated use.[68]

For publishers under infrastructure strain, this model is appealing. It acknowledges that AI companies derive value from web content and creates a mechanism to recover some of the costs associated with hosting, bandwidth, and maintenance. In theory, it also provides clearer boundaries than today's ad hoc blocking, replacing silent extraction with explicit agreements.

This approach may also replace some of the revenue from models that depend on human attention.[69] Many sites have historically relied on visits, advertising impressions, or voluntary donations, including projects like Wikipedia as well as community resources and news organisations. Those income streams are increasingly under pressure as AI systems answer questions directly without sending users back to the original source. Monetisation gateways

[67] Wikipedia contributors. 2025. "Distributed Checksum Clearinghouse." Wikipedia. May 31, 2025. https://en.wikipedia.org/wiki/Distributed_Checksum_Clearinghouse.
[68] Zeff, Maxwell. 2025. "Cloudflare Launches a Marketplace That Lets Websites Charge AI Bots for Scraping." TechCrunch, July 1, 2025. https://techcrunch.com/2025/07/01/cloudflare-launches-a-marketplace-that-lets-websites-charge-ai-bots-for-scraping/.
[69] "How AI Is Rewriting the Web's Attention Economy." 2025. The Economics Review. December 22, 2025. https://theeconreview.com/2025/12/12/how-ai-is-rewriting-the-webs-attention-economy/.

promise an alternative source of income that does not depend on page views, but whether payments from AI companies could realistically replace lost traffic, advertising revenue, or reader donations remains an open question.

Critics also argue that this model rests on fragile assumptions. Writing in *TechRadar*, Alistair Vigier points out that Pay-Per-Crawl treats all pages as equally valuable, despite vast differences in the cost, effort, and originality behind different kinds of content. High-value investigative journalism, long-term research, or carefully maintained documentation is priced the same as generic or publicly sourced material, making the marketplace unattractive to both publishers and AI companies.[70] As a result, many AI firms are unlikely to participate when comparable data is already available through large, unpaid datasets such as Common Crawl.

There are additional concerns beyond centralisation. Smaller publishers may lack the leverage to set meaningful prices or negotiate fair terms, while large platforms are better positioned to benefit. Sites that opt out may still face scraping from actors outside the marketplace, leaving them with enforcement costs but no compensation.

More broadly, monetisation gateways risk normalising the idea that large-scale automated extraction is acceptable so long as it is paid for. Rather than reinforcing norms of restraint or respect for publisher preferences, they shift the debate toward who controls access and who collects fees. As a result, scraping is reframed as an economic transaction rather than a governance issue, leaving unresolved questions about fairness, enforceability, and the long-term balance of power on the web.

Monetisation gateways therefore offer a partial response to the economic pressures created by AI scraping, but they do not resolve the underlying governance problem. They shift the question from whether content should be accessed to who controls access, under what conditions, and with whose interests prioritised.

Impact on the Internet

The technical steps publishers are taking to manage AI crawler traffic are understandable. In many cases, they are necessary to keep websites working for human users. But taken together, these measures also have wider effects on how the internet functions and who can participate in it. Not only that, these measures are likely to break down over time as bots escalate their attacks. This section looks at those effects through the lens of what the internet needs to exist and to thrive.

Access and Common Protocols

The internet was built on a simple idea: if you follow shared technical rules, you should be able to connect and communicate. That idea is now under real strain. Many of the ways websites are

[70] https://www.techradar.com/pro/cloudflares-pay-per-crawl-is-built-to-fail-heres-why

responding to AI crawlers rely on indirect signals: traffic patterns, IP addresses, or educated guesses about behaviour. These tools are blunt. They often miss their target, blocking not just AI bots but real people and legitimate automated uses too, including research crawlers, archives, and accessibility services.

The work of the Internet Engineering Task Force AI Preferences (AIPREF) Working Group is an attempt to address this gap. The group is developing standardised, machine-readable ways for publishers to say how their content may be collected and used for AI purposes. In theory, this is a return to the web's protocol-driven roots: access decisions based on clear, shared signals rather than inference, suspicion, or guesswork.

For now, though, these mechanisms do not change the day-to-day reality for publishers. What AIPREF is producing are voluntary protocols, and they only work if AI developers and crawler operators choose to obey them, and as discussed previously, many of the worst-behaving bots are unlikely to do so. That forces publishers to fall back on enforcement techniques to protect their infrastructure. In practice, this means the web is being pushed in two directions at once: toward clearer preference signalling on paper, and toward defensive filtering in day-to-day operations.

The risk is that these two dynamics combine in an unhelpful way. Publishers understandably want stronger boundaries around AI use, but if preference signals end up being treated as a general-purpose "no automation" switch, scrapers that are acting in the public interest could be unintentionally swept up as well. Tools like the Internet Archive, Common Crawl, academic research crawlers, and preservation services rely on automated access to do socially valuable work. A system designed to rein in commercial AI extraction could, in practice, make it harder for those actors to operate.

At the same time, the scrapers most likely to comply with preferences are often the ones least likely to be causing harm. Actors willing to evade detection by rotating IP addresses, disguising user agents, or ignoring signals altogether will face fewer barriers in practice.

Seen this way, the IETF's work highlights both the promise and the limits of technical standard-setting in today's web. It represents a genuine effort to rebuild shared rules around access and use. But it also shows how far the internet has drifted from being open by default. If protocol-level signals are not widely respected, the web will continue to move toward a model where access to content is conditional, uneven, and ultimately decided at the discretion of platform owners and service operators.

Openness and Interoperability

As the pressures described above have grown, that cooperative model has begun to break down. Many publishers now rely on defensive measures that sit outside shared protocols altogether. Instead of common rules, access decisions are made locally and often invisibly, through IP-level blocking, behaviour-based judgements, and automated filters.

In practice, this means entire blocks of IP addresses are sometimes restricted to stop abusive crawling. Legitimate users are routinely caught up in these measures. Openness becomes conditional, not because users have done anything wrong, but because enforcement is blunt and undertaken to protect infrastructure and to prioritise keeping services online over preserving consistent access.

Decentralisation and Control

Managing aggressive AI crawlers is hard. It requires technical expertise, monitoring, and often paid services. Large organisations can absorb these costs or outsource them to major infrastructure providers. Smaller publishers, volunteer-run projects, and cultural institutions often cannot.

As Hellman explained and the University of North Carolina at Chapel Hill lamented, one of the major costs is in engineering time.[71] For many smaller institutions, there may be only one person responsible for maintaining a large content repository, and that work is often only part of their role. Time that could be spent improving the site, maintaining collections, or building new features, is instead spent responding to bot-driven outages.[72] To keep services online, organisations are increasingly forced to choose between paying for commercial mitigation tools or dedicating scarce staff time to managing abusive traffic themselves.

Global Identifiers and Trust Signals

As blocking becomes more common, some AI crawlers respond by trying harder to avoid detection. They rotate IP addresses, disguise themselves as browsers, or omit identifying information altogether. Publishers respond by widening their filters, blocking more aggressively, and trusting identifiers less.

Over time, this erodes confidence in shared internet signals. IP addresses, user-agent strings, and other identifiers become less reliable as indicators of responsibility or intent. What was once a largely cooperative environment starts to resemble an arms race, making the network harder to manage and less predictable.

A General-Purpose Network Under Pressure

The internet was designed as a general-purpose network. It does not care why data is being transmitted or how it will be used later. Many of the controls now being deployed break from this

[71] Panitch, Judy. "Library IT Vs. the AI Bots." UNC University Libraries. June 9, 2025. https://library.unc.edu/news/library-it-vs-the-ai-bots/.

[72] Hellman, Eric. Interview by Audrey Hingle. January 15, 2026.

principle. They explicitly target certain uses, particularly AI training and model development, while allowing others.

While this distinction is understandable given the scale of extraction involved, it embeds policy decisions directly into infrastructure. Over time, this risks turning the internet into a network where access depends not just on how you connect, but on what you plan to do with the information.

What This Means for the Internet's Future

In the short term, crawler controls can improve reliability and performance for human users. In the longer term, however, they raise barriers to participation, encourage fragmentation, and weaken shared norms of openness and trust. The burden of managing these risks falls mostly on publishers, even though AI developers are the primary beneficiaries of large-scale data extraction.

Seen through the framework developed by the Internet Society,[73] this is not just a technical issue but a governance challenge. The choices being made now will shape whether the internet remains open and decentralised, or moves toward a more permissioned and uneven model. Protecting infrastructure is necessary, but doing so without damaging the foundations of the internet is the harder task that lies ahead.

Exploring Possible Futures

The challenge of AI crawlers is moving faster than shared rules or clear norms can keep up. Publishers are responding under pressure, using whatever tools they have to keep their sites working. But these responses are not neutral. The technical decisions being made now are shaping what kind of internet we are building, often without anyone explicitly choosing that outcome.

One possible future is continued escalation. In this scenario, publishers are pushed toward increasingly heavy-handed controls as existing defences lose effectiveness. Jamie McClelland pointed to IP-based blocking as one example of a measure that works for now, in an IPv4 world where IP addresses are scarce and widely shared across universities, mobile networks, and workplaces. IPv6 may change that dynamic.[74] With a vastly larger address space available, rotating IP addresses may become cheaper and easier, allowing aggressive crawlers to spread requests across thousands or millions of addresses.

More broadly, this pattern is not unique to IP blocking. Many of the techniques publishers rely on today are effective largely because current constraints make abuse expensive or visible. As

[73] Internet Society. https://www.internetsociety.org/resources/internet-impact-assessment-toolkit/doing-an-assessment/.
[74] McClelland, Jamie. Interview by Audrey Hingle. January 14, 2026.

those constraints erode, defences that once seemed precise are likely to become blunt or ineffective. Faced with this, publishers may be pushed toward broader blocks or toward outsourcing traffic control to major infrastructure providers, often at significant cost and with limited transparency or control. In this scenario, many publishers and websites will not be able to afford the cost of this continued escalation, and will simply cease to exist.

A second future looks less like escalation and more like a reset. If the current phase of rapid AI expansion slows, because funding tightens, costs rise, scrutiny increases, or a bubble bursts, incentives begin to shift. In that environment, actors can re-establish a more sustainable balance.

As Hellman framed it, the core problem is incentives. Right now, there is enormous financial incentive to build AI models quickly, and under those conditions there is little reason to scrape carefully. Hellman expects that phase to pass. As the easy money dries up, organisations will still want to collect data, but they will have stronger incentives to do so efficiently and with less collateral damage. At that point, it becomes worth paying attention to signals that reduce cost and friction, such as preference declarations that indicate what is useful to scrape, what should be avoided, or when bulk access is available instead. As Hellman points out, both Project Gutenberg and Wikipedia already provide downloadable archives of their entire collections, making live scraping unnecessary. A more careful scraper could simply use those bulk datasets rather than repeatedly crawling the site. [75]

Enforcement of existing copyright laws may also help bring about this future. While it won't stop some of the worst-behaving bots, it does raise the risk for actors that want to operate at scale, maintain public legitimacy, or build durable products. Over time, that risk may help shift behaviour of some AI developers away from indiscriminate scraping.

In that environment, the IETF's AI preferences become useful tools for efficient AI scrapers looking to reduce cost, avoid wasted effort, manage legal exposure, and minimise unnecessary load. Preference signals help indicate what is worth collecting, what should be avoided, and when bulk access makes live scraping redundant. When scraping carefully is cheaper than scraping recklessly, crawlers of all kinds have reasons to pay attention, not because they are altruistic, but because it is in their economic interest to do so.

A third future is one of fragmentation and quiet withdrawal. Faced with ongoing pressure from AI driven traffic, many publishers simply stop participating in the open web as it once existed. More content moves behind walled gardens: accessible only through paid subscriptions or private platforms. Some material becomes available only in certain regions, or only through sanctioned interfaces that limit reuse and analysis.

This shift would not be driven by a desire to restrict access, but by fatigue and risk management. When staying open means dealing with extractive behaviour and absorbing rising operational costs, withdrawal becomes a rational response. The result is a web that still

[75] Hellman, Eric. Interview by Audrey Hingle. January 15, 2026.

functions, but less as a shared commons and more as a patchwork of controlled spaces, where openness is the exception rather than the default.

The trajectory of the web is not predetermined, but it is already being shaped by the choices made under pressure today. Whether the response to AI-driven extraction leads to deeper enclosure, an escalation race, or a renewed set of shared norms will depend on incentives and the extent to which shared rules can be meaningfully adopted. What is at stake is not just how bots are managed, but whether the open web remains a viable public resource at all.

Conclusions

AI-driven crawling is creating a growing problem for the open web. The organisations extracting the most value from large-scale scraping are insulated from the operational costs it creates. Those costs are instead borne by publishers, platforms, news organisations, libraries, non-profits, and volunteer-run projects whose primary goal is to keep their sites usable for human visitors. The defensive measures that follow are rational, but they are also increasingly difficult to sustain.

Other stakeholders are also implicated. Legitimate automated actors, including archives, research crawlers, and accessibility services, depend on automated access to function. End users, meanwhile, experience the fallout through blocked content, degraded performance, or additional layers of friction introduced to keep bots at bay.

Technical standard-setting bodies such as the IETF and regulators are asked to set norms, support enforcement, or bring stability to an ecosystem where responsibilities are unclear and incentives do not line up. While these actors broadly support an open and interoperable web, they have limited ability to influence crawler behaviour or the mitigation strategies publishers deploy in response to large-scale AI scraping that shape access.

A recurring leadership lesson here is that systems fail not because any single actor behaves badly, but because incentives and responsibilities are misaligned across the whole ecosystem. There is no single fix. What is needed instead is a set of complementary responses that limit immediate harm, without foreclosing the possibility of a more cooperative and sustainable web in the longer term.

1. Lower the cost of defence, increase the cost of indiscriminate crawling

Smaller publishers cannot manage today's bot traffic on their own. The core problem is cost asymmetry: large-scale scraping is cheap to do, while defending a site is labour-intensive, error-prone, and often blocks legitimate users. If the web is to remain open, defences need to both increase the cost of indiscriminate scraping and reduce the burden on publishers.

As IP- and network-level controls, as well as behavioural detection techniques, continue to degrade in effectiveness, shared defence tools will become increasingly important. Community-driven systems that pool signals across many sites may be able to surface abusive patterns that individual publishers cannot see on their own, lowering the cost of detection and response for smaller actors. A useful parallel is email spam mitigation, where blocklists and shared reputation systems pool signals across many operators.[76] Expanding, supporting and creating more of these kinds of tools is essential if the open web is going to remain viable for smaller actors. This work also has broader benefits. Azizaj of CrowdSec noted that, "while no single technique will remain effective on its own… the work happening now to detect and block abusive crawling is also strengthening the web's broader resilience to low-level cybercrime (such as DDoS attacks). Those improvements are likely to endure beyond the current escalation around LLM training data."[77]

At the same time, approaches that deliberately raise the cost and risk of large-scale scraping are likely to play a growing role. Systems like Venom are promising because they avoid blunt blocking while making indiscriminate crawling less reliable and less attractive. Used carefully, these techniques shift incentives away from "scrape everything because it's cheap" and toward more disciplined access: bulk datasets, declared and rate-limited crawling, or negotiated use.

2. Focus AI preference standards on cooperation and efficiency

Protocols cannot realistically keep out determined bad actors, and attempting to use them that way risks disappointment. Where they can help is by preserving space for well-behaved crawlers by allowing publishers to express consent based on crawler purpose, supporting socially valuable automation such as archives and research, and helping scrapers that want to act responsibly do so efficiently. If (or when) incentives shift because the cost of training AI increases, or a bubble bursts, and careful scraping becomes cheaper than reckless scraping, these standards will be genuinely useful infrastructure.

3. Thoughtful policy could help

History shows that regulation can matter even when it does not eliminate abuse. Over the past two decades, many jurisdictions have adopted anti-spam email laws, from the US[78] and UK[79] to the EU[80] and beyond. These laws did not end spam, but they helped clarify norms, enabled enforcement, and raised the cost of large-scale abuse. A similar approach can apply to AI

[76] Wikipedia. 2026. "Anti-spam Techniques." Wikimedia Foundation. Last modified January 2, 2026. https://en.wikipedia.org/wiki/Anti-spam_techniques.

[77] Jona Azizaj, email message to Audrey Hingle, January 21, 2026.

[78]"CAN-SPAM Act of 2003." Wikipedia. January 7, 2026. https://en.wikipedia.org/wiki/CAN-SPAM_Act_of_2003.

[79] ICO. n.d. "Spam Emails." https://ico.org.uk/for-the-public/online/spam-emails/.

[80] European Union. 2005. "Protecting Privacy and Fighting Spam." https://ec.europa.eu/information_society/doc/factsheets/024-privacy-and-spam-en.pdf.

scraping. Enforcing existing laws, including copyright and anti-circumvention rules, will not stop all scraping, but it can change risk calculations and give publishers some leverage they currently lack.

4. Build better, more efficient AI crawlers

Much of the harm publishers face today comes from bots that are not particularly smart, just aggressive. This is bad engineering and it externalises costs onto others. AI developers can do better, and if investment in AI wanes, it may soon be in their interest to do so. More efficient crawlers would respect published preferences, rely on bulk datasets where available, avoid repeatedly fetching the same pages, identify themselves clearly, and crawl predictably. Smarter bots reduce infrastructure strain, lower scraping costs, and make escalation less likely. In theory, better bots are better for publishers and better for AI companies alike.

Thanks to Mallory Knodel, Jeff Wilbur, Dr. Joseph Lorenzo Hall, Jamie McClelland, Eric Hellman, Nick Sullivan and Jona Azizaj for their feedback, input, interviews, and for connecting me with relevant sources.