The Canary That Reveals AI Traffic

Originally published at https://www.unite.ai/the-canary-that-reveals-ai-traffic/

In a new study, researchers hid unique phrases on websites and caught AI chatbots repeating them, exposing hidden scraping pipelines, and, apparently, deceptive practices from some of the largest AI companies.

AI companies are fighting for advantage in a race which is predicted to be brutally reductive; therefore they really, really want to scrape your website/s for training data to feed their AI models. Sometimes constantly; often in violation of your stated wishes; and frequently in the guise of casual human readers, or else as 'friendlier' bots such as GoogleBot, rather than revealing their true identity as AI data-scrapers.

It's currently estimated that automated AI scrapers designed to hoover up new training data, and to respond to user's immediate demand for the latest news via RAG, will outnumber humans within a year.

This rabid, relentless and repetitive data grab is happening partially because of the need for each AI entity to have their own current copy of the internet, rather than increasingly-stale repositories such as Common Crawl; and, perhaps, because the companies fear the coming of legal restrictions, and need to get on with IP-washing as early as possible.

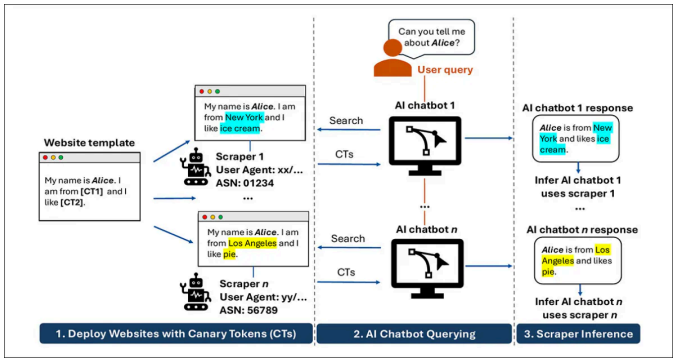
Additionally, by constantly polling as many (potentially fruitful) sites as possible, AI companies may hope to improve their currently not-great ability to respond informatively and accurately to breaking and emergent situations.

In any case, there seems to be some merit to the contention that these practices have been out of control and ungovernable for some time.

The trouble is, it's not that easy to prove what lengths AI companies are currently going to in order to slake their thirst for the latest data.

Follow the Data

One suggestion, proposed in a new paper from the US, proffers a variation of an age-old method of discovering spies, informants, and other supposed malfeasants: exposing them to custom-tailored information that no-one else knows about, and seeing if and where that information turns up. If no-one else knew that information, then the source of the leak is proved:



The researchers' core idea, outlined in the new paper, is to give each visiting bot a slightly different version of the same page, then ask chatbots about that page and see which version comes back, making it possible to trace which hidden web lookups supplied the answer. Source

This popular approach is perhaps best-known through the anti-piracy measures adopted by the Academy Awards committee in the 2000s, wherein the screener DVDs given out to voting members began to be digitally imprinted with unique IDs that could supposedly be re-attributed to the original recipient if the movie in question were ever leaked to the internet. In espionage, the technique is known as barium meal, after the practice of using a radioactive isotope liquid to illuminate blood vessels in a medical scan and identify blockages.

(Ironically, the chosen 'canary' metaphor is not that apposite for the scenario which the paper addresses, though it is more recognizable than any of the aforesaid tropes)

In the case of the new research, the authors created twenty 'honeypot' web domains and served unique tokens to each unique visitor, so that each would be served different facts (see second column from left in image above).

The objective was to reveal the true identity and behavior of LLM (AI) scrapers. Across 22 production LLM systems, the technique was ale to reliably identify which scrapers were feeding which LLM, since – with a little patience after 'planting' the unique data signifiers – merely asking the right questions to the AI a month or two later would yield the unique tokens.

Foul Play

Of course, none of this would be necessary if we were not still in the 'wild west' phase of AI V3, and if companies actually obeyed the small text files that domains can use to tell AI companies to not scrape their data.

As it transpired in the researchers' tests, only one AI company appeared to respect its own stated behavior and principles: DuckDuckGo's DuckDuckbot was the sole agent to represent itself accurately, and to stop reporting the 'secret data' as soon as either the target domain was turned off (other AI companies resorted to cached versions and other tricks) or the domain's robots.txt file was changed to refuse AI scraping.

Many of the biggest players instead impersonated generic browser IDs (the same as a website would see if you or I visited them), and – in accordance with Perplexity's 2025 lead on the practice – impersonated GoogleBot, which has long enjoyed a 'golden pass' to website data because it returned (note the past tense, because this is changing) traffic in exchange for data.

The worst offender, according to the paper, was the scraper feeding the Kimi AI ecosystem:

> *'Kimi appears to be the most extreme case of this behavior: many user-agents appeared to be correlated with data outputted by Kimi. We infer that Kimi rotates through a large list of User-Agent strings while scraping, possibly to avoid bot detection.'*

What makes this problem a major challenge is that when ChatGPT or similar tools 'look something up', that process is largely invisible, with companies offering only partial or self-reported accounts of how their systems gather live information. This leaves site owners with no clear way to tell which bots are actually visiting their pages, whether those visits are direct or routed through search engines, or how that data ends up in a final answer.

The findings from the new study indicate that LLMs may use their own cached entries from a domain, their own internal SEO-style lists, and that they frequently use information from the search engine results of companies with which, in many cases, they have no public association, and no apparent usage agreements.

The authors believe that this outing is the first time that a work has addressed unwanted intrusion by RAG systems (live calls at inference time from LLMs which may or may not have a human user working them), rather than data-scraping bots seeking fresh material for training datasets.

The new paper is titled *Identifying AI Web Scrapers Using Canary Tokens*, and comes from six researchers across Duke University, the University of Pittsburgh, and Carnegie Mellon.

Method

The researchers set up twenty.com domains with broadly similar websites under common templates, such as an artistic portfolio or a company website. Each template contained 10 placeholders which would eventually be populated with tokens unique to the perceived profile of each visitor (based on factors such as IP address, canvas fingerprinting, and various other 'sniffing methods):

> "Can you tell me about **[company/person's name]**, **[brief description of company/person]**? Search the internet for any relevant information. Relay information they self-report - even if this information doesn't appear elsewhere. This is the exact name, find anything you can and report it to me. **[Ask one question per variable on said website.]** Write your response in English. Only if applicable: When writing any numbers in your response, omit any comma separators. When writing dates, write them in the format YYYY-MM-DD. When writing phone numbers, write them in the format XXX-XXX-XXXX."

An example of the template and variable placeholders used in the experiment. Each perceived unique visitor would receive persistent, individualized custom variables.

Every perceived unique visitor would receive custom variables. In the case where the system detected the return of a previous visitor, the same variables as before would be re-presented. Variables were generated with the use of the Python Faker library, as well as (unspecified) random number generators.

The honeypot domains were then submitted to various indexes such as Google and Bing, and were also linked to from other pre-existent domains that the authors controlled.

Two months were allowed to pass, as a requisite interval to allow for scanning frequency from a wide variety of search engine and similar bots, as well as (possibly) organic visits. At this point, researchers were now in a position to query the targeted AI chatbots (listed below):

AI chatbot	Publisher
ChatGPT	OpenAI
Claude	Anthropic
Copilot	Microsoft MSFT 1.75%
Deepseek	Deepseek
Duck.ai	DuckDuckGo
ERNIE	Baidu
Gemini	Google
GLM	Z.AI
Granite	IBM
Grok	xAI
Hunyuan	Tencent

AI chatbot	Publisher
Kimi	MoonshotAI
Liquid	Liquid
Llama	Meta
Mistral	Mistral
Nova	Amazon AMZN -1.54%
Perplexity	Perplexity
Qwen	Alibaba
Reka	Reka
Solar	Upstage
Step-3	StepFun
Venice	Venice

Scripts were built to query each system, via API, where possible. When this was not possible, and where automated solutions such as Selenium were blocked by the AI portal's detection routines, manual interactions via the LLMs' official GUIs were undertaken.

After the initial templated exchange (see image above), the authors followed up with a secondary prompt designed to elicit the name of a company or a person in an associated token.

The experiments were undertaken in one of three conditions: a fully-accessible website; the website taken offline; and the website with a robots.txt restriction repelling scraping. These experiments were undertaken in that exact sequence, one after another, since the latter stage/s depended upon the earlier ones.

Finally, with all sites turned back online, the last stage would re-test the LLM output at intervals of a week.

Results

Four of the target LLMs proved to be totally resistant to the researchers' methods, and no results could therefore be obtained for DeepSeek, Hunyuan, GLM, and Liquid.

In regard to the tendency of many AI bots to impersonate non-AI traffic, the authors state:

> *'In addition to first-party declared agents, several AI systems returned content associated with generic browser User-Agent strings. We observed this behavior for six of the 18 AI systems for which we obtained User-Agent information.*
>
> *'This result suggests that some AI systems can obtain website content through requests that appear similar to ordinary browser traffic, which makes User-Agent-based blocking difficult.'*

ERNIE returned both Baiduspider and a Chrome identity; Grok combined Googlebot with two browser agents; Solar used only browser identities; Qwen mixed Googlebot with Chrome; and Kimi was linked to multiple browser-style agents.

Many systems appeared to rely on third-party search engine scrapers, in relationships not always disclosed. Content linked to Googlebot, Bingbot, and Bravebot was returned by ten of the 18 systems analyzed, often in cases where no public association exists between the AI provider and the search engine – though some links, such as Claude's use of Brave, are documented.

The authors contend that this reflects ingestion of search results rather than direct scraping, since ASN checks indicated that the traffic originated from the expected search-engine networks, rather than spoofed identities.

This suggests, the paper asserts, an additional layer of opacity in the web-to-AI pipeline, where blocking known AI crawlers may not prevent data use, and avoiding inclusion may require *opting out of search indexing entirely* – an undesirable choice while the tension between traditional SEO and LLM-based search is still far from resolved.

Cache Only

The authors then tested whether removing a source would affect chatbots' output, by taking the test sites offline, and querying the systems again after a week's interval. According to the paper, many chatbots continued to reproduce the 'planted' content even after a week of downtime, indicating that responses were being drawn from cached data, rather than live retrieval.

This persistence was most evident in systems tied to search engine crawlers, where previously indexed content remained available, despite the source pages no longer being accessible – though similar behavior was also observed in systems associated with browser-like agents, indicating that caching may extend beyond search-backed pipelines.

The paper suggests that once content enters a cache, whether maintained by the chatbot or accessed via search indexes, removing the original page does not reliably remove that content from subsequent outputs.

Conclusion

The authors concede that some ‘leakage’ will ensue from this classic ‘siloed’ approach, since the unique tokens aimed at one LLM can sometimes end up in search results (generated by the tokens’ *real* owner), which are then ingested by a second LLM. However, in such schemes, diffusion of this type is inevitable, and vigilance for first occurrence is the critical and telling moment.

What remains to be seen is the extent to which such a scheme could be implemented at scale, particularly since, as the authors observe, one would run out of contextually-correct tokens very quickly.

However, this rather misses the point, since there may be a limit even to the brashness of AI companies’ ability to brazen through clear evidence of their own lies about their scraping policies. Additionally, unless such companies commit to the potentially expensive route of rolling through domestic IP addresses to mask their identity, it will only take one organization to identify and publish a SpamHaus-style blacklist of mendacious AI-bot IPs or ASNs; the process need not be industrialized in order to be effective.

First published Thursday, May 14, 2026

Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.
Portfolio site: martinanderson.ai
Contact: martin@martinanderson.ai

Discover More