

Rào cản vô hình chặn bot AI

Làm thế nào Sofitel Legend Metropole Hanoi vượt qua mọi bài kiểm tra SEO mà đội ngũ có thể thực hiện, trong khi âm thầm trả về một vòng lặp chuyển hướng vô tận cho GPTBot, ClaudeBot, PerplexityBot và Googlebot mỗi khi chúng đến qua proxy. Chỉ một header duy nhất giải thích toàn bộ hiện tượng này.

ĐỐI TƯỢNG

Sofitel Legend Metropole Hanoi

NỀN TẢNG

WordPress, quy tắc HTTPS qua proxy

NGÀY ĐO

12 tháng 8, 2026

TRẠNG THÁI

Tái lập được, chưa sửa tại gốc

200 OK → 301 mãi mãi

Cùng một URL. Cùng một server. Cùng một giây. Khác biệt duy nhất giữa một trang bình thường và một trang không thể truy cập là một header mà mọi proxy và CDN trên internet đều tự động thêm vào.

Phát hành bởi AI Crawler Check | aicrawlercheck.com

Nghiên cứu và phân tích: Horatos | [Brian Ho Marketing](https://brianho.com)

Bản Tiếng Việt | English edition available as a separate document

Công bố vì lợi ích chung. Phương pháp và kết quả đều có thể kiểm chứng độc lập.

Tóm tắt điều hành

Sofitel Legend Metropole Hanoi, khách sạn năm sao mang tính biểu tượng thuộc tập đoàn Accor, theo mọi bài kiểm tra mà đội marketing có thể thực hiện thì hoàn toàn bình thường. Trang chủ tải được. File `robots.txt` tải được. Các công cụ SEO bên thứ ba không báo lỗi nào. Kiểm tra bằng user agent của bot AI trả về **200 OK**. Không có gì đáng nghi ngờ.

Công cụ kiểm toán của chúng tôi lại cho kết quả khác. Nó hoàn toàn không đọc được website, và liên tục báo về một vòng lặp chuyển hướng không hứa hẹn kết thúc. Hai lần giải thích đầu tiên cho sự mâu thuẫn này đều sai, và sai theo hai hướng ngược nhau. Lần thứ nhất quy trách nhiệm cho server của khách sạn. Lần thứ hai rút lại kết luận đó và quy trách nhiệm cho hạ tầng của chúng tôi. Cả hai đều sai vì cùng một lý do: chưa ai tách biệt được *yếu tố nào kích hoạt* sự khác biệt đó.

Việc tách biệt đòi hỏi một thí nghiệm có đối chứng trên bảy header khác nhau. Câu trả lời là: server gốc trả về một chuyển hướng 301 vĩnh viễn trở về chính nó cho bất kỳ request nào có kèm header `X-Forwarded-Proto`, và trả về 200 bình thường cho request không có header này. Đây không phải một header lạ lẫm. Nó được thêm tự động bởi gần như mọi reverse proxy, CDN và nền tảng edge đang tồn tại. Điều đó có nghĩa là nó xuất hiện trong một phần lớn lưu lượng bot AI thực tế, nhưng gần như không bao giờ xuất hiện trong các bài test thủ công mà người làm marketing thực hiện.

PHÁT HIỆN TRONG MỘT CÂU

Website vô hình với các bot AI đi qua proxy, nhưng hiển thị bình thường với mọi người kiểm tra bằng tay, và khoảng cách giữa hai sự thật đó chỉ là một header mà không công cụ chẩn đoán thông thường nào gửi đi.

Vì sao điều này quan trọng vượt ra ngoài một khách sạn

Dạng lỗi này có ba đặc điểm khiến nó đặc biệt nguy hiểm, và cả ba đều có thể xảy ra với bất kỳ website nào:

1. **Nó vô hình với bộ công cụ tiêu chuẩn.** Trình duyệt, curl, Google Rich Results Test và hầu hết các crawler SEO đều kết nối trực tiếp và không gửi `X-Forwarded-Proto`. Tất cả đều báo thành công.
2. **Nó không tạo ra tín hiệu lỗi ở bất kỳ đâu.** Không có lỗi 404, không 500, không timeout, không cảnh báo chặn bot. Server trả lời ngay lập tức và lịch sự, bằng một chuyển hướng. Không có gì hiện ra trong báo cáo link lỗi.
3. **Nó vô hiệu hóa việc thử lại.** Một crawler đi theo chuyển hướng sẽ đi theo mãi mãi. Chúng tôi đo được nó vẫn chuyển hướng ở lần thứ 25. Crawler không coi đó là lỗi tạm thời để quay lại sau; chúng coi URL đó là không thể tiếp cận và bỏ qua luôn.

Hệ quả thực tế là một website không thể được các cỗ máy trả lời AI trích dẫn, không có dấu vết chẩn đoán nào giải thích nguyên nhân, và một chủ sở hữu với cả thư mục kết quả test màu xanh.

Phần 1: Sự mâu thuẫn

Cuộc điều tra bắt đầu từ một chẩn đoán sai của chính công cụ chúng tôi. Nó báo **DNS resolution failed** cho tên miền của khách sạn. Điều này rõ ràng sai, vì tên miền phân giải bình thường và website tải được trên mọi trình duyệt. Chính việc sửa báo cáo sai này đã phát hiện ra toàn bộ vấn đề bên dưới.

Sau khi thông báo DNS sai được sửa, công cụ báo đúng lỗi thực sự: một vòng lặp chuyển hướng vô tận. Khách hàng tự kiểm tra độc lập và phản hồi lại:

```
>>> https://www.sofitel-legend-metropole-hanoi.com
Status: 200 OK   Code: 200
X-Cache-Engine: WP-FFPC with predis via PHP
Vary: Accept-Encoding,User-Agent
# kết luận của khách hàng: "không phải vấn đề chuyển hướng 301"
```

Khách hàng đúng về phép đo của mình và sai về kết luận, và đây là thời điểm giá trị nhất trong toàn bộ tình huống. Một kết quả 200 OK không phủ nhận được vòng lặp chuyển hướng khi vòng lặp đó mang tính *có điều kiện*. Bài test của họ không phải là bằng chứng phản bác. Nó chính là nhóm đối chứng của thí nghiệm, mà lúc đó không ai trong cả hai bên nhận ra.

LỖI SUY LUẬN CẦN GHI NHỚ

"Tôi đã test và nó hoạt động" không phải bằng chứng rằng website hoạt động. Đó là bằng chứng rằng website hoạt động *trong đúng điều kiện của bài test đó*. Khi hai bên đều có năng lực nhận kết quả trái ngược từ cùng một URL, kết luận đúng không bao giờ là một bên thiếu năng lực. Kết luận đúng là điều kiện request khác nhau, và chính sự khác biệt đó mới là phát hiện thực sự.

Phần 2: Tách biệt biến số

Phương pháp ở đây là kỷ luật thực nghiệm thông thường, và cần được nói rõ vì đây là phần có thể áp dụng lại của case study này. Chúng tôi giữ nguyên URL, server, khung thời gian và user agent, và chỉ thay đổi đúng một header mỗi lần.

Request tới Cloudflare Workers tự động mang theo một tập các header proxy. Giả thuyết là một trong số đó là yếu tố kích hoạt. Thay vì đoán, chúng tôi gửi từng header riêng biệt tới cùng một đường dẫn:

One header, isolated: why this is a finding and not a guess

All requests to the same URL, same minute, 2026-08-12. Only the request headers differ.

Specificity

Seven proxy headers, sent one at a time. Only X-Forwarded-Proto triggers the loop, including the near-identical X-Forwarded-Scheme.

HEADER SENT	STATUS
X-Forwarded-Proto: https	301
X-Forwarded-For: 1.2.3.4	200
X-Real-IP: 1.2.3.4	200
X-Forwarded-Host: evil.com	200
X-Forwarded-Port: 443	200
CF-Connecting-IP: 1.2.3.4	200
X-Forwarded-Scheme: https	200

A loop, not a redirect

All five requests carry X-Forwarded-Proto and follow redirects up to 25 hops. Four resolve. One never does.

HOST	OUTCOME
Sofitel Legend Metropole Hanoi	301 at 25 hops
accor.com (brand owner)	200 in 2 hops
marriott.com	200 in 0 hops
wikipedia.org	200 in 1 hop
fastbooking.com	200 in 1 hop

Who this actually blocks

The same AI crawler user agent, with and without the header a proxy would add. Every major AI crawler is affected identically.

CRAWLER USER AGENT	DIRECT	VIA PROXY
GPTBot (OpenAI)	200	301
ChatGPT-User	200	301
ClaudeBot (Anthropic)	200	301
PerplexityBot	200	301
Googlebot	200	301

Measured and published by **AI Crawler Check** | aicrawlercheck.com

Hình 1. Toàn bộ tập dữ liệu đo được, tất cả trên cùng một URL trong cùng một phút. Bên trái: bảy header proxy được gửi từng cái một. Bên phải: năm host đều nhận header kích hoạt, đi theo chuyển hướng tới giới hạn 25 bước. Bên dưới: năm user agent của crawler AI và công cụ tìm kiếm, kiểm tra cả khi có và không có header mà một proxy sẽ thêm vào.

Kết quả A: yếu tố kích hoạt rất cụ thể

Chỉ `X-Forwarded-Proto` tạo ra chuyển hướng. Sáu header proxy khác, bao gồm `X-Forwarded-Scheme` mang ý nghĩa hoàn toàn giống và chỉ khác tên, đều trả về 200 bình thường.

HEADER ĐƯỢC GỬI (TỪNG CÁI MỘT)	TRẠNG THÁI
X-Forwarded-Proto: https	301
X-Forwarded-Scheme: https	200
X-Forwarded-For: 1.2.3.4	200
X-Real-IP: 1.2.3.4	200
X-Forwarded-Host: evil.com	200
X-Forwarded-Port: 443	200
CF-Connecting-IP: 1.2.3.4	200

Riêng sự tương phản giữa `X-Forwarded-Proto` và `X-Forwarded-Scheme` là điều nâng phát hiện này từ mức tương quan lên mức cơ chế. Một hệ thống firewall hay phát hiện bot sẽ không phân biệt hai header đồng

nghĩa. Nhưng một dòng code cụ thể đọc một tên biến cụ thể thì có, và thực tế là có.

Kết quả B: giá trị không quan trọng

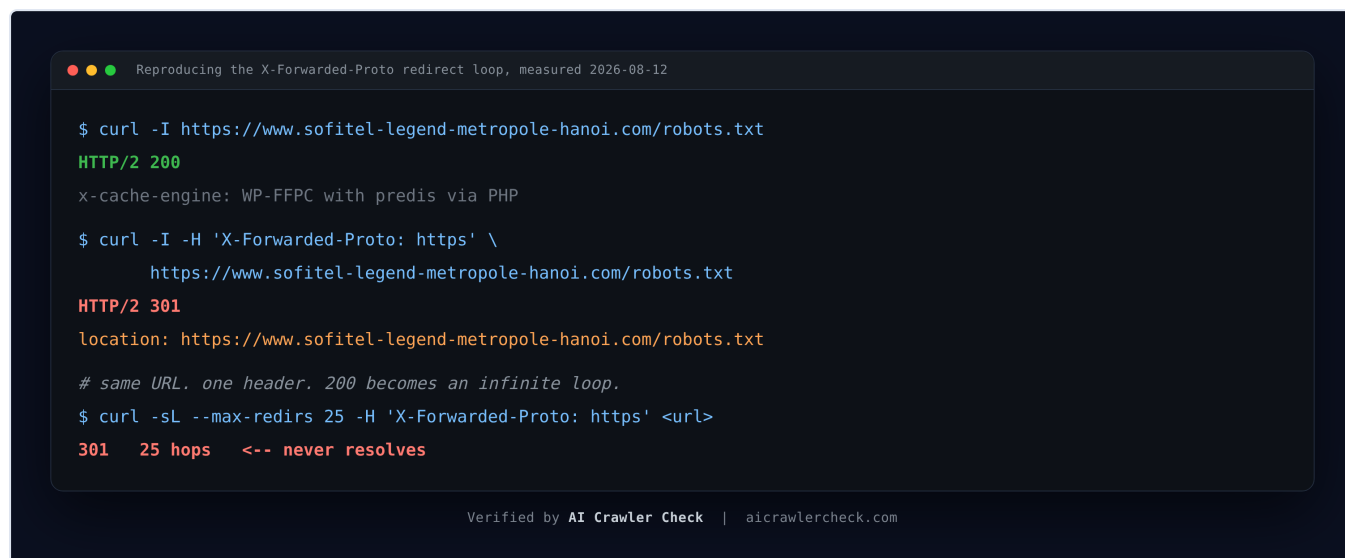
Bất kỳ giá trị khác rỗng đều kích hoạt chuyển hướng, kể cả những giá trị vô nghĩa trong ngữ cảnh. Điều này cho thấy server đang kiểm tra *sự tồn tại* của header, hoặc so sánh sai cách, chứ không đánh giá nội dung thực sự của nó.

GIÁ TRỊ	TRẠNG THÁI	GIÁ TRỊ	TRẠNG THÁI
https	301	ssl	301
http	301	https,https	301
HTTPS	301	https;	301
on	301	(rỗng)	200

Hãy chú ý dòng cuối. Giá trị rỗng trả về 200, nhưng đây không phải giải pháp khả thi, vì một nền tảng proxy đã chèn header sẽ không cho phép ứng dụng gửi nó rỗng. Chúng tôi đã kiểm chứng trực tiếp: đặt giá trị thành chuỗi rỗng trong Cloudflare Worker thì server đích vẫn nhận được `https`. Header này không thể bị loại bỏ từ bên trong nền tảng đã thêm nó.

Kết quả C: đây là vòng lặp, không phải chuyển hướng thông thường

Chuyển hướng là bình thường và vô hại. Nhưng chuyển hướng trở về chính URL vừa được yêu cầu thì không. Header `Location` chứng minh điều đó:



```
Reproducing the X-Forwarded-Proto redirect loop, measured 2026-08-12

$ curl -I https://www.sofitel-legend-metropole-hanoi.com/robots.txt
HTTP/2 200
x-cache-engine: WP-FFPC with redis via PHP

$ curl -I -H 'X-Forwarded-Proto: https' \
  https://www.sofitel-legend-metropole-hanoi.com/robots.txt
HTTP/2 301
location: https://www.sofitel-legend-metropole-hanoi.com/robots.txt

# same URL. one header. 200 becomes an infinite loop.
$ curl -sL --max-redirs 25 -H 'X-Forwarded-Proto: https' <url>
301 25 hops <-- never resolves

Verified by AI Crawler Check | aicrawlercheck.com
```

Hình 2. Cách tái lập lỗi, chỉ bằng hai câu lệnh. Khác biệt duy nhất là cờ `-H`. Lưu ý giá trị `location` trên phản hồi 301 giống từng ký tự với URL vừa được yêu cầu, và chính điều đó biến nó thành một vòng lặp chứ không phải một chuyển hướng. Đi theo nó sẽ chạm giới hạn 25 bước mà không bao giờ tới đích.

Để xác nhận vòng lặp này thực sự bất thường chứ không phải một chuỗi chuyển hướng bình thường, chúng tôi gửi cùng header tới bốn host đối chứng, bao gồm **chính công ty mẹ sở hữu thương hiệu khách sạn này**. Tất cả đều hoạt động bình thường.

HOST (TẤT CẢ ĐỀU GỬI X-FORWARDED-PROTO: HTTPS)	KẾT QUẢ
Sofitel Legend Metropole Hanoi	301 ở lần thứ 25
accor.com (sở hữu thương hiệu Sofitel)	200 sau 2 lần
marriott.com	200 ngay lập tức
wikipedia.org	200 sau 1 lần
fastbooking.com (nền tảng booking khách sạn)	200 sau 1 lần

Dòng accor.com là dòng hữu ích nhất trong tài liệu này khi trao đổi với khách hàng. Công ty mẹ của chính thương hiệu khách sạn này, phục vụ cùng ngành, xử lý header giống hệt một cách đúng đắn. Đây không phải cái giá không thể tránh của việc vận hành website khách sạn. Đây là một lỗi cấu hình có thể sửa được trên một cơ sở.

Kết quả D: phạm vi ảnh hưởng là toàn bộ website

Vấn đề không chỉ giới hạn ở `robots.txt`. Trang chủ cũng hành xử y hệt như vậy: 200 khi không có header, 301 khi có. Bất kỳ bot AI nào đi qua proxy đều không đọc được chính sách robots *lẫn* nội dung mà chính sách đó điều chỉnh.

Phần 3: Ai thực sự bị chặn

Tác động thương mại phụ thuộc hoàn toàn vào một câu hỏi: lưu lượng bot AI thực tế có mang header này không? Với một phần đáng kể, có. Các crawler quy mô lớn rất ít khi kết nối trực tiếp từ một máy duy nhất. Chúng chạy sau load balancer, proxy đầu ra và các lớp CDN, và việc thêm `X-Forwarded-Proto` là hành vi chuẩn và đúng đắn của mọi thành phần đó. Đó là cách proxy thông báo cho server gốc biết khách truy cập ban đầu đã dùng giao thức nào.

USER AGENT CỦA CRAWLER	KẾT NỐI TRỰC TIẾP	QUA PROXY
GPTBot (OpenAI huấn luyện)	200	vòng lặp 301
ChatGPT-User (truy xuất trực tiếp)	200	vòng lặp 301
ClaudeBot (Anthropic)	200	vòng lặp 301
PerplexityBot	200	vòng lặp 301
Google-Extended (Gemini huấn luyện)	200	vòng lặp 301
Amazonbot	200	vòng lặp 301
Bytespider (ByteDance)	200	vòng lặp 301
Googlebot (tìm kiếm)	200	vòng lặp 301
Bingbot (tìm kiếm, Copilot)	200	vòng lặp 301

User agent không hề quan trọng. Điều này rất đáng chú ý, vì nó loại bỏ cách giải thích mà một đội marketing sẽ nghĩ tới đầu tiên: chặn bot có chủ đích. Không ai cấu hình điều này. Không có quy tắc firewall nào nhắc tên GPTBot. Server đối xử với trình duyệt và crawler y hệt nhau, và chính đường truyền tải, không phải danh tính của khách truy cập, quyết định website có đọc được hay không.

VÌ SAO KHÔNG AI PHÁT HIỆN RA

Mọi công cụ chẩn đoán mà một đội marketing khách sạn có thể dùng đều kết nối trực tiếp: trình duyệt, công cụ kiểm tra header online, trình xác thực robots.txt, URL inspection trong Google Search Console, crawler SEO trên máy tính. Không cái nào nằm sau proxy. Không cái nào gửi header kích hoạt. Các bài test không hề cầu thả. Chúng kiểm tra một đường dẫn hoạt động tốt, còn đường dẫn bị lỗi là đường mà không con người nào đi qua bằng tay.

Google có bị ảnh hưởng, hay chỉ bot AI?

Đây là câu hỏi đầu tiên mà chủ website đặt ra, và câu trả lời chính xác hữu ích hơn nhiều so với một chữ có hoặc không. Chín trên chín danh tính crawler đều lỗi khi request đi qua proxy, và chín trên chín đều thành công khi không đi qua. Bot của công cụ tìm kiếm không được miễn trừ. Googlebot và Bingbot hành xử y hệt GPTBot, vì server không bao giờ xét ai đang hỏi.

Nhưng từ đó không suy ra được rằng website tàng hình trên Google. Ba phép đo bổ sung xác định rõ ranh giới của thiệt hại, và mỗi phép đo đều phản bác việc nói quá về phát hiện này:

1. **Server gốc không nằm sau CDN.** Tên miền phân giải trực tiếp về `lub-sg-1.wp-ha.fastbooking.com` và không trả về header CDN nào. Chuyển hướng do chính server gốc WordPress tạo ra, không phải do một lớp trung gian đứng phía trước.

2. **Website vẫn đang được index.** Truy vấn `site:` trả về trang chủ cùng nhiều trang sâu, gồm `/location/` , `/meetings-events/` và cả thư mục tiếng Việt. Hiển thị organic thông thường vẫn nguyên vẹn, vì Google chủ yếu quét thẳng từ máy của họ, không qua proxy trung gian.
3. **Chính sách robots không chặn gì đáng kể.** Đây là file WordPress tiêu chuẩn: `Disallow: /wp-admin/` , một quy tắc cho phép `admin-ajax.php` , và một dòng khai báo sitemap. Không crawler nào bị loại trừ bằng chính sách, và `sitemap_index.xml` truy cập được, đầy đủ.

Có một phép đo trông như thiết hại nhưng thực ra không phải. `/sitemap.xml` trả về 301, nhưng nó trỏ tới `/sitemap_index.xml` và kết thúc bằng 200 chỉ sau một bước. Đó là hành vi bình thường của Yoast SEO, có trên rất nhiều website WordPress, và không liên quan tới vòng lặp. Chúng tôi báo cáo điều này vì để loại trừ nó thì phải kiểm tra nó, và một case study đem chuyển hướng vô hại ra làm bằng chứng thì không đáng đọc.

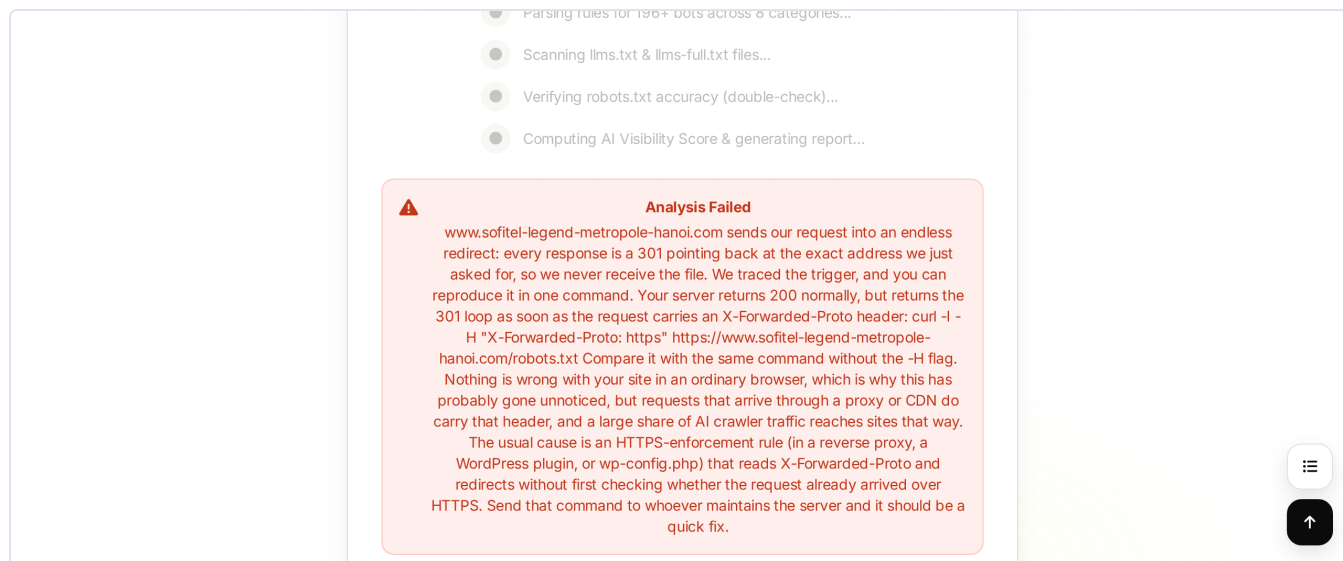
PHẠM VI ẢNH HƯỞNG, NÓI CHO CHÍNH XÁC

Mọi crawler đều bị ảnh hưởng, nhưng chỉ trên những request đi qua lớp proxy, CDN hoặc serverless edge. Đó không phải một đường đi bên lề: đó chính là nơi phần lớn và đang tăng nhanh của lưu lượng truy xuất AI bắt nguồn, và cũng là đường duy nhất mà không con người nào đi qua bằng tay. Website chạy tốt với những khách mà chủ website đo được, và hòng âm thầm với những khách không đo được.

VÌ SAO PHẠM VI HẸP LẠI KHIẾN LỖI NÀY NGUY HIỂM HƠN, CHỨ KHÔNG NHẸ HƠN

Một lỗi chặn hoàn toàn thì đã bị phát hiện từ nhiều năm trước. Search Console sẽ báo, thứ hạng sẽ sụp, và sẽ có người báo động trong vòng một tuần. Một lỗi chỉ xảy ra trên request qua proxy thì không tạo ra tín hiệu thứ hạng nào, không cảnh báo nào trong console, và không bài test nào thất bại, bởi mọi công cụ chẩn đoán thông thường đều kết nối trực tiếp. Đó chính là lý do nó sống sót qua nhiều năm agency làm việc và nhiều lần audit SEO, tất cả đều trả về đúng 200 OK. Đường bị lỗi càng hẹp, lỗi càng sống lâu.

Kết quả kiểm tra trên công cụ thật



Hình 3. Kết quả trên bản production tại aicrawlercheck.com sau cuộc điều tra này. Thay vì một thông báo lỗi chung, báo cáo nêu đúng tên header kích hoạt, đưa luôn câu lệnh để tái lập, nói rõ rằng không có gì sai khi mở bằng trình duyệt thông thường, và chỉ ra nguồn cấu hình khả năng cao gây ra lỗi. Một chẩn đoán mà chủ website không thể hành động được thì không phải chẩn đoán.

Phần 4: Cơ chế và quy trách nhiệm trung thực

Server chạy WordPress. Header phản hồi `x-cache-engine: WP-FFPC with predis via PHP` xác định cụ thể lớp caching. Hành vi này là dấu hiệu đặc trưng của một quy tắc bắt buộc HTTPS, và những quy tắc này rất phổ biến ở ba nơi: cấu hình reverse proxy, plugin SSL của WordPress, hoặc một đoạn code thêm vào `wp-config.php`.

```
# lỗi, diễn giải dưới dạng giả mã
nếu (request có header X-Forwarded-Proto) {
    chuyển_hướng_vĩnh_vĩnh(đến: bản https của URL này)
}
# request VÔN ĐÃ là https. nên "bản https"
# chính là URL đó. nên nó chuyển hướng về chính mình. mãi mãi.
```

Quy tắc này gần như chắc chắn được viết để giải quyết một vấn đề thực: buộc khách truy cập HTTP chuyển sang HTTPS khi server gốc nằm sau một proxy xử lý TLS. Trong cấu hình đó, server gốc thực sự không thể tự biết khách đã dùng HTTPS hay chưa, nên đọc `X-Forwarded-Proto` là cách làm đúng. Lỗi không nằm ở việc đọc header. Lỗi nằm ở việc chuyển hướng mà không kiểm tra trước xem request đã đến bằng HTTPS chưa.

QUY TRÁCH NHIỆM MỘT CÁCH CHÍNH XÁC

Rất dễ quy lỗi cho một bên. Nhưng cả hai cách quy lỗi đều không đứng vững trước bằng chứng.

Yếu tố kích hoạt đến từ hạ tầng bên truy cập: các nền tảng proxy và edge tự động thêm `X-Forwarded-Proto`, và ứng dụng chạy trên nền tảng đó không thể loại bỏ nó. **Vòng lặp đến từ cấu hình server gốc:** quy tắc bắt buộc HTTPS chuyển hướng vô điều kiện. Bỏ đi một nửa nào thì lỗi cũng biến mất. Nhưng chỉ nửa ở phía server gốc mới có thể được sửa bởi người có lợi ích trong việc website hiển thị.

Cách sửa

Việc sửa nhỏ và ít rủi ro. Quy tắc bắt buộc HTTPS phải kiểm tra giao thức hiện tại trước khi chuyển hướng, và không được chuyển hướng khi request đã đến bằng HTTPS:

```
# logic đúng
nếu (X-Forwarded-Proto tồn tại VÀ giá trị khác "https") {
    chuyển_hướng_vĩnh_vĩnh(đến: bản https)
}
# đã là https? trả trang về: không chuyển hướng.
```

Người quản trị server có thể kiểm chứng bản sửa bằng đúng hai câu lệnh đã dùng để tìm ra lỗi. Sau khi sửa, cả hai phải trả về 200:

```
curl -I https://example.com/robots.txt
curl -I -H 'X-Forwarded-Proto: https' https://example.com/robots.txt
```

Phần 5: Bài học rút ra

Dành cho chủ sở hữu website

- **Hãy kiểm tra đúng đường mà khách truy cập thực sự đi, không phải đường thuận tiện để kiểm tra.** Trình duyệt là kết nối trực tiếp. Phần lớn lưu lượng máy thì không.
- **Kết quả xanh từ một bài test trực tiếp nói ít hơn bạn tưởng.** Nó xác nhận một đường dẫn hoạt động, và không nói gì về hành vi có điều kiện trên các đường khác.
- **Hãy rà soát riêng các quy tắc bắt buộc HTTPS.** Chúng được viết một lần, thường từ nhiều năm trước, thường bởi người không còn tham gia dự án, và chúng đứng trước mọi request mãi mãi.
- **Lỗi im lặng tốn kém hơn lỗi ồn ào.** Một lỗi 500 sẽ được sửa ngay trong ngày xuất hiện. Một vòng lặp chuyển hướng có điều kiện có thể triệt tiêu khả năng hiển thị trên AI vô thời hạn trong khi mọi dashboard vẫn xanh.

Dành cho người làm chuyên môn: phương pháp chẩn đoán

KỸ THUẬT CÓ THỂ ÁP DỤNG LẠI

1. **Coi kết quả trái ngược là dữ liệu, không phải lỗi.** Khi hai bài test đều đáng tin cậy cho kết quả khác nhau, chính sự khác nhau đó khoanh vùng được lỗi.
2. **Chỉ thay đổi một điều kiện mỗi lần.** Cùng URL, cùng thời điểm, cùng user agent, thay đổi một header. Ít hơn thế thì không tách biệt được nguyên nhân.
3. **Luôn có đối chứng gần giống.** Việc test `X-Forwarded-Scheme` song song với `X-Forwarded-Proto` chính là điều loại bỏ giả thuyết "server không thích header proxy" và chứng minh rằng một tên biến cụ thể đang được đọc.
4. **Luôn có đối chứng đồng hạng.** Việc gửi cùng header tới `accor.com`, `marriott.com` và `wikipedia.org` đã xác lập rằng xử lý đúng là bình thường, còn hành vi này thì không.
5. **Phân biệt "có phản hồi" và "hoàn tất được".** Một server trả lời ngay bằng chuyển hướng không phải là server hoạt động tốt. Hãy luôn đi theo chuỗi chuyển hướng và đếm số lần.
6. **Kiểm chứng kết luận của chính mình trước khi công bố.** Cuộc điều tra này đã tạo ra hai chẩn đoán tự tin nhưng sai trước khi tìm ra câu trả lời đúng. Cả hai đã được rút lại bằng văn bản. Uy tín trong chẩn đoán đến từ sự sẵn sàng được phép đo điều chỉnh.

Dành cho công việc tối ưu hiển thị trên AI nói chung

Bài học rộng hơn là về nơi mà khả năng hiển thị trên AI thực sự được quyết định. Rất nhiều lời khuyên GEO được công bố tập trung vào cấu trúc nội dung: schema markup, định dạng câu trả lời, sự rõ ràng của thực thể, cách diễn đạt dễ được trích dẫn. Tất cả đều quan trọng, và tất cả đều vô nghĩa nếu crawler nhận được một vòng lặp 301 thay vì tài liệu.

Khả năng tiếp cận về mặt kỹ thuật không phải một điều kiện tiên quyết được kiểm tra một lần rồi quên. Nó là một bề mặt sống có thể hỏng một cách im lặng, có điều kiện, và không để lại dấu vết trong bất kỳ báo cáo tiêu chuẩn nào. Kiểm chứng nó là việc đầu tiên, không phải thủ tục hình thức trước khi làm việc chính.

TỰ KIỂM CHỨNG

Mọi phép đo trong tài liệu này được thực hiện bằng `curl`, có sẵn trên macOS, Linux và Windows. Thay bằng bất kỳ tên miền bạn muốn kiểm tra:

```
curl -I https://WEBSITE.com/robots.txt
curl -I -H 'X-Forwarded-Proto: https' https://WEBSITE.com/robots.txt
# cả hai phải trả về 200. nếu câu thứ hai trả về 3xx
# trở về cùng URL, website của bạn có lỗi này.
```

Tất cả phép đo trong case study này được thực hiện ngày 12 tháng 8 năm 2026 trên website production đang hoạt động và đã được lặp lại để xác nhận tính ổn định. Bất kỳ bên thứ ba nào cũng có thể tái lập kết quả bằng các câu lệnh đã trình bày. Website không bị can thiệp, không bị dò quét lỗ hổng bảo mật, và không bị truy cập ngoài các request HTTP công khai thông thường. Chúng tôi nêu tên khách sạn vì bằng chứng chỉ có giá trị khi có thể kiểm chứng được, mọi câu lệnh trong tài liệu này bất kỳ ai cũng chạy lại được trên website công khai, và đây là lỗi cấu hình chứ không phải lỗ hổng bảo mật. Phân tích thực hiện bằng AI Crawler Check, công cụ theo dõi chính sách truy cập crawler cho 248 bot trên 8 nhóm, trong đó 177 bot được ghi nhận trong tài liệu chính thức của nhà vận hành.