Security Headers on Directory-Listed U.S. Local-Business Websites

By RACKCRUNCH Team · v7.4.2 · Scans 2026-09-24

Artifacts: report, dataset and code at https://rackcrunch.com/security-headers-2026/

Why we looked

RACKCRUNCH has a free Security Header Check. You give it a URL, it makes one request, and it tells you which response headers are set, which are set badly, and which are missing.

Checking one site at a time made us curious about the bigger picture. We didn't mean banks or big tech. We meant the plumber, the law office, the car lot, the pizza place: businesses that may lack dedicated web-security staff.

We needed a list of those businesses, and there isn't a clean one. So we used the closest thing we could find: the public Curlie web directory, the human-edited successor to DMOZ. We drew 7,040 directory rows at random from its US local "Business and Economy" categories, as of the 2026-02-02 snapshot. They represent 7,022 unique initial registrable domains. Think of it as an SMB-oriented directory sample. We did not check how big any of these businesses are. Some will be larger than "small", and the directory tilts toward businesses established enough to get listed. When this report says "sites", it means these directory-listed sites, not every small business in the US.

On 2026-09-24 we ran two HTTPS request-chain scans of each sampled URL, following at most three redirects; all reported estimates come from the second scan. We read the response headers and never the pages themselves.

We expected low adoption. That is roughly what we got. The surprises came from how low, and from one number we had to take apart.

Which headers sites send

5,642 of the 7,040 sampled rows gave us a usable HTTPS response. 4,701 of those were HTTP-200 responses. They resolve to 4,688 unique final registrable domains (the registrable domain of the final URL after redirects). Those 4,688 are our principal base, which we also call `dedup-200`: one row per final registrable domain, keeping the last row in scan-2 check-completion order (see Bases and terms). Every headline figure uses this base, and every figure in this report is on it unless it says otherwise.[1]

We started with the simplest question: which of seven study-defined explicit-header criteria does each site meet? These are criteria under our rubric, not a security grade, and a site can reasonably leave some of them out. Three need a word up front. A missing Referrer-Policy gets the browser's secure default, so we don't treat it as a failure; the point goes only to sites that set a secure value themselves. Permissions-Policy is still experimental and unevenly supported

across browsers, and our criterion counts an explicit declaration, not protection gained. Cross-Origin-Opener-Policy depends on context: it matters most for sites that open pop-ups or handle cross-origin windows, and it can break sign-in and payment flows.

No explicit criterion in our rubric was near-universal.

HSTS was the most commonly observed header, present on 43.8% of sites. The most commonly passed criterion in our seven-item rubric was X-Content-Type-Options: nosniff, at 39.7%; only 12.3% met our stronger HSTS criterion of at least one year plus includeSubDomains. We call that study-defined strong HSTS, and it is measured on the final host only (the full hostname of the final URL). If a site redirects to www, includeSubDomains there covers hosts under www, not the bare domain or sibling hosts such as shop. Of the 578 strong-HSTS passes, 260 (45.0%) came from a www final host, 316 from the bare domain and 2 from another subdomain. Thus, for nearly half of strong-HSTS passes, the observed policy was on a www host and does not establish coverage of the bare domain. 31.2% send a clickjacking header with a recognized restrictive value under our parser. 8.0% send an explicit non-wildcard Permissions-Policy declaration, and 1.1% send a recognized non-default Cross-Origin-Opener-Policy value.

86.6% of sites don't send a Referrer-Policy at all, and as above, that is less bad than it sounds. When the header is missing, modern browsers fall back to strict-origin-when-cross-origin, which is a sensible default. The more concerning cases are explicit weak legacy values.

And one site in five gives something away. **20.5% [19.3-21.7]** of sites matched our version-token rule: a digit in any of Server, X-Powered-By, X-AspNet-Version or X-AspNetMvc-Version, such as Apache 2.4.x, Microsoft-IIS/10.0 or nginx 1.x. For 27 sites, the only leak was an ASP.NET version header. That can aid version-targeted reconnaissance, but it is a low-severity finding: hiding a version number patches nothing.

Here is the full table on the principal base. Intervals quantify sampling uncertainty under the stated sample model, not frame bias, parser misclassification, CDN variation, or selective platform attribution.

Control (principal base, n=4,688)	Sites	Share	95% CI
HSTS present	2,053	43.8%	42.4-45.2
Strong HSTS: at least 1 year + includeSubDomains (final host)	578	12.3%	11.4-13.3
HSTS shorter than 1 year	665	14.2%	13.2-15.2
CSP enforced (any policy)	992	21.2%	20.0-22.4
CSP report-only	37	0.8%	0.6-1.1
CSP restricts scripts	186	4.0%	3.4-4.6
Passed our header-only script-CSP rule	8	0.17%	0.1-0.3
Clickjacking protection (recognized restrictive value)	1,463	31.2%	29.9-32.5
Clickjacking header with ineffective value	15	0.3%	0.2-0.5
X-Content-Type-Options: nosniff	1,861	39.7%	38.3-41.1
Referrer-Policy secure value set	373	8.0%	7.2-8.8

Control (principal base, n=4,688)	Sites	Share	95% CI
Referrer-Policy weak (legacy values)	252	5.4%	4.8-6.1
Referrer-Policy invalid (unrecognized value)	1	0.02%	0.0-0.1
Referrer-Policy missing (browser default applies)	4,062	86.6%	-
Explicit non-wildcard Permissions-Policy declaration	374	8.0%	7.2-8.8
Permissions-Policy wildcard-only	10	0.2%	0.1-0.4
COOP same-origin	35	0.7%	0.5-1.0
COOP same-origin-allow-popups	15	0.3%	0.2-0.5
COOP noopener-allow-popups or restrict-properties	0	0.0%	0.0-0.1
Recognized non-default COOP value (any of the above)	50	1.1%	0.8-1.4
COOP no-op (unsafe-none)	19	0.4%	0.3-0.6
COOP malformed	4	0.09%	0.0-0.2
Version-token disclosure	960	20.5%	19.3-21.7

The one invalid Referrer-Policy is a site that sends "Referrer-Policy: yes".

Put the seven criteria together and the headline number follows. **On the principal base, 49.7% met none of seven study-defined explicit-header criteria** [95% CI 48.3-51.2], or 2,331 of 4,688.[2] This is an adoption measure, not an estimate of the proportion of insecure websites.

The same measures on every usable response, including error and bot-challenge pages, are in the sensitivity analyses at the end. Such responses may reflect CDN, WAF, hosting-platform, or application error-page configurations rather than the headers on the site's ordinary homepage, which is why they are not the principal base.

The CSP number that fell apart

In our first draft, the CSP result looked almost respectable. Roughly one site in six "passed". That seemed high. So in a revised analysis we stopped counting CSPs and started reading them, with a parser implementing the documented subset of CSP Level 3 described in the rubric.

About one site in five, 21.2% [20.0-22.4], sends an enforced CSP. That sounds fine until you look at what the policies say. Most are framing rules (frame-ancestors, which is anti-clickjacking, not anti-script) or upgrade-insecure-requests. The three most common exact policies were `frame-ancestors 'self'` (170 sites), the Shopify platform default `block-all-mixed-content; frame-ancestors 'none'; upgrade-insecure-requests;` (167), and bare `upgrade-insecure-requests` (130); together they account for 47.1% of the 992 sites sending an enforced Content-Security-Policy. None of the three has a script directive.[3]

CSP is an important defense-in-depth control against injected scripts. Here is how the policies break down by what they actually do for scripts.

CSP class (principal base, n=4,688)	Sites	Share
No enforced CSP	3,696	78.8%
CSP present, no script-restricting directive	804	17.2%
Script-restricting: allowlist-based	173	3.7%
Script-restricting: nonce/hash strict	5	0.1%
Script-restricting: strict-dynamic with nonce/hash anchor	8	0.2%
Invalid or unparseable	2	0.04%
Script-restricting total	**186**	**4.0%**

What undoes the rest? A policy can fall into more than one row:

Characteristics of script-restricting CSPs - count and share of all 4,688 sites	Sites	Share
'unsafe-inline' effective for script elements	172	3.7%
'unsafe-inline' effective for inline event handlers (script-src-attr chain)	170	3.6%
'unsafe-inline' present but ignored (neutralized by a nonce or hash)	4	0.1%
'unsafe-eval' effective	157	3.3%
Full wildcard (bare * or scheme source) effective	81	1.7%
Subdomain wildcard (*.host) in effective script directive	55	1.2%
Passed our header-only script-CSP rule	**8**	**0.17%**

That leaves **eight sites in the principal base, 0.17% [0.1-0.3], that passed our header-only script-CSP rule.** This does not establish that the policy is unbypassable or that nonces are fresh and correctly applied in page markup. We did not test nonce freshness, whether nonces match the scripts in the page, or whether allowlisted script hosts can be abused, for example through JSONP endpoints. We did read all eight policies in full by hand to check the parser's reading of the header. None of the eight sends script-src-attr, so inline event handlers fall back to the same restricted script-src. Two are policies issued by a hosted site builder rather than written by the site owner. One allows no scripts at all. Two still allow subdomain-wildcard hosts as script sources, which our rubric reports separately and does not count against them.

Eight. Most observed CSPs consisted primarily of framing or mixed-content directives.

We got one thing wrong

A draft of this study had a nice moment in it: six sites out of 4,673 met all eight rubric criteria. We liked that line. It was wrong.

Our first rubric gave CSP frame-ancestors credit twice, once as clickjacking protection and again as a passing CSP. The second analysis pass fixed that. It also made the CSP check

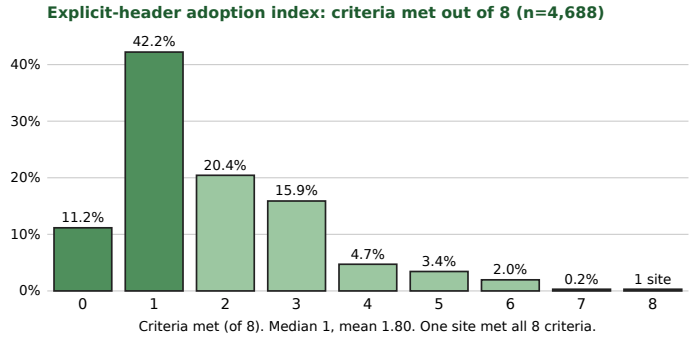
require a script-restricting policy that passes our header-only rule, and it stopped counting no-op values like COOP unsafe-none or a wildcard-only Permissions-Policy. Each of the six lost the double-counted point.

The first draft said six sites met all eight rubric criteria; the corrected rubric says one, and eight more met seven. The one got there on a CSP that uses strict-dynamic with a nonce, which our earlier parser misread.

A few of those nine sites share header patterns consistent with a reused configuration: one developer's or vendor's checklist applied to several sites. We are not naming any of them. Nine sites is too few to support a story, and headers change. Meeting all eight criteria also says nothing about the rest of a site: it can still have serious application flaws.

The explicit-header adoption index

Adding up the criteria gives a single number per site, which we call the explicit-header adoption index: the seven criteria above plus an eighth, version hygiene (no version in the software banner). It mixes unlike things, from broadly useful hardening to context-dependent controls and a low-severity banner check, all weighted equally. So treat it as a summary of adoption. The per-control table above is the main result.



Criteria met (of 8)	Sites	Share	95% CI
0	523	11.2%	10.3-12.1
1	1,979	42.2%	40.8-43.6
2	958	20.4%	19.3-21.6
3	745	15.9%	14.9-17.0
4	221	4.7%	4.1-5.4
5	161	3.4%	2.9-4.0
6	92	2.0%	1.6-2.4
7	8	0.2%	0.1-0.3
8	1	0.02%	0.0-0.1

The median is 1 and the mean is 1.80. The 0 row is smaller than the "none of seven" headline because many sites meet the eighth criterion, version hygiene, simply by not announcing their software. We haven't validated the index as a measure of how secure a site is.

Header-identified server and hosting labels

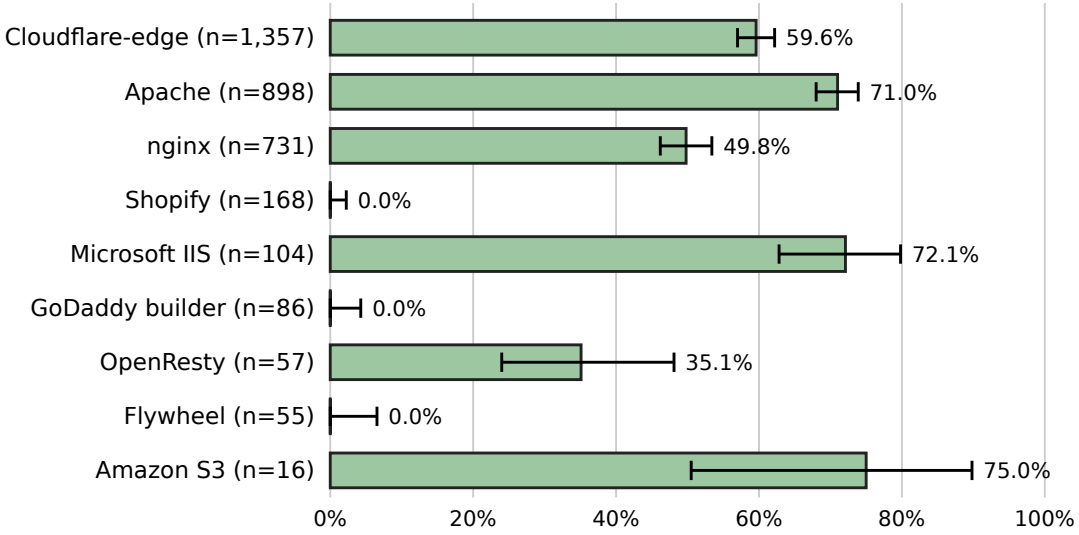
On the second scan we kept the full Server header, not just the version-leaking ones. Between Server tokens and platform-specific CSPs, we could attach a server or hosting label to 3,481 of the 4,701 HTTP-200 responses (74.0%). Our earlier labeling managed 15.0%. The biggest single group, 1,357 sites (28.9%), is one we call Cloudflare-edge: the Server header reads exactly "cloudflare". That tells you the response came through Cloudflare's network. It says nothing about the server behind it.

These numbers are descriptive. Labels only exist for servers that identify themselves, so the labeled group is selective, and nothing here tells you what would happen if a given business moved hosts. Some labels come from platform-specific CSPs, which are among the headers being measured, so the comparison is partly built from its own outcomes. Each response gets one label under fixed rules, and the first rule that matches wins, so no site carries two labels. Builder and platform signatures come first, because they name whoever controls the configuration. Server tokens come after. Matching is case-sensitive, as the header was sent. The order is GoDaddy builder, Flywheel, Pagely, Shopify, Wild Apricot, Cloudflare-edge, Microsoft IIS, Apache, nginx, OpenResty and Amazon S3. The exact rules are listed after the rubric below and in the data dictionary.

We made many comparisons here and in the sector section and applied no multiplicity correction. Differences between labels or categories may reflect platform composition, category assignment, response propensity or which servers label themselves, rather than anything about the businesses themselves.

Subgroup tables retain response rows because sector and state belong to the sampled listing; de-duplicating final registrable domains would require deciding which sampled listing's attributes to keep.

None of seven criteria met, by header-identified label (95% CI)



Exploratory. HTTP-200 responses with a label identified in headers (3,481 of 4,701, 74.0%). Final rubric.

Label (exploratory)	n	Mean adoption index (of 8)	None of 7	Strong HSTS	Passed script-CSP rule	Clickjacking (recognized value)	nosniff
Cloudflare-edge	1,357	1.72	59.6% [57.0-62.2]	8.5% [7.2-10.2]	0.2% [0.1-0.6]	18.5% [16.5-20.7]	32.6% [30.1-35.1]
Apache	898	1.35	71.0% [68.0-73.9]	10.1% [8.3-12.3]	0.1% [0.0-0.6]	19.6% [17.1-22.3]†	20.5% [18.0-23.3]
nginx	731	1.80	49.8% [46.2-53.4]†	8.2% [6.4-10.4]	0.3% [0.1-1.0]	38.9% [35.4-42.4]†	36.5% [33.1-40.1]
Shopify	168	3.01	0.0% [0.0-2.2]	0.0% [0.0-2.2]	0.0% [0.0-2.2]	100.0% [97.8-100.0]	100.0% [97.8-100.0]
Microsoft IIS	104	0.58†	72.1% [62.8-79.8]†	10.6% [6.0-18.0]	1.0% [0.2-5.2]	24.0% [16.8-33.1]†	14.4% [8.9-22.4]
GoDaddy builder	86	2.00	0.0% [0.0-4.3]	100.0% [95.7-100.0]	0.0% [0.0-4.3]	100.0% [95.7-100.0]	0.0% [0.0-4.3]
OpenResty	57	1.77	35.1% [24.0-48.1]	7.0% [2.8-16.7]	0.0% [0.0-6.3]	56.1% [43.3-68.2]	63.2% [50.2-74.5]
Flywheel	55	1.16	0.0% [0.0-6.5]	0.0% [0.0-6.5]	0.0% [0.0-6.5]	3.6% [1.0-12.3]	100.0% [93.5-100.0]
Amazon S3	16	0.88	75.0% [50.5-89.8]	18.8% [6.6-43.0]	0.0% [0.0-19.4]	25.0% [10.2-49.5]	25.0% [10.2-49.5]

HTTP-200 responses (n=4,701 rows). Wild Apricot (6 sites) and Pagely (3) are too small to report. Intervals on the mean adoption index are omitted for readability. † Recomputed under the final rubric (X-Frame-Options duplicate normalization and COOP re-run); these cells moved from the earlier draft. Cloudflare-edge is a new row; it was unlabeled in earlier drafts.

Every Shopify-labeled site had a recognized clickjacking header and nosniff, and none met study-defined strong HSTS. Every GoDaddy-builder site met strong HSTS and had a recognized clickjacking header, and none had nosniff. Sites labeled Apache, nginx or IIS average between 0.58 and 1.80 on the adoption index, and between 49.8% and 72.1% of them meet none of the seven criteria. Cloudflare-edge sites average 1.72, and 59.6% meet none of the seven.

Policies that passed our header-only script-CSP rule are rare under every label. Of the eight, 3 are Cloudflare-edge, 2 nginx, 1 Apache and 1 IIS, and 1 has no label. Shopify, GoDaddy builder, Wix and Squarespace sites have none.

By kind of business

Directory categories give us a rough way to split the sample by type of business. Here are the nine categories with at least 80 HTTP-200 responses.

Sector (exploratory)	n	Mean adoption index (of 8)	None of 7	Strong HSTS	CSP enforced	Clickjacking (recognized value)	nosniff
Shopping	372	2.11†	35.5% [30.8-40.5]†	12.1% [9.2-15.8]	33.3% [28.7-38.3]	48.1% [43.1-53.2]†	56.2% [51.1-61.1]
Restaurants & Bars	408	2.07†	46.1% [41.3-50.9]	11.3% [8.6-14.7]	23.3% [19.4-27.6]	31.1% [26.8-35.8]†	45.6% [40.8-50.4]
General	1,620	1.82†	48.9% [46.5-51.3]†	12.8% [11.3-14.6]	21.5% [19.6-23.6]	32.5% [30.3-34.9]†	39.9% [37.6-42.3]
Computers & Internet	226	1.81†	58.0% [51.4-64.2]	15.5% [11.4-20.8]	20.8% [16.0-26.6]	30.5% [24.9-36.8]†	33.6% [27.8-40.0]
Automotive	175	1.73†	54.3% [46.9-61.5]	11.4% [7.5-17.0]	21.7% [16.2-28.4]	36.0% [29.3-43.3]†	32.0% [25.5-39.2]
Industrial	169	1.72†	50.3% [42.8-57.7]†	12.4% [8.3-18.2]	20.1% [14.8-26.8]	32.5% [25.9-39.9]†	37.3% [30.3-44.8]
Legal Services	194	1.69†	59.3% [52.2-65.9]	13.4% [9.3-18.9]	17.5% [12.8-23.5]	27.3% [21.5-34.0]†	32.0% [25.8-38.8]
Construction & Maintenance	368	1.67†	51.6% [46.5-56.7]†	10.3% [7.6-13.9]	14.9% [11.7-18.9]	25.3% [21.1-30.0]†	39.1% [34.3-44.2]
Real Estate	541	1.49†	56.2% [52.0-60.3]†	11.5% [9.0-14.4]	18.7% [15.6-22.2]	22.6% [19.2-26.3]†	32.2% [28.4-36.2]

HTTP-200 responses (n=4,701 rows). Categories under 80 responses are left out. Intervals on the mean adoption index are omitted for readability. † Recomputed under the final rubric; these cells moved from the earlier draft.

The same multiplicity caveat applies here. Shopping has the lowest observed none-of-seven share: 35.5% meet none of the seven criteria, against 56.2% for real estate. That is a descriptive, exploratory gap. Shopping also leads on recognized clickjacking headers (48.1%), nosniff (56.2%) and enforced CSP (33.3%). That fits the label picture above, since hosted shop builders tend to ship those headers. But this table can't tell you why.

Real estate has the lowest observed mean index (1.49 out of 8). It also has the lowest clickjacking rate (22.6%).

The CSP column needs the same warning as before. Enforced CSPs show up everywhere, but only eight policies passed our header-only script-CSP rule across the full principal base; sector-specific counts were too small to interpret.

A recognized non-default COOP value is too rare to compare by sector. The one exception is restaurants and bars, at 4.4% [2.8-6.9], and even there it looks like it comes from platform defaults.

The cobbler's children

Then there is the row we had been half-dreading.

Computers & Internet is the Curlie category whose listings are about technology. 58.0% [51.4-64.2] of sites listed there meet none of the seven criteria. That is the second-highest share of any category, behind legal services at 59.3%. Their strong-HSTS rate, 15.5%, is the highest in the table, but only by a little, and the intervals overlap.

Sites listed in Curlie's Computers & Internet category did not show higher header-criterion attainment. Category membership does not establish the business's services or staff expertise. The interval is also wide, and it overlaps most of the table.

By state

We cut the results by state. Nine states have at least 150 HTTP-200 responses, from 170 to 570 each, and the share meeting none of the seven criteria ranges from 42.9% [36.0-50.0] in New York to 56.2% [49.2-63.0] in Florida. State estimates varied descriptively. We did not prespecify or power the study for state comparisons, and we did not conduct multiplicity-adjusted state tests, so we do not rank states or infer state-level differences. A Pearson chi-square test of the 9-by-2 table (state by none-of-seven) found no detectable heterogeneity (chi-square 8.77, 8 degrees of freedom, p = 0.36), and the unadjusted New York-Florida difference does not survive a correction for the 36 possible pairs. The state-level counts behind this test are in the released analysis output.

The floor under everything

There is one more number, and it sits under all the others.

1,398 of the 7,040 sampled rows, 19.9% [18.9-20.8], gave us no usable HTTPS response at all. 737 failed at the TLS or connection stage. 215 redirected to plain HTTP. 204 timed out, and 192 had DNS that doesn't resolve. The rest were redirect loops, DNS errors and a handful of odd cases.

That is its own problem, about availability and TLS rather than headers. We can't say anything about the header posture of those sites, because we never got that far. What we can say is that for roughly one sampled row in five, our scanner did not obtain a usable HTTPS response under our scan rules.

What a site owner can do, in order

The fixes are mostly short. They are not all risk-free. Roll them out in stages and test as you go.

1. **Deploy HSTS in stages.** Begin with a short max-age, verify all affected services, then increase toward a long duration. Add includeSubDomains only after confirming that every current and anticipated subdomain supports HTTPS, because browsers will refuse plain-HTTP connections to any subdomain that doesn't. Consider preload only after meeting its requirements and accepting its operational consequences. Set the header on the bare domain as well as www.
2. **Add clickjacking protection**: X-Frame-Options: SAMEORIGIN, or CSP frame-ancestors 'self'. Check first that nothing legitimate embeds your pages (booking widgets, partner sites, your own apps). This will break them.
3. **Add X-Content-Type-Options: nosniff.** One line, and generally low-risk, but test MIME-dependent downloads and legacy behavior.
4. **Check your Referrer-Policy.** If it is missing, the browser default is reasonable. If it is set to a legacy value like no-referrer-when-downgrade, change it to strict-origin-when-cross-origin.
5. **Patch your server software.** Removing version numbers from Server, X-Powered-By and the ASP.NET version headers is a low-severity cleanup: it hides the problem and doesn't fix it. Keeping the software current matters more.
6. **Start a Content-Security-Policy in report-only mode**, then move toward a nonce- or hash-based strict CSP after testing. Strict-CSP guidance also recommends object-src 'none' and base-uri 'none', which our rubric does not require, so passing our rule is not the same as a production-quality strict CSP. The eight header sets that passed our rule do not establish implementation quality or nonce freshness.
7. **Be careful with Cross-Origin-Opener-Policy.** same-origin can break pop-up sign-in and payment flows. Test those before turning it on.

On hosted platforms, part of this is already done for you, and part isn't. Check which headers your platform leaves out.

How we measured

Population. Domains listed in US locality "Business and Economy" categories in the Curlie snapshot (the files in the curlie-rdf-all.tar.gz archive dated 2026-02-02, downloaded 2026-09-24 from https://curlie.org/download): 110,444 unique domains. Categories covering government, healthcare, education, financial services and military were excluded, as were listing directories and social platforms. Business size was not verified. Curlie is human-edited and runs dead-link cleanup, so the pool skews toward established businesses with maintained listings.

Sample. 7,040 rows drawn at random with a fixed seed. These cover 7,022 unique initial registrable domains; 18 rows shared a registrable domain with another row.

Bases and terms. These definitions are the same across the report, dataset and data dictionary.

- *Sampled rows* (n=7,040): directory rows drawn at random, covering 7,022 unique initial registrable domains.
- *Registrable domain*: the domain name with its last two labels, or three when the second-level part is a common public suffix such as co.uk. This is a heuristic, not a full Public Suffix List lookup.
- *Final registrable domain*: the registrable domain of the final URL after redirects. *Final host* means the full hostname of the final URL and is used only for HSTS host scope.
- *Usable responses* (n=5,642 rows, 5,614 unique final registrable domains): rows with a usable HTTPS response.
- *HTTP-200 responses* (n=4,701 rows): usable rows whose final status was 200.
- *Principal base*, also called `dedup-200` (n=4,688): unique final registrable domains among the HTTP-200 responses, with one row kept per final registrable domain: the last row in scan-2 check-completion order (the released file's row order; sorting by row_id, i.e. scan order, changes the dedup). Every headline figure uses this base. Code and column names such as `dedup200` refer to this base.

Scan. We ran two HTTPS request-chain scans of each sampled URL, following at most three redirects; all reported estimates come from the second scan. Method: HTTPS GET. The response body was never read; the connection was closed after the response headers. TLS verification was on, and only public IP addresses were contacted. Timeouts were 3 s for DNS, 3 s per hop and 8 s total, with 24 concurrent workers. The scanner identified itself as RACKCRUNCH-header-check/1.0 with a contact URL for opt-out. Scans ran from Google Cloud (Oregon, US) at 12:26-12:31 EDT (first) and 13:16-13:21 EDT (second) on 2026-09-24. For scan-two requests that produced a final HTTP response, the scanner retained the complete final-response header block internally (see Measurement ethics and data release).

Drift between scans. All 7,040 sampled URLs were scanned both times. Status transitions from the first scan to the second:

First scan -> second scan	Sampled rows
200 -> 200	4,608
200 -> not 200	65
not 200 -> 200	93
unusable -> unusable	1,348
403 -> 403	574
202 -> 202	229
404 -> 404	59
All other transitions (each under 20)	64

The first scan recorded only per-check values truncated at 400 characters, so a full comparison under the final rubric is not possible. Of the 7,040 sampled rows, 4,608 were HTTP-200

responses in both scans. Under the original rubric, 13 of those rows (0.28%) changed at least one check, and the largest change in any headline prevalence between scans was 0.25 points (X-Content-Type-Options, all-usable base). Those are old-rubric diagnostics, not evidence that the corrected classifications are stable. As a bounded check, we applied the final rubric to the 4,484 of those rows (97.3%) whose scan-1 check values were complete. 7 rows changed at least one index item, 0.16% [0.08-0.32]: clickjacking 2, Permissions-Policy 3, X-Content-Type-Options 3, version hygiene 2 (some rows changed more than one). The 124 excluded rows are the ones with the longest CSPs, so this supports short-term stability but doesn't prove it.

Disposition. 5,642 usable HTTPS responses: status 200: 4,701; 403: 577; 202: 245; 404: 59; 500: 11; 307: 9; 429: 8; 401: 5; 503: 4; 405: 4; 520: 3; 526: 3; 400: 3; 406: 2; 521: 2; 502, 525, 410, 423, 523 and 530: 1 each. 1,398 unusable (each failure category is defined in the data dictionary): TLS or connect failure 737, redirect to non-HTTPS 215, timeout 204, DNS does not resolve 192, too many redirects 29, DNS lookup failed 13, non-public address 4, invalid redirect 4. Responses with HTTP status 202 (245 responses), often generated by bot-protection layers, were classified as reachable but excluded from the HTTP-200 analysis.

Duplicates and parked pages. The 5,642 usable responses resolved to 5,614 unique final registrable domains, and the 4,701 HTTP-200 responses to 4,688, the principal base. Among usable responses, only 15 final destinations served more than one sampled row (43 responses in total, at most 9 for any one destination). Deduplication moves estimates by 0.3 points or less. 9 HTTP-200 responses (0.19%) landed on parked, expired-domain or platform pages. These were identified from the hostname of the final URL (for example expireddomains.com or forsale.godaddy.com), not from page content. They stay in the base because their headers are what the scanner received.

Intervals. Wilson 95% confidence intervals throughout (Wilson, 1927). The random draw was of directory rows, and the principal base is unique final registrable domains after redirects, so the intervals on that base are approximate. They quantify sampling uncertainty under the stated sample model, not frame bias, parser misclassification, CDN variation, or selective platform attribution. Label, sector and state cuts are exploratory.

Rubric

The eight criteria of the explicit-header adoption index, as used in this report. The full rubric code and data dictionary are published at https://rackcrunch.com/security-headers-2026/.

Item	Passes when
HSTS (study-defined strong)	max-age of at least one year and includeSubDomains, on the final host. This does not show that the bare domain or sibling hosts are covered

Item	Passes when
Content-Security-Policy	passes our header-only script-CSP rule. We evaluate three script channels: script elements (script-src-elem, then script-src, then default-src), inline event handlers (script-src-attr, then script-src, then default-src) and eval (script-src, then default-src). A site passes when, across all enforced policies evaluated together as browsers do, no channel allows 'unsafe-inline', 'unsafe-eval', a bare * or a bare scheme source. 'unsafe-inline' next to a nonce or hash is ignored, as are host sources, 'self' and 'unsafe-inline' next to strict-dynamic with a nonce or hash. Duplicate directives: the first wins. Subdomain wildcards are reported but not failed. The rule reads headers only: it does not test nonce freshness, markup matching or allowlist bypasses, and it does not require object-src or base-uri
Clickjacking	an enforced CSP frame-ancestors directive, if present, decides the result; X-Frame-Options is used only when no enforced frame-ancestors is present. Our scanner joins repeated header fields with a comma, so a site that sent SAMEORIGIN twice was recorded as 'SAMEORIGIN, SAMEORIGIN'; we treat identical repeats as the single value. Conflicting repeats and the obsolete ALLOW-FROM count as ineffective
X-Content-Type-Options	nosniff
Referrer-Policy	a secure value is set (missing = browser default, not a failure, but no point; legacy values = weak; unrecognized values = invalid)
Permissions-Policy	an explicit non-wildcard declaration: at least one directive whose allowlist is not a bare *, such as camera=() or geolocation=(self). A policy whose directives are all * does not count. The rule does not compare a declared allowlist with the feature's browser default, so a directive may simply restate the default
Cross-Origin-Opener-Policy	a recognized non-default COOP value under our rubric: same-origin, same-origin-allow-popups, noopener-allow-popups or restrict-properties (the last is a proposal that was discontinued and never shipped in any browser; neither of the last two appeared in the data). Parameters after ';' are ignored. unsafe-none does not count. Unknown token-like values are reported separately from malformed ones; the space variant 'same-origin allow-popups', which browsers ignore, is malformed
Version hygiene (low severity)	no digit in any of Server, X-Powered-By, X-AspNet-Version or X-AspNetMvc-Version. Any digit counts, including dates or digits in a product name; a banner with no digits passes

The "none of seven" headline uses the first seven items and leaves out version hygiene.

Hosting labels. Rules are checked in this order and the first match wins. Matching is case-sensitive.

1. GoDaddy builder: CSP contains "godaddy.com", or Server starts with "DPS"
2. Flywheel: Server starts with "Flywheel"

3. Pagely: Server starts with "Pagely"
4. Shopify: CSP starts exactly with "block-all-mixed-content; frame-ancestors 'none'; upgrade-insecure-requests"
5. Wild Apricot: CSP contains "wildapricot"
6. Cloudflare-edge: Server is exactly "cloudflare"
7. Microsoft IIS: Server starts with "Microsoft-IIS"
8. Apache: Server starts with "Apache"
9. nginx: Server starts with "nginx"
10. OpenResty: Server starts with "openresty"
11. Amazon S3: Server starts with "AmazonS3"

Anything else gets no label.

Caveats

- One page, one moment. Headers can differ by page, CDN edge or login state. We read the home page only.
- Sites behind a CDN or hosted platform often show the platform's headers, not the owner's configuration. That is real protection for the site, and it also means some passes are inherited.
- We did not test the http:// to https:// redirect. Our checker does, but these scans followed one request chain per site.
- The sample comes from a human-edited directory. It skews toward businesses established enough to be listed, and business size was not verified.
- Labels cover the 74.0% of HTTP-200 responses that identify a server or hosting platform in their headers. The comparisons describe those sites. They don't estimate what a platform causes.
- The clickjacking figure includes 116 sites credited because we treat identical repeated X-Frame-Options values, joined by our scanner, as one value (see Rubric). Under the current HTML processing model, identical repeated values such as SAMEORIGIN, SAMEORIGIN retain same-origin framing protection; older browser implementations have differed. Counting those as ineffective would give 1,347 (28.7%).
- This is a light scan, not a security audit of any individual business.

Sensitivity analyses

All usable responses. Every usable HTTPS response, including error and bot-challenge pages (n=5,642). On this base, 48.4% [47.1-49.7] meet none of the seven criteria (2,733 of 5,642).

Control (all usable responses, n=5,642)	Sites	Share	95% CI
HSTS present	2,233	39.6%	38.3-40.9
HSTS at least 1 year	1,449	25.7%	24.6-26.8

Control (all usable responses, n=5,642)	Sites	Share	95% CI
Strong HSTS: at least 1 year + includeSubDomains (final host)	609	10.8%	10.0-11.6
HSTS shorter than 1 year	784	13.9%	13.0-14.8
CSP enforced	1,341	23.8%	22.7-24.9
CSP report-only	43	0.8%	0.6-1.0
CSP restricts scripts	510	9.0%	8.3-9.8
Passed our header-only script-CSP rule	8	0.14%	0.1-0.3
Clickjacking (recognized restrictive value)	1,955	34.7%	33.4-35.9
X-Content-Type-Options: nosniff	2,263	40.1%	38.8-41.4
Referrer-Policy secure	853	15.1%	14.2-16.1
Referrer-Policy weak (legacy values)	254	4.5%	4.0-5.1
Referrer-Policy invalid (unrecognized value)	1	0.02%	0.0-0.1
Referrer-Policy missing (browser default applies)	4,534	80.4%	-
Explicit non-wildcard Permissions-Policy declaration	701	12.4%	11.6-13.3
Permissions-Policy wildcard-only	10	0.2%	0.1-0.3
COOP same-origin	352	6.2%	5.6-6.9
COOP same-origin-allow-popups	19	0.3%	0.2-0.5
Recognized non-default COOP value (any)	371	6.6%	6.0-7.3
COOP no-op (unsafe-none)	20	0.4%	0.2-0.5
COOP malformed	4	0.07%	0.0-0.2
Version-token disclosure	976	17.3%	16.4-18.4

The COOP and CSP figures here need a caveat. Most of the 352 same-origin COOP values come from CDN challenge pages, and so do the CSPs: 325 of the usable responses (5.8%) carry nonce/hash-strict CSPs, but these are mostly CDN bot-challenge interstitials (Cloudflare challenge pages carry nonce CSPs), not deployments by the sites; 316 of the 318 script-src-attr uses on this base sit on those Cloudflare responses. On the principal base that class is 5 sites. The rest of the mix: no CSP 4,301 (76.2%), no script directive 829 (14.7%), allowlist-based 177 (3.1%), strict-dynamic with anchor 8 (0.1%), invalid 2.

Base comparison. The same measures across bases:

Measure	All usable (5,642)	HTTP-200 (4,701)	Principal base (4,688)
None of seven	48.4%	49.7%	49.7%
HSTS present	39.6%	43.8%	43.8%
Strong HSTS	10.8%	12.4%	12.3%
CSP enforced	23.8%	21.3%	21.2%
CSP restricts scripts	9.0%	4.1%	4.0%
Passed our header-only script-CSP rule	0.14%	0.17%	0.17%
Clickjacking (recognized value)	34.7%	31.3%	31.2%
nosniff	40.1%	39.8%	39.7%
Referrer-Policy secure	15.1%	8.0%	8.0%
Explicit non-wildcard Permissions-Policy declaration	12.4%	8.0%	8.0%
Recognized non-default COOP value	6.6%	1.1%	1.1%
Version-token disclosure	17.3%	20.5%	20.5%

COOP drops from 6.6% to 1.1% on the HTTP-200 responses, which exclude most identified challenge responses, and Referrer-Policy and Permissions-Policy drop by about a third. Deduplication barely moves anything. The none-of-seven share is about half on all three bases.

Measurement ethics and data release

We scanned only public home pages over HTTPS, one request chain per sampled URL per scan, following at most three redirects. We never read page bodies, submitted forms, logged in or probed for vulnerabilities. The scanner contacted only public IP addresses, ran at most 24 concurrent connections across the whole sample with short timeouts, and identified itself as RACKCRUNCH-header-check/1.0 with a contact URL for opt-out. We have received no opt-out requests to date.

For every scan-two request that produced a final HTTP response, the scanner retained the complete final-response header block internally. The only limit was a cap of 65,536 bytes on the whole block, applied at capture and before any classification; the largest captured block was 15,362 bytes, so no scan-two final-response header block was truncated by the capture-size cap. Headers from redirect hops were not kept. The first scan kept only per-check values, cut at 400 characters after classification, which is why drift between scans can only be bounded. The released dataset carries class labels, not header values, so readers can recompute every figure in this report from it but cannot re-grade the headers themselves. The scans were unauthenticated GET requests with no cookies sent, so the data holds nothing beyond what any visitor's browser would receive.

The released dataset has exactly one row for each of the 7,040 sampled rows, and nothing more. Rows without a usable response carry only their error class and HTTP status. Each row lists the sector, state, whether the response was usable, the error class, HTTP status, number

of redirects, a class label for each graded header (HSTS, CSP, X-Frame-Options, X-Content-Type-Options, Referrer-Policy, Permissions-Policy, COOP and version-token disclosure) and the per-criterion results. It holds no header values, no scan times and no raw header map, so no cookies, CSP nonces, reporting endpoints or version strings leave our side.

The dataset is de-identified. Every business appears only under a stable pseudonymous ID, and neither this report nor the dataset names any site as having weak or missing headers. We chose this level because the point of the study is the overall picture, and naming small businesses with weak headers would add risk for them without making that picture any clearer. If you think your business might be in the sample, run your own domain through the free RACKCRUNCH Security Header Check (https://rackcrunch.com/security-headers) to see what the scanner saw. We will confirm a specific domain's row only to a verified owner of that domain.

Technical sources

- Content Security Policy Level 3 (W3C): https://www.w3.org/TR/CSP3/
- Strict CSP guidance: https://csp.withgoogle.com/docs/strict-csp.html and https://web.dev/articles/strict-csp
- Weichselbaum et al., "CSP Is Dead, Long Live CSP! On the Insecurity of Whitelists and the Future of Content Security Policy", ACM CCS 2016: https://doi.org/10.1145/2976749.2978363
- MDN, Content-Security-Policy: https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/Content-Security-Policy
- MDN, script-src-attr: https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/Content-Security-Policy/script-src-attr
- MDN, Strict-Transport-Security: https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/Strict-Transport-Security
- MDN, Referrer-Policy: https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/Referrer-Policy
- MDN, Permissions-Policy: https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/Permissions-Policy
- Permissions Policy (W3C): https://www.w3.org/TR/permissions-policy/
- MDN, Cross-Origin-Opener-Policy: https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/Cross-Origin-Opener-Policy
- MDN, X-Frame-Options: https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/X-Frame-Options
- OWASP HTTP Headers Cheat Sheet: https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Headers_Cheat_Sheet.html
- OWASP Content Security Policy Cheat Sheet: https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html
- Curlie directory license: https://curlie.org/docs/en/license.html

- Wilson, E. B. (1927). "Probable Inference, the Law of Succession, and Statistical Inference." Journal of the American Statistical Association 22(158): 209-212. https://doi.org/10.1080/01621459.1927.10502953

Data and reproducibility

Artifacts: the report, the de-identified dataset, scan and analysis scripts, the random seed and a data dictionary are published at https://rackcrunch.com/security-headers-2026/.

Sample drawn from the Curlie directory (https://curlie.org), used under the Creative Commons Attribution 3.0 Unported License. Sampling frame: the files in the curlie-rdf-all.tar.gz archive dated 2026-02-02, downloaded 2026-09-24 from https://curlie.org/download.

Errata

- 2026-09-24: terminology for the analysis bases standardized across all artifacts; no numbers changed.
- 2026-09-24 (v7.4): removed the eight named qualifying sites from the data release, and confirmed the report names none of them, so the de-identification claim holds; no numbers changed.
- 2026-09-24 (v7.4.1): added the redirect count to the list of released fields; no numbers changed.

Notes

1. Principal base, also called `dedup-200`: unique final registrable domains among the HTTP-200 responses (n=4,688), one row kept per final registrable domain, the last row in scan-2 check-completion order. The label, sector and state figures use all 4,701 HTTP-200 responses, as labeled. All-usable results (n=5,642) are in the sensitivity analyses.
2. The seven criteria are HSTS, CSP, clickjacking, nosniff, Referrer-Policy, Permissions-Policy and COOP, graded under the corrected rubric in the Rubric section. Our first draft reported 47.7% under an earlier rubric that double-counted one CSP directive. The corrected figure is slightly higher.
3. Principal base. The ranking uses exact strings, so a trailing character splits a policy: `upgrade-insecure-requests;` with a semicolon is fourth, at 89 sites. Merging such variants, bare upgrade-insecure-requests would be first at 219 and `frame-ancestors 'self'` second at 197.