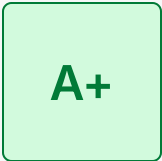


HTTP Observatory Report

 Report Feedback




Scan summary: customercheckin.com








Score: 130 / 100
Scan Time: 3 minutes ago
Tests Passed: 10 / 10

Scan results


Scoring

Test	Score	Reason	Recommendation
<u>Content Security Policy (CSP)</u>	+10 	Content Security Policy (CSP) implemented with default-src 'none' , no 'unsafe' and form-action is set to 'none' or 'self'	None
<u>Cookies</u>	-	No cookies detected	None
<u>Cross Origin Resource Sharing (CORS)</u>	0 	Content is not visible via cross-origin resource sharing (CORS) files or headers.	None
<u>Redirection</u>	0 	Initial redirection is to HTTPS on same host, final destination is HTTPS	None

Test	Score	Reason	Recommendation
<u>Referrer Policy</u>	+5 	Referrer-Policy header set to no-referrer , same-origin , strict-origin or strict-origin-when-cross-origin .	None
<u>Strict Transport Security (HSTS)</u>	0 	Strict-Transport-Security header set to a minimum of six months (15768000).	Consider preloading: this requires adding the preload and includeSubDomains directives and setting max-age to at least 31536000 (1 year), and submitting your site to https://hstspreload.org/ .
<u>Subresource Integrity</u>	-	Subresource Integrity (SRI) not implemented, but all scripts are loaded from a similar origin.	Add SRI for bonus points.
<u>X-Content-Type-Options</u>	0 	X-Content-Type-Options header set to nosniff .	None
<u>X-Frame-Options</u>	+5 	X-Frame-Options (XFO) implemented via the CSP frame-ancestors directive.	None
<u>Cross Origin Resource Policy</u>	+10 	Cross Origin Resource Policy (CORP) implemented, prevents leaks into cross-origin contexts.	None

CSP analysis

 Content Security Policy (CSP) implemented with **default-src 'none'**, no **'unsafe'** and form-action is set to **'none'** or **'self'**

Test	Result	Info
Blocks execution of inline JavaScript by not allowing 'unsafe-inline' inside script-src		Blocking the execution of inline JavaScript provides CSP's strongest protection against cross-site scripting attacks. Moving JavaScript to external files can also help make your site more maintainable.

Test	Result	Info
Blocks execution of JavaScript's eval() function by not allowing 'unsafe-eval' inside script-src	✓	Blocking the use of JavaScript's eval() function can help prevent the execution of untrusted code.
Blocks execution of plug-ins, using object-src restrictions	✓	Blocking the execution of plug-ins via object-src 'none' or as inherited from default-src can prevent attackers from loading Flash or Java in the context of your page.
Blocks inline styles by not allowing 'unsafe-inline' inside style-src	✓	Blocking inline styles can help prevent attackers from modifying the contents or appearance of your page. Moving styles to external stylesheets can also help make your site more maintainable.
Blocks loading of active content over HTTP or FTP	✓	Loading JavaScript or plugins can allow a man-in-the-middle to execute arbitrary code or your website. Restricting your policy and changing links to HTTPS can help prevent this.
Blocks loading of passive content over HTTP or FTP	✓	This site's Content Security Policy allows the loading of passive content such as images or videos over insecure protocols such as HTTP or FTP. Consider changing them to load them over HTTPS.
Clickjacking protection, using frame-ancestors	✓	The use of CSP's frame-ancestors directive offers fine-grained control over who can frame your site.
Deny by default, using default-src 'none'	✓	Denying by default using default-src 'none' can ensure that your Content Security Policy doesn't allow the loading of resources you didn't intend to allow.
Restricts use of the <base> tag by using base-uri 'none' , base-uri 'self' , or specific origins.	✓	The <base> tag can be used to trick your site into loading scripts from untrusted origins.
Restricts where <form> contents may be submitted by using form-action 'none' , form-action 'self' , or specific URIs	✓	Malicious JavaScript or content injection could modify where sensitive form data is submitted to or create additional forms for data exfiltration.

Test	Result	Info
Uses CSP3's ' strict-dynamic ' directive to allow dynamic script loading (optional)	-	' strict-dynamic ' lets you use a JavaScript shim loader to load all your site's JavaScript dynamically, without having to track script-src origins.

Cookies

No cookies detected

Raw server headers

Header	Value
Date	Fri, 29 Aug 2025 16:06:25 GMT
Vary	Accept-Encoding
Server	Apache/2.4.62 (Rocky Linux) OpenSSL/3.2.2
Upgrade	h2
Connection	Upgrade, close
Content-Type	text/html; charset=UTF-8
Accept-Ranges	bytes
Content-Length	4458
Referrer-Policy	same-origin
X-Frame-Options	SAMEORIGIN
X-Content-Type-Options	nosniff

Header	Value
Content-Security-Policy	default-src 'none';font-src 'self';style-src 'self' *.stripe.com;img-src 'self' data: *.stripe.com;connect-src 'self'; form-action 'self' *.mobilecheckin.net *.stripe.com;object-src 'none';base-uri 'none';script-src 'self' 'nonce-aLHQAfbN4JXGd-cuHfK69AAAAII' *.stripe.com; media-src 'self';frame-src 'self' *.stripe.com; frame-ancestors *.stripe.com;
Strict-Transport-Security	max-age=63072000; includeSubdomains; preload
Cross-Origin-Opener-Policy	same-origin
Cross-Origin-Embedder-Policy	require-corp
Cross-Origin-Resource-Policy	same-origin

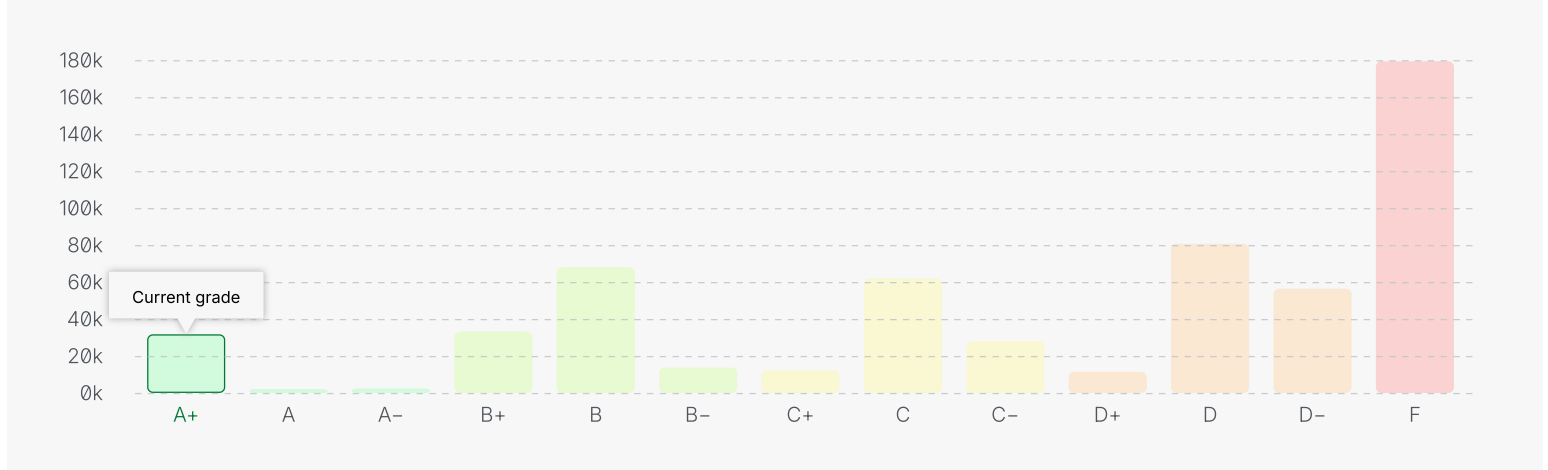
Scan history

Changes in score over time

Date	Score	Grade
May 21, 2025, 3:49:09 PM	130	A+
Mar 5, 2025, 2:05:19 PM	125	A+
Mar 5, 2025, 2:03:33 PM	115	A+
Mar 5, 2025, 1:52:57 PM	110	A+
Mar 5, 2025, 1:46:52 PM	105	A+
Mar 5, 2025, 1:41:20 PM	110	A+

Benchmark comparison

Performance trends from the past year



Refer to this graph to assess the website's current status. By following the recommendations provided and rescanning, you can expect an improvement in the website's grade.



Your blueprint for a better internet.