



Security Assessment Report

Powered by Barrion.io

tomachie.com

2026-10-03 00:52:40 UTC



178 / 180 points

HIGHEST SEVERITY

Medium

RATING

A

SCORE TREND

Improving

What improved

Trusted Types readiness

New issues

None

Open issues

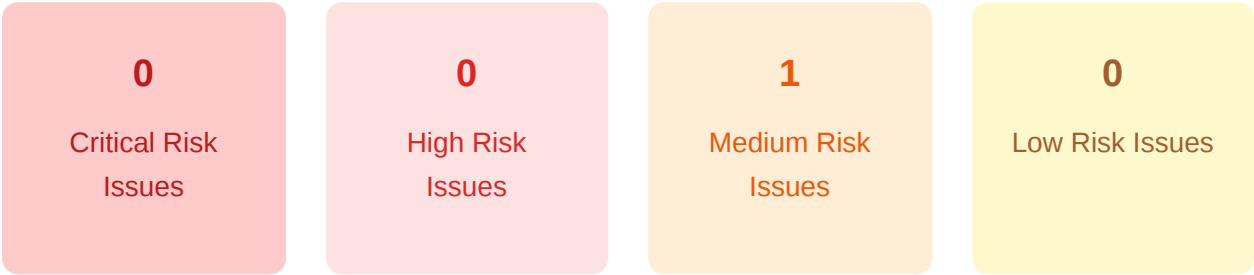
Content Security Policy (CSP) config

Free security report. Upgrade at barrion.io for full audit-ready reports and continuous monitoring.

Executive Summary

This assessment identifies vulnerabilities and risks in your web application that could affect business operations, data integrity, and security posture. Your overall security score is 99% (Rating A).

Security Findings Overview



Understanding the Risks

The identified vulnerabilities can expose your application to serious security threats, including:

- **Unauthorized access** to sensitive customer data and system resources
- **Malicious code injection** that could compromise application integrity
- **Service disruptions** that impact business operations
- **Damage to brand reputation** and loss of customer trust
- **Compliance violations** leading to regulatory penalties

Security Checks

Content Security Policy (CSP) config

Failed

Medium Severity

Description

CSP (Content Security Policy) is a security control mechanism that helps websites defend against attacks such as Cross-Site Scripting (XSS) and data injection attacks. It does this by defining trusted sources for content loading and execution. CSP acts as a security guard that prevents unauthorized script execution and resource loading from malicious origins. While missing or misconfigured CSP doesn't create a vulnerability itself, it represents a security gap that reduces your defense-in-depth posture. Proper CSP configuration significantly enhances website security by mitigating the impact of potential XSS vulnerabilities in your application code. It's an important component of a robust web security policy.

Finding

Content Security Policy is configured but contains insecure directives that reduce protection effectiveness.

Security Value

default-src 'self'; style-src 'self' 'unsafe-inline'; img-src 'self' data: blob:; font-src 'self' data:; connect-src 'self'; object-src 'none'; base-uri 'self'; frame-ancestors 'self'; script-src 'self'; require-trusted-types-for 'script'; trusted-types default 'allow-duplicates'

Recommendation

Your current Content Security Policy header has security vulnerabilities that need to be addressed. The Content Security Policy header is a security feature that helps prevent Cross-Site Scripting (XSS) attacks by controlling which resources can be loaded by your web page. See the specific CSP issues at the subchecks section below.

Recommended Actions:

1. Review your Content Security Policy header - Examine your Content Security Policy directives for security weaknesses
2. Remove unsafe directives like `unsafe-inline` and `unsafe-eval` - These directives allow potentially dangerous inline scripts and dynamic code evaluation
3. Use specific sources instead of wildcards (`*`) where possible - Wildcards allow resources from any origin, reducing security
4. Test in Report-Only mode first: `Content-Security-Policy-Report-Only` - This allows you to test Content Security Policy changes without breaking your website
5. Monitor browser console for Content Security Policy violations during testing - Check for blocked resources and adjust your policy accordingly

Example Secure CSP:

```
Content-Security-Policy: default-src 'self'; script-src 'self' https://trusted-cdn.com; style-src 'self' 'unsafe-inline'; img-src 'self' data: https:; font-src 'self'; connect-src 'self'; frame-ancestors 'none';
```

CVSS: Base Score: 5.4 (Severity: MEDIUM), Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:L/A:N

Recommendation Note

When implementing CSP, ensure that the policy is correctly configured to allow necessary content sources while blocking unnecessary, malicious and unauthorized sources. Implementing a robust CSP is essential for security but can break functionality if not configured correctly. Test your policy carefully, starting with the "[Content-Security-Policy-Report-Only](#)" header to identify issues without enforcing restrictions, and adjust as needed to avoid disrupting your

website. Configure the configuration value to maintain a balance between functionality, development experience, and your preferred level of security.

Detailed Content Security Policy (CSP) config Findings

script-src

Failed

Medium Severity

Security Value

'self'

Recommendation

'self' can be problematic if you host JSONP, AngularJS or user uploaded files. Consider changing this.

Access-Control-Allow-Credentials header

Requires upgrade

Description

The "Access-Control-Allow-Credentials" (ACAC) response header indicates whether the response to the request can be exposed when the credentials flag is true. Do not combine credentials with a wildcard Access-Control-Allow-Origin (*) since browsers will block this configuration.

Access-Control-Allow-Headers header

Requires upgrade

Description

The "Access-Control-Allow-Headers" (ACAH) response header lists permitted request headers for preflighted CORS (Cross-Origin Resource Sharing) requests.

Access-Control-Allow-Origin header

Requires upgrade

Description

The "Access-Control-Allow-Origin" (CAAO) response header facilitates Cross-Origin Resource Sharing (CORS) by specifying which other origins are permitted to access resources from a server. It governs whether a web browser allows JavaScript code from one domain to interact with resources hosted on another domain. Proper configuration of this header is vital for safeguarding against unauthorized access and ensuring the protection of sensitive data.

Access-Control-Expose-Headers header

Requires upgrade

Description

The "Access-Control-Expose-Headers" (ACAEH) response header indicates which headers are safe to expose to the browser.

Access-Control-Max-Age header

Requires upgrade

Description

The "Access-Control-Max-Age" (ACAMA) response header defines how long the results of a preflight request can be cached.

Anti-CSRF Tokens

Requires upgrade

Description

Anti-CSRF (Cross-Site Request Forgery) tokens are unique, randomly generated values associated with user sessions or actions. They prevent unauthorized actions by validating the token values included in requests. CSRF attacks trick users into performing unwanted actions on websites where they're authenticated. Implementing these tokens mitigates CSRF attack risks and enhances web application security.

Certificate Hostname & Chain

Requires upgrade

Description

If the TLS certificate doesn't match your domain or the server doesn't send the full intermediate chain to a trusted root, browsers can't verify the connection. Users will see certificate errors or failed connections, and if warnings are bypassed, sensitive data could be exposed. Keep the certificate valid for the hostname and include all required intermediates. SAN (Subject Alternative Name) or CN (Common Name) must include the hostname.

Cross-Origin-Embedder-Policy header

Requires upgrade

Description

The "Cross-Origin-Embedder-Policy" (COEP) response header is a security mechanism aimed at preventing certain types of security vulnerabilities, such as cross-origin information leaks and cross-site scripting (XSS) attacks. The header allows web developers to declare their desired level of isolation for cross-origin content embedded within their pages, thereby enhancing the security posture of web applications.

Cross-Origin-Opener-Policy header

Requires upgrade

Description

The "Cross-Origin-Opener-Policy" (COOP) response header is a security feature that isolates your resources against attacks such as Spectre. It controls the behavior of window references to other windows. By setting this header, you can ensure that your window's global object will be in a separate browsing context group, preventing other windows from directly referencing it. This can help mitigate certain types of attacks by isolating your window's process from others.

Cross-Origin-Resource-Policy header

Requires upgrade

Description

The "Cross-Origin-Resource-Policy" (CORP) response header allows you to prevent other origins from loading your resources, helping to enforce stronger cross-origin isolation in combination with the COEP header.

DNS Security

Requires upgrade

Description

DNS (Domain Name System) security encompasses various measures to protect against DNS-based attacks such as DNS spoofing (redirecting users to malicious sites), cache poisoning (corrupting DNS cache with false information), and DNS amplification attacks (using DNS servers to flood targets with traffic). Implementing DNSSEC (DNS Security Extensions) adds cryptographic signatures to DNS records to verify their authenticity, using secure DNS resolvers (like Cloudflare 1.1.1.1 or Google 8.8.8.8), and monitoring for suspicious DNS activity are essential for maintaining DNS security.

Email Domain Security

Requires upgrade

Description

Email Domain Security encompasses three key protocols: SPF (Sender Policy Framework) - specifies which servers can send emails for your domain, DKIM (DomainKeys Identified Mail) - adds digital signatures to verify email authenticity, and DMARC (Domain-based Message Authentication, Reporting, and Conformance) - tells receiving servers how to handle emails that fail SPF or DKIM checks. These protocols work together to prevent email spoofing, phishing, and other email-based attacks by verifying the authenticity of email senders and ensuring the integrity of email messages.

OCSP Stapling

Requires upgrade

Description

OCSP (Online Certificate Status Protocol) stapling lets your server send a recent certificate revocation status during the TLS handshake, so browsers don't contact the certificate authority themselves. This improves privacy and performance by avoiding third-party lookups and reducing latency. Note that stapling only applies when your CA operates OCSP responders: Let's Encrypt retired OCSP in 2025 in favor of CRLs, so its certificates cannot be stapled.

Open Ports

Requires upgrade

Description

Open ports on a server can expose services and applications that may be vulnerable to attacks. Common ports like 21 (FTP), 22 (SSH), 23 (Telnet), 25 (SMTP), 80 (HTTP), 443 (HTTPS), 3306 (MySQL), and 5432 (PostgreSQL) should be carefully managed. Unnecessary open ports increase the attack surface and should be closed to reduce security risks.

Subdomain Takeover

Requires upgrade

Description

Subdomain takeover occurs when a subdomain points to a service that no longer exists, allowing attackers to claim control of the subdomain. This can lead to phishing attacks, cookie theft, and other security breaches. Regular monitoring of DNS records and ensuring all subdomains point to valid services is crucial for preventing subdomain takeover attacks.

Vary: Origin header (CORS caching)

Requires upgrade

Description

The "Vary" response header with value "Origin" tells caches (CDNs, proxies, browsers) to store separate copies of a response per requesting Origin when your server returns a specific "Access-Control-Allow-Origin" value (not ""). Without it, a response cached for one origin could be served to another, potentially leaking data.

Vulnerable JS Libraries

Requires upgrade

Description

JavaScript libraries with vulnerabilities may pose significant security risks to web applications. These vulnerabilities usually exist in outdated library versions. Therefore, it's crucial for developers to regularly audit and update their JavaScript libraries to mitigate these risks.

Trusted Types readiness

Passed

Fixed

Description

Trusted Types is enforced via the `Content-Security-Policy` response header using the directives `require-trusted-types-for 'script'` and optional `trusted-types <policy-names>`. It prevents DOM-based XSS (Cross-Site Scripting) by blocking writes to dangerous sinks unless values are created by approved policies.

Finding

This check passed, no issues found.

Security Value

`require-trusted-types-for='script'` | `trusted-types policies: default, 'allow-duplicates' | allow-duplicates`

Recommendation

Advisory: remove `allow-duplicates` unless strictly required.

HTTPS enabled

Passed

Description

HTTPS (Hypertext Transfer Protocol Secure) is a secure version of HTTP that encrypts data transmitted between a user's browser and the website they are visiting. It ensures data security, protects against attacks, enhances user trust, and offers SEO benefits. Implementing HTTPS is essential for safeguarding sensitive information and maintaining the integrity of web communications.

Finding

This check passed, no issues found.

Security Value

Hops: 0

TLS Version

Passed

Description

TLS (Transport Layer Security) is the protocol that ensures privacy and data integrity between communicating applications over the Internet. It's crucial to use the latest TLS version as it provides protection against known vulnerabilities and offers the most up-to-date security features. Older versions may have weaknesses that could be exploited by attackers.

Finding

This check passed, no issues found.

Security Value

TLSv1.3

Certificate Expiry

Passed

Description

SSL/TLS certificates have a validity period, after which they expire. Certificate expiry checks ensure that the server's certificate is current and hasn't exceeded its validity period. Using an expired certificate can lead to security warnings in browsers, loss of user trust, and potentially leave connections vulnerable to interception. Regular monitoring and timely renewal of certificates are crucial for maintaining a secure and trusted web presence.

Finding

This check passed, no issues found.

Security Value

2027-01-15T23:59:59.000Z

Cipher Suite

Passed

Description

A cipher suite is a set of cryptographic algorithms used to secure network connections in TLS (Transport Layer Security). It typically includes algorithms for key exchange, bulk encryption, and message authentication. The strength of a TLS connection largely depends on the security of the chosen cipher suite. Using strong, modern cipher suites is essential for maintaining robust security in web communications.

Finding

This check passed, no issues found.

Security Value

TLS_AES_256_GCM_SHA384

Content-Type header

Passed

Description

The "Content-Type" response header specifies the type of content being served by the server. It is crucial for both security and browser compatibility as it ensures that browsers interpret the content correctly. Omitting this header may lead to unexpected behavior and security vulnerabilities.

Finding

This check passed, no issues found.

Security Value

text/html; charset=UTF-8

Cross-Origin Resource Isolation

Passed

Description

When the "Cross-Origin-Embedder-Policy" response header is set to "require-corp", any cross-origin resource (scripts, images, iframes, etc.) must either be served from the same origin, explicitly allowed via the Cross-Origin-Resource-Policy (CORP) header, or fetched with proper CORS headers that grant access to your origin. Otherwise, the browser will block the resource to preserve cross-origin isolation.

Finding

This check passed, no issues found.

CSP Bypass Detection

Passed

Description

CSP Bypass Detection actively tests your Content Security Policy for common attack vectors that could circumvent its protection. It identifies specific vulnerabilities in your CSP configuration that attackers could exploit to inject malicious scripts despite having a policy in place, providing actionable remediation steps to strengthen your defenses against sophisticated XSS attacks.

Finding

This check passed, no issues found.

Security Value

No bypasses detected

Deprecated TLS versions

Passed

Description

Detects whether deprecated TLS versions (TLS 1.0/TLS 1.1) are still accepted by the server. These versions are obsolete and should be disabled in favor of TLS 1.2+ to prevent downgrade and known-protocol attacks.

Finding

This check passed, no issues found.

Security Value

TLS versions: TLSv1.2, TLSv1.3

Frame Security Policy

Passed

Description

Frame Security Policy, encompassing the X-Frame-Options header and frame-ancestors directive within the Content Security Policy (CSP) header, is a security mechanism designed to prevent clickjacking attacks. It allows webmasters to control whether a page can be loaded within a frame or iframe, and from which origins, thereby enhancing the security posture of web applications.

Finding

This check passed, no issues found.

HSTS enabled

Passed

Description

HSTS (HTTP Strict Transport Security) is a security control mechanism that instructs web browsers to only interact with a website over HTTPS connections, even if the user attempts to access it via HTTP. It prevents protocol downgrade attacks and man-in-the-middle attacks by ensuring all communication remains encrypted. While missing HSTS doesn't create a vulnerability itself, it represents a security gap that reduces your defense-in-depth posture. Even when servers redirect HTTP to HTTPS, HSTS provides additional benefits by eliminating the initial HTTP request and ensuring browsers never attempt insecure connections after the first visit.

Finding

This check passed, no issues found.

Security Value

max-age=63072000; includeSubDomains; preload

Mixed Content

Passed

Description

Mixed content occurs when an HTTPS page loads resources (images, scripts, stylesheets, etc.) over HTTP connections. This creates a security vulnerability because while the main page is encrypted, the HTTP resources can be intercepted, modified, or replaced by attackers. Modern browsers block mixed content by default, which can break website functionality. All resources on HTTPS pages should be loaded over HTTPS to maintain the security benefits of encryption and prevent man-in-the-middle attacks.

Finding

This check passed, no issues found.

Security Value

No mixed content detected

Permissions-Policy header

Passed

Description

The "Permissions-Policy" header lets you allow or deny specific browser features (e.g., geolocation, camera, microphone) per page or origin. Tight scoping reduces attack surface and protects privacy by disabling capabilities your app does not need.

Finding

This check passed, no issues found.

Security Value

Permissions-Policy header: camera=(), microphone=(), geolocation=(), usb=(), payment=()

Referrer Policy

Passed

Description

The "Referrer-Policy" header allows website owners to manage the information transmitted in the HTTP request header about the referring URL. It determines what data is included in the "Referer" header when users navigate between pages. This header is crucial for handling privacy and security concerns related to referrer data leakage. It's worth mentioning that the default setting, "strict-origin-when-cross-origin", provides a secure configuration when this policy is not explicitly set.

Finding

This check passed, no issues found.

Security Value

strict-origin-when-cross-origin

Set-Cookie headers

Passed

Description

The "Set-Cookie" HTTP response header is used to send cookies from the server to the user agent, usually for maintaining authentication/session state and user preferences across multiple requests. Proper header configuration is important for security, as cookies often contain sensitive information. The "Secure" and "HttpOnly" attributes should be set to prevent the cookie from being accessed over unencrypted connections or by client-side scripts, respectively. Using the "SameSite=Strict" attribute when possible is also preferred as this can help prevent cross-site request forgery (CSRF) attacks by restricting when cookies are sent in a cross-site context.

Finding

This check passed, no issues found.

X-Content-Type-Options header

Passed

Description

The "X-Content-Type-Options" response header with the value "nosniff" instructs browsers not to perform MIME-sniffing on the response content. This prevents browsers from overriding the declared content-type and helps mitigate various types of attacks, including cross-site scripting (XSS).

Finding

This check passed, no issues found.

Security Value

nosniff

CSP Console Errors

Passed

Description

Content Security Policy (CSP) console errors occur when your CSP policy blocks legitimate resources that your website needs to function properly. These violations appear in the browser's developer console and indicate that scripts, images, stylesheets, or other resources are being prevented from loading. While CSP is essential for security, overly restrictive policies or outdated configurations can silently break website features, degrade user experience, and prevent third-party services (like analytics, widgets, or CDNs) from working correctly. Monitoring and fixing CSP console errors ensures your security policy maintains the right balance between protection and functionality, preventing features from failing without obvious error messages to users.

Finding

This check passed, no issues found.

Server information disclosure

Passed

Description

Exposing server details through headers like "Server" and "Powered-By" can give attackers valuable insights into your server configuration and software, including the operating system. This could potentially make your website a target for exploits that leverage known server vulnerabilities. To mitigate this risk, it's recommended to avoid disclosing such server details.

Finding

This check passed, no issues found.

X-XSS-Protection header (deprecated)

Passed

Description

The "X-XSS-Protection" HTTP response header was created to stop browsers from loading pages when they detect reflected cross-site scripting (XSS) attacks. XSS attacks inject malicious scripts into web pages viewed by other users. However, the use of this header has been deprecated due to its implementation causing more security issues than it solves and its poor compatibility across different browsers. Therefore, it should be avoided and not implemented.

Finding

This check passed, no issues found.

Next Steps

Take control of your security today and eliminate vulnerabilities before attackers exploit them. We recommend addressing the high-severity vulnerabilities detected first to minimize risk. After applying a fix, Barrion's scan can help validate whether the issue has been properly resolved.

To further enhance your security, consider the following options:

Continuous Security Monitoring

Stay protected with Barrion's upgraded plans, to get automated, real-time security monitoring with advanced checks. Receive instant alerts for detected vulnerabilities to mitigate risks quickly.

- ▢ 24/7 vulnerability scanning across all your domains
- ▢ Real-time threat detection & security alerts
- ▢ Detailed reports with remediation guidance

Get Started

Expert Penetration Testing

Uncover hidden security flaws with a manual security assessment by certified penetration testers.

- ▢ Hands-on & in-depth security testing by experts
- ▢ Comprehensive vulnerability report with real-world attack simulations
- ▢ Actionable remediation plan to strengthen security

Request Pentest

Security Consulting

Need help securing your application? Our specialists will help you fix vulnerabilities and implement best security practices.

- ▢ Dedicated security expert for your case
- ▢ Assistance with vulnerability remediation
- ▢ Guidance on industry best practices

Schedule Consultation