DOCUMENTATION SECURITY

Web security checks and protection against attacks

DEMO VERSION

Demo Content

This demo version presents **4 out of 29** checks from the category Security.

The full version includes all **29 checks** checks with:

Generated on: 28.01.2026 23:05:53

FISCAL IDENTITY

PFA BODNARIU RAZVAN TRAIAN

CUI: 52188722 | **Reg. com.:** F2025026473001

Tel: +4 0775 278326 | **Email:** office@webrt.eu

About Security

What is Security?

Web security represents the set of technical measures and configurations that protect a website against malicious attacks, data theft and exploitation of vulnerabilities. A proper security configuration is essential to protect user data and business reputation.

Benefits

- Protection against XSS, CSRF, clickjacking attacks
- Data encryption in transit through HTTPS
- GDPR compliance and other data protection regulations
- Increased user trust
- Prevention of financial and reputation losses

Standards and Guidelines

Checks in this category are based on: **OWASP Top 10, Mozilla Security Guidelines**

Checks in Demo version

4 out of **29** checks

The full version includes all 29 checks.

Security — Demo (4 out of 29 checks)

Verification	Details
Allowed HTTP Methods	**What it means:** Restricting HTTP methods to those necessary prevents attacks that could exploit advanced web server features. **What It Checks:** Checks if only necessary HTTP methods (GET, POST, HEAD) are allowed and dangerous ones (TRACE, DELETE, PUT) are disabled. **Why It's Important:** Dangerous HTTP methods can be exploited for cross-site tracing attacks or unauthorized content modification. **How to Fix:** Add to .htaccess: <LimitExcept GET POST HEAD> Require all denied </LimitExcept> RewriteEngine On RewriteCond %{REQUEST_METHOD} ^(TRACE|DELETE|TRACK) [NC] RewriteRule .* - [F] **Official Documentation:** • OWASP Testing for HTTP Methods • Apache Limit Module Documentation
CSP Upgrade Insecure Requests	**What it means:** This CSP directive automatically upgrades all HTTP resources on your site to HTTPS, eliminating mixed content security issues. **What It Checks:** Checks the use of the upgrade-insecure-requests directive in Content Security Policy. **Why It's Important:** The upgrade-insecure-requests directive automatically forces all HTTP resources to be loaded via HTTPS, eliminating mixed content. **How to Fix:** Enable upgrade-insecure-requests for automatic HTTP→HTTPS conversion. Apache: Header always set Content-Security-Policy "upgrade-insecure-requests". Nginx: add_header Content-Security-Policy "upgrade-insecure-requests" always;. Alternative meta tag: <meta http-equiv="Content-Security-Policy" content="upgrade-insecure-requests">. Test mode: Content-Security-Policy-Report-Only with report-uri. Test in DevTools Security/Console. **Official Documentation:** • W3C Upgrade Insecure Requests • Google Mixed Content Guide
Complete Security Headers	**What it means:** This complete security configuration for .htaccess provides comprehensive protection against the most common web attacks. **What It Checks:** Checks if all essential security headers are configured correctly. **Why It's Important:** A complete set of security headers provides defense in depth against various types of web attacks. **How to Fix:** Configure .htaccess completely: HTTPS redirect (RewriteCond %{HTTPS} off), Security headers (X-Frame-Options SAMEORIGIN, X-Content-Type-Options nosniff, X-XSS-Protection, HSTS, CSP), Hide server (Header unset Server), Secure cookies (HttpOnly;Secure), File protection (FilesMatch .env|.htaccess|.ini), Directory protection (Options -Indexes), HTTP methods restriction (LimitExcept GET POST HEAD), Disable TRACE. Check: securityheaders.com. **Official Documentation:** • OWASP Secure Headers Project • Mozilla Security Guidelines • Apache Security Configuration

Verification	Details
Content Security Policy	**What it means:** Content Security Policy controls what resources (JavaScript, CSS, images) your site can load, preventing malicious attacks. **What It Checks:** Checks if the site has Content Security Policy (CSP) configured to prevent XSS and injection attacks. **Why It's Important:** CSP is a powerful defense barrier against XSS, clickjacking, and other injection attacks. **How to Fix:** Add to .htaccess: Header always set Content-Security-Policy "default-src 'self'; script-src 'self' 'unsafe-inline'; style-src 'self' 'unsafe-inline'; img-src 'self' data:;" **Official Documentation:** • CSP Quick Reference Guide • Google CSP Evaluator Tool

Available in full version

The category contains **Security 25 additional checks** in the full version.

Checks available in full version:

- Content Security Policy Enhanced
- Cookie Flags
- Cookie Secure
- Cross Origin Headers
- DNS Info
- Directory Browsing
- Expect Ct
- HSTS Preload
- HTTPS Redirect
- Hpkp Header
- Mixed Content
- Permissions Policy
- Referrer Policy
- SSL Certificate
- Security Headers
- Sensitive Files Exposed
- Server Signature
- Set Cookie
- Strict Transport Security
- TLS Version
- Trace Method
- XContent Type Options
- XFrame Options
- XPermitted Policies
- XXss Protection

Each check includes: detailed explanations, business impact,
technical recommendations, implementation examples and links to official documentation.

Contact us for the full version: office@webrt.eu