Your Business
August 30, 2026 · 12:49 PM

SECURITY REPORT

3
/10
MEDIUM

SCAN TARGET

xturbo.com.br

Open ports detected: **0** · Public IP: **147.79.94.195**

A risk score of 3/10 (MEDIUM) indicates your business has security issues that need attention. The higher the score, the greater the chance of a breach, ransomware, or data theft.

EXECUTIVE SUMMARY

The security scan of xturbo.com.br found 19 medium and low severity issues. No critical vulnerabilities were detected, but 6 issues should be addressed this week to harden your security posture. Detailed fix instructions are included for each finding.

✓ No high-risk open ports were detected from the internet — your firewall appears to be blocking dangerous services.

TOP RECOMMENDATIONS

1. Missing HSTS Header: Add the HSTS header to your web server.
2. Missing Clickjacking Protection (X-Frame-Options): Add the X-Frame-Options header.
3. Exposed .git Directory (Unconfirmed): Block public access to /.

DETAILED FINDINGS

Missing HSTS Header

Port HTTPS · HSTS Header

MEDIUM

WHAT IT IS	Missing Strict-Transport-Security (HSTS) — browsers aren't forced to always use HTTPS, leaving visitors open to downgrade attacks
BUSINESS RISK	An attacker on the same network as a visitor (public wifi, a compromised router, etc.) can trick their browser into using the insecure version of your site and intercept what they type — raising the odds of stolen logins or payment details.
REAL-WORLD EXAMPLE	Example: An attacker on a shared network intercepts a visitor's first request (which defaults to HTTP) before the redirect happens, and silently serves them a fake version of the page instead of the real site.
HOW TO FIX	Add the HSTS header to your web server. Nginx: add 'add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;' in your server block. Apache: add 'Header always set Strict-Transport-Security "max-age=31536000"' in your config. Cloudflare: SSL/TLS > Edge Certificates > enable HTTP Strict Transport Security.
URGENCY	**Fix within 1 week**
REFERENCE	CWE-319 — Cleartext Transmission of Sensitive Information Sensitive data is sent over the network unencrypted, so anyone monitoring the connection can read it.

Missing Clickjacking Protection (X-Frame-Options)

Port HTTPS · Clickjacking Protection

MEDIUM

WHAT IT IS	Missing X-Frame-Options — your website can be embedded in an attacker's invisible iframe to trick users into unwanted actions
BUSINESS RISK	An attacker could trick your customers into clicking hidden buttons — like 'change password' or 'confirm purchase' — without realizing it, potentially leading to account takeovers or unauthorized actions carried out under your brand's name.
REAL-WORLD EXAMPLE	Example: An attacker embeds the site's 'delete account' or 'confirm payment' button inside an invisible iframe on their own page, disguised under something like a fake 'play video' button — a visitor's real click triggers the hidden action on your site.
HOW TO FIX	Add the X-Frame-Options header. Nginx: 'add_header X-Frame-Options "SAMEORIGIN" always;' Apache: 'Header always set X-Frame-Options SAMEORIGIN' This tells browsers to only allow your site to be framed by pages on the same domain.
URGENCY	**Fix within 1 week**
REFERENCE	CWE-1021 — Improper Restriction of Rendered UI Layers (Clickjacking) The site doesn't prevent itself from being embedded inside another page, which attackers can exploit to trick users into clicking something they didn't intend to.

Exposed .git Directory (Unconfirmed)

Port HTTPS · Exposed .git Directory

MEDIUM

WHAT IT IS	The path /.git/HEAD returns a restricted-access response (401/403) that is genuinely distinct from how this site handles random nonexistent paths — this confirms *something* is being specially handled at this path, but the response contains no content confirming what it actually is — this exposes the site's Git repository metadata, not an admin login
BUSINESS RISK	This is not a login page to brute-force — a publicly accessible /.git/HEAD can let an attacker reconstruct this site's Git repository: full source code, configuration files, commit history, internal file paths, and any secrets that were ever committed, even ones later removed from the latest version but never rotated.
REAL-WORLD EXAMPLE	Example: A free, widely available tool rebuilds this site's entire source code and commit history from the exposed .git folder within minutes — sometimes turning up a hardcoded API key or password that was deleted from the current code but never actually rotated.
HOW TO FIX	Block public access to /.git/HEAD entirely — it should never be servable. Nginx: 'location ~ /\.git { deny all; }'. Apache: '<DirectoryMatch "\.git"> Require all denied </DirectoryMatch>'. Better yet, don't deploy the .git folder to the live server at all. If any credentials were ever committed to this repo, rotate them — treat them as compromised.
URGENCY	**Fix within 1 week**
REFERENCE	CWE-527

DMARC Monitor-Only (p=none) — Not Enforced

Port DNS · DMARC Policy

MEDIUM

WHAT IT IS	DMARC is set to monitor-only (p=none) — phishing emails pretending to be you aren't blocked, just reported
BUSINESS RISK	Phishing emails pretending to be your business can still reach customers' inboxes today — you'll get aggregate reports about it after the fact (if 'rua=' reporting is configured), but nothing actually stops the fraudulent emails from being delivered right now. p=none is a legitimate and recommended first step — it lets you review reports and confirm all your real mail sources pass DMARC alignment (the From-header domain matching either an aligned SPF pass or an aligned DKIM signature) before you start blocking anything — the risk is only in staying at p=none indefinitely instead of using it as a monitoring phase.
REAL-WORLD EXAMPLE	Example: Forged emails impersonating the business keep reaching customers' inboxes; DMARC reports quietly pile up showing exactly that it's happening, but because the policy is monitor-only, nothing actually blocks a single one of them.
HOW TO FIX	Review DMARC aggregate reports (they require 'rua=mailto:...' in the record — this record doesn't appear to have one, add it first) for a few weeks to confirm every legitimate mail source for this domain is passing DMARC alignment. Once confirmed, move to 'p=quarantine' (suspicious mail goes to spam) and monitor again before finally moving to 'p=reject' (spoofed mail is rejected outright). Don't jump straight to p=reject — if a legitimate sender was missed, that skips straight to real mail being dropped. Example: 'v=DMARC1; p=quarantine; rua=mailto:dmarc@yourdomain.com'
URGENCY	**Fix within 1 week**
REFERENCE	CWE-290 — Authentication Bypass by Spoofing The login or identity check can be tricked by faking trusted information, letting an attacker in without real credentials.

CSP Missing script-src Restriction

Port HTTPS · Content Security Policy

MEDIUM

WHAT IT IS	CSP is set but has no script-src or default-src directive — script execution isn't restricted at all
BUSINESS RISK	If an attacker compromises any third-party script you load — a classic 'watering hole' tactic, where they hijack a widget or library your site trusts instead of attacking you directly — nothing stops that code from running on every visitor's browser and stealing logins or payment data.
REAL-WORLD EXAMPLE	Example: A third-party widget the site trusts and loads on every page gets compromised by attackers upstream; because nothing restricts which scripts can run, the malicious update executes on every visitor's browser the next time they load the page.
HOW TO FIX	Add a script-src directive listing only the exact domains you actually load scripts from, e.g. "script-src 'self' https://cdn.yourtrustedvendor.com;". Avoid 'unsafe-inline' and wildcards.
URGENCY	**Fix within 1 week**
REFERENCE	CWE-693 — Protection Mechanism Failure A security safeguard that should be protecting the system is missing, disabled, or not strong enough.

Third-Party Scripts Missing Subresource Integrity (SRI)

Port HTTPS · Third-Party Script Loading

MEDIUM

WHAT IT IS	3 third-party script source(s) loaded without Subresource Integrity (SRI): cdnjs.cloudflare.com, www.googletagmanager.com, xturbo.matrixdobrasil.ai
BUSINESS RISK	If any of these third-party providers is ever compromised — a real, recurring attack pattern called a 'watering hole' or supply-chain attack, where attackers hit a widely-trusted vendor instead of you directly — the malicious code they inject would run on your site with no verification and no warning to you or your visitors.
REAL-WORLD EXAMPLE	Example: A widely-used analytics or widget provider gets compromised (a real, recurring attack pattern), and because the script loads without integrity verification, the malicious version executes on every visitor's browser with no warning to you or them.
HOW TO FIX	Add integrity and crossorigin attributes to each third-party <script> tag, e.g. <script src="..." integrity="sha384-..." crossorigin="anonymous"></script>. Most CDNs (cdnjs, jsdelivr, unpkg) publish the correct hash on their site — copy it directly.
URGENCY	**Fix within 1 week**
REFERENCE	CWE-353 — Missing Support for Integrity Check There's no way to verify that data wasn't altered in transit, so tampering would go unnoticed.

Missing MIME Sniffing Protection

Port HTTPS · MIME Sniffing — LOW

WHAT IT IS	Missing X-Content-Type-Options — browsers may guess file types incorrectly, which can enable content injection
BUSINESS RISK	This is a minor gap on its own, but it slightly raises the odds that a malicious file could be misread as something else by a visitor's browser, helping a separate attack succeed.
REAL-WORLD EXAMPLE	Example: A user-uploaded file intended to be harmless (like an image) is reinterpreted by the browser as executable script because the server never told it what the file actually was, letting an unrelated vulnerability turn into a working attack.
HOW TO FIX	Add: 'add_header X-Content-Type-Options "nosniff" always;' (Nginx) or 'Header always set X-Content-Type-Options nosniff' (Apache). This is a one-line fix that takes 2 minutes.
URGENCY	**Fix within 1 month**
REFERENCE	CWE-693 — Protection Mechanism Failure A security safeguard that should be protecting the system is missing, disabled, or not strong enough.

Missing Referrer-Policy Header

Port HTTPS · Referrer Policy — LOW

WHAT IT IS	Missing Referrer-Policy — page URLs (which may include sensitive data) are shared with third-party sites your pages link to
BUSINESS RISK	If any of your page addresses contain sensitive details (like a password-reset token or account ID), that information could leak to outside sites your pages link to — a small but easily avoidable privacy gap.
REAL-WORLD EXAMPLE	Example: A customer clicks an outbound link from a page whose URL happens to include an account ID or a password-reset token, and that full address — token included — is handed to the destination site in the Referer header.
HOW TO FIX	Add: 'add_header Referrer-Policy "strict-origin-when-cross-origin" always;' (Nginx) or 'Header always set Referrer-Policy strict-origin-when-cross-origin' (Apache).
URGENCY	**Fix within 1 month**
REFERENCE	CWE-16 — Configuration The weakness comes from how the software was configured, not from a flaw in its code.

Missing Permissions-Policy Header

Port HTTPS · Permissions Policy

LOW

WHAT IT IS	Missing Permissions-Policy — browser features like camera, microphone, and location aren't restricted for embedded third-party scripts
BUSINESS RISK	If you ever embed third-party ads, widgets, or analytics scripts, they could request a visitor's camera, microphone, or location without you intending to allow it — an avoidable privacy risk for your customers.
REAL-WORLD EXAMPLE	Example: An embedded ad network's script requests the visitor's location or microphone access through a permission prompt the site owner never intended to allow, simply because nothing in the page's headers restricted it.
HOW TO FIX	Add: 'add_header Permissions-Policy "camera=(), microphone=(), geolocation=()" always;' Adjust based on what your site actually uses. This limits what ad/analytics scripts can access.
URGENCY	**Monitor**
REFERENCE	CWE-693 — Protection Mechanism Failure A security safeguard that should be protecting the system is missing, disabled, or not strong enough.

Technology Stack Disclosed (X-Powered-By)

Port HTTPS · Technology Disclosure

LOW

WHAT IT IS	X-Powered-By header discloses your tech stack (PHP/8.2.30) — makes targeted attacks easier
BUSINESS RISK	Knowing your exact tech stack lets an attacker focus their effort on vulnerabilities specific to that platform, slightly increasing the odds of being targeted compared to a generic, unidentified site.
REAL-WORLD EXAMPLE	Example: Knowing the exact framework and version in use, an attacker skips general reconnaissance and goes straight to testing the specific, publicly known weaknesses for that platform.
HOW TO FIX	Remove the X-Powered-By header. PHP: set 'expose_php = Off' in php.ini. Node/Express: 'app.disable("x-powered-by")'. Nginx: 'more_clear_headers X-Powered-By;' (with headers_more module) or handle in your app.
URGENCY	**Fix within 1 month**
REFERENCE	CWE-200 — Exposure of Sensitive Information The system reveals information to someone who shouldn't have access to it.

Protected Path — /.env

Port PATH · Content Discovery

LOW

WHAT IT IS	/.env returned HTTP 403. This confirms the path exists, but access control prevented access — it was not accessed.
BUSINESS RISK	/.env returned HTTP 403. This confirms the path exists, but access control prevented access — it was not accessed.
HOW TO FIX	No action needed if this is intentionally restricted. Confirm the credentials protecting it are strong and not left at a default.
URGENCY	**Fix within 1 month**
REFERENCE	CWE-284

Protected Path — /wp-config.php.bak

Port PATH · Content Discovery

LOW

WHAT IT IS	/wp-config.php.bak returned HTTP 403. This confirms the path exists, but access control prevented access — it was not accessed.
BUSINESS RISK	/wp-config.php.bak returned HTTP 403. This confirms the path exists, but access control prevented access — it was not accessed.
HOW TO FIX	No action needed if this is intentionally restricted. Confirm the credentials protecting it are strong and not left at a default.
URGENCY	**Fix within 1 month**
REFERENCE	CWE-284

Protected Path — /.git/HEAD

Port PATH · Content Discovery

LOW

WHAT IT IS	/.git/HEAD returned HTTP 403. This confirms the path exists, but access control prevented access — it was not accessed.
BUSINESS RISK	/.git/HEAD returned HTTP 403. This confirms the path exists, but access control prevented access — it was not accessed.
HOW TO FIX	No action needed if this is intentionally restricted. Confirm the credentials protecting it are strong and not left at a default.
URGENCY	**Fix within 1 month**
REFERENCE	CWE-284

Protected Path — /.git/config

Port PATH · Content Discovery

LOW

WHAT IT IS	/.git/config returned HTTP 403. This confirms the path exists, but access control prevented access — it was not accessed.
BUSINESS RISK	/.git/config returned HTTP 403. This confirms the path exists, but access control prevented access — it was not accessed.
HOW TO FIX	No action needed if this is intentionally restricted. Confirm the credentials protecting it are strong and not left at a default.
URGENCY	**Fix within 1 month**
REFERENCE	CWE-284

Protected Path — /.svn/entries

Port PATH · Content Discovery

LOW

WHAT IT IS	/.svn/entries returned HTTP 403. This confirms the path exists, but access control prevented access — it was not accessed.
BUSINESS RISK	/.svn/entries returned HTTP 403. This confirms the path exists, but access control prevented access — it was not accessed.
HOW TO FIX	No action needed if this is intentionally restricted. Confirm the credentials protecting it are strong and not left at a default.
URGENCY	**Fix within 1 month**
REFERENCE	CWE-284

Protected Path — /.htpasswd

Port PATH · Content Discovery

LOW

WHAT IT IS	/.htpasswd returned HTTP 403. This confirms the path exists, but access control prevented access — it was not accessed.
BUSINESS RISK	/.htpasswd returned HTTP 403. This confirms the path exists, but access control prevented access — it was not accessed.
HOW TO FIX	No action needed if this is intentionally restricted. Confirm the credentials protecting it are strong and not left at a default.
URGENCY	**Fix within 1 month**
REFERENCE	CWE-284

No Abuse Reports Found for Public IP

Port HTTPS · IP Reputation

INFO

WHAT IT IS	Public IP 147.79.94.195 (hosted by Brander Group Inc.) has no abuse reports on AbuseIPDB in the last 90 days.
BUSINESS RISK	A clean IP reputation is a good sign — no other indication needed here.
HOW TO FIX	No action required.
URGENCY	Informational

Hosting Network Identified

Port HTTPS · IP WHOIS / Network Ownership

INFO

WHAT IT IS	Public IP 147.79.94.195 is registered to Private Customer (BR) — network: HOSTINGER-HOSTING — block: 147.79.88.0 – 147.79.95.255.
BUSINESS RISK	This is informational — it tells you (and anyone else who looks it up) which hosting provider or network actually operates this IP. Useful for confirming you're hosted where you expect, or for context if the IP reputation check above flags anything.
HOW TO FIX	No action required — this is identifying information, not a vulnerability.
URGENCY	Informational

Detected Technology Stack

Port HTTPS · Technology Fingerprint

INFO

WHAT IT IS	Detected technology stack: Php 8.2.30, Wordpress 7.0.2, Jquery 2.0.3.
BUSINESS RISK	Knowing your exact software versions helps attackers pick targeted exploits instead of guessing — this isn't a vulnerability by itself, but it's worth knowing what your site publicly reveals about its own software.
HOW TO FIX	No action required unless you'd prefer to hide version banners (e.g. removing WordPress's generator meta tag and readme.html, or disabling Apache/nginx's Server header version string) to make automated targeting slightly harder. Keeping the software itself patched matters far more than hiding the version number.
URGENCY	Informational

PORT REFERENCE TABLE

PORT	SERVICE	RISK	DESCRIPTION

HTTPS	HSTS Header	**MEDIUM**	Missing Strict-Transport-Security (HSTS) — browsers aren't forced to always use HTTPS, leaving visit
HTTPS	Clickjacking Protection	**MEDIUM**	Missing X-Frame-Options — your website can be embedded in an attacker's invisible iframe to trick us
HTTPS	Exposed .git Directory	**MEDIUM**	The path /.git/HEAD returns a restricted-access response (401/403) that is genuinely distinct from h
DNS	DMARC Policy	**MEDIUM**	DMARC is set to monitor-only (p=none) — phishing emails pretending to be you aren't blocked, just re
HTTPS	Content Security Policy	**MEDIUM**	CSP is set but has no script-src or default-src directive — script execution isn't restricted at all
HTTPS	Third-Party Script Loading	**MEDIUM**	3 third-party script source(s) loaded without Subresource Integrity (SRI): cdnjs.cloudflare.com, www
HTTPS	MIME Sniffing	**LOW**	Missing X-Content-Type-Options — browsers may guess file types incorrectly, which can enable content
HTTPS	Referrer Policy	**LOW**	Missing Referrer-Policy — page URLs (which may include sensitive data) are shared with third-party s
HTTPS	Permissions Policy	**LOW**	Missing Permissions-Policy — browser features like camera, microphone, and location aren't restricte
HTTPS	Technology Disclosure	**LOW**	X-Powered-By header discloses your tech stack (PHP/8.2.30) — makes targeted attacks easier
PATH	Content Discovery	**LOW**	/.env returned HTTP 403. This confirms the path exists, but access control prevented access — it was
PATH	Content Discovery	**LOW**	/wp-config.php.bak returned HTTP 403. This confirms the path exists, but access control prevented ac
PATH	Content Discovery	**LOW**	/.git/HEAD returned HTTP 403. This confirms the path exists, but access control prevented access — i
PATH	Content Discovery	**LOW**	/.git/config returned HTTP 403. This confirms the path exists, but access control prevented access —
PATH	Content Discovery	**LOW**	/.svn/entries returned HTTP 403. This confirms the path exists, but access control prevented access
PATH	Content Discovery	**LOW**	/.htpasswd returned HTTP 403. This confirms the path exists, but access control prevented access — i
HTTPS	IP Reputation	**INFO**	Public IP 147.79.94.195 (hosted by Brander Group Inc.) has no abuse reports on AbuseIPDB in the last
HTTPS	IP WHOIS / Network Ownership	**INFO**	Public IP 147.79.94.195 is registered to Private Customer (BR) — network: HOSTINGER-HOSTING — block:

HTTPS	Technology Fingerprint	**INFO**	Detected technology stack: Php 8.2.30, Wordpress 7.0.2, Jquery 2.0.3.

NEXT STEPS

→ Address the 6 MEDIUM risk finding(s) this week. Start with 'Missing HSTS Header': Add the HSTS header to your web server.