

SECURITY REPORT



SCAN TARGET

<https://restaurants.applebees.com/en-us/wa/kennewick/606-n-columbia-center-blvd-84061>

Open ports detected: 0 · Public IP: 54.203.238.30

A risk score of 8/10 (CRITICAL) 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 restaurants.applebees.com found 16 issues including 4 high-risk and 4 medium-risk findings. The most critical issue is 'Missing SPF Record — Email Spoofing Possible' which requires immediate attention. Full remediation steps are provided for each finding below.

✓ Your server is reachable, which means the web checks and SSL analysis were able to run successfully.

TOP RECOMMENDATIONS

- 1 Missing SPF Record — Email Spoofing Possible: Add an SPF TXT record to your DNS listing every service that actually sends mail for this domain — leaving one out will cause its mail to fail SPF and possibly get rejected.
- 2 Missing DMARC Record — Email Impersonation Risk: Add a DMARC TXT record to your DNS.
- 3 Exposed Google API Key in Page Source: Revoke/rotate this key immediately in the relevant provider's dashboard, then remove it from the homepage HTML.

DETAILED FINDINGS

Missing SPF Record — Email Spoofing Possible

Port DNS · SPF Record

HIGH

WHAT IT IS	No SPF record — there's no DNS-level list of which mail servers are authorized to send using this domain's envelope sender, so any server can pass an SPF check for your domain
BUSINESS RISK	Without SPF, nothing tells receiving mail servers which senders are legitimate for this domain's SMTP envelope, making it easier for forged mail to get through — and easier for your own legitimate mail to be misjudged as spam. Note that SPF on its own, even correctly configured, doesn't stop the visible 'From:' address a recipient sees from being spoofed — that requires DMARC (see below) to enforce alignment between SPF/DKIM and the From header.
REAL-WORLD EXAMPLE	Example: A customer receives an email that looks exactly like it's from the business — same display name, same domain — asking them to 'confirm' a payment or click a link, with nothing in DNS to stop the forgery or warn the recipient.
HOW TO FIX	Add an SPF TXT record to your DNS listing every service that actually sends mail for this domain — leaving one out will cause its mail to fail SPF and possibly get rejected. Log into your domain registrar, go to DNS, add a TXT record for '@' with a value matching your provider: Google Workspace: 'v=spf1 include:_spf.google.com -all' Microsoft 365: 'v=spf1 include:spf.protection.outlook.com -all'. If you use additional senders (helpdesk, marketing platform, invoicing tool, etc.), add each as its own 'include:' entry in the same record — a domain can only have one SPF TXT record, so don't create a second one. Use https://mxtoolbox.com/spf.aspx to verify, then add a DMARC record too so the visible From address is covered as well.
URGENCY	Fix immediately
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.

Missing DMARC Record — Email Impersonation Risk

Port DNS · DMARC Record

HIGH

WHAT IT IS	No DMARC record — your domain has zero email authentication enforcement, making it trivial to impersonate your business
BUSINESS RISK	This makes it significantly easier for scammers to send convincing fake emails 'from' your business — a common tactic in invoice fraud and phishing — which can directly cost your customers money and damage trust in your brand.
REAL-WORLD EXAMPLE	Example: A scammer sends an invoice-fraud email that appears to come straight from the business's own domain; with no DMARC record in place, nothing tells the recipient's mail provider the message is forged, so it lands in the inbox looking completely legitimate.
HOW TO FIX	Add a DMARC TXT record to your DNS. Go to your domain registrar's DNS settings, add a TXT record for '_dmarc' (not '@') with value: 'v=DMARC1; p=none; rua=mailto:youremail@yourdomain.com' Start at p=none — this only turns on reporting, nothing is blocked yet. Review the aggregate reports for a few weeks to confirm every legitimate mail source for this domain (email provider, helpdesk, marketing tools, etc.) is passing DMARC alignment, then move to p=quarantine, and finally p=reject once you're confident nothing legitimate will be caught. Verify at https://mxtoolbox.com/dmarc.aspx
URGENCY	Fix immediately
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.

Exposed Google API Key in Page Source

Port SECRET · Exposed
Credential

HIGH

WHAT IT IS	A Google API Key was found exposed in the homepage HTML (AlzaSy...p7Ds) — visible to anyone who views the page source
BUSINESS RISK	Anyone who views your page source or downloaded JavaScript can copy this credential and use it directly — potentially running up charges on your account, sending data through your services, or accessing whatever this key controls.
REAL-WORLD EXAMPLE	Example: An automated bot scans newly indexed pages and JavaScript files for exactly this pattern, and the moment a key like this becomes public, it's tested against the relevant provider's API within minutes — sometimes resulting in surprise usage charges or a data breach before anyone at the company notices.
HOW TO FIX	Revoke/rotate this key immediately in the relevant provider's dashboard, then remove it from the homepage HTML. Secret keys must only ever be used in backend/server code, never embedded in HTML or JavaScript sent to the browser. Use environment variables on the server instead.
URGENCY	Fix immediately
REFERENCE	CWE-798 — Use of Hard-coded Credentials A username, password, or key is built directly into the code or configuration, where it can be found and reused by an attacker.

CORS Misconfiguration (Reflected Origin + Credentials)

Port CORS · CORS
Misconfiguration

HIGH

WHAT IT IS	the site reflects any Origin header back in Access-Control-Allow-Origin while also allowing credentials, letting any other website read logged-in users' data through their browser
BUSINESS RISK	A malicious website a logged-in customer visits can silently make authenticated requests to your site from their browser and read the response — potentially stealing account data, orders, or session details without the customer ever noticing.
REAL-WORLD EXAMPLE	Example: A customer logged into your site visits an unrelated, malicious page in another tab; that page silently fires a request to your site using the customer's own session and reads back their account data, with nothing for the customer to notice or approve.
HOW TO FIX	Stop reflecting the request's Origin header back unconditionally. Use an explicit allow-list of trusted origins for Access-Control-Allow-Origin instead of echoing the request's own Origin, and only set Access-Control-Allow-Credentials: true for origins on that list.
URGENCY	Fix immediately
REFERENCE	CWE-942 — Permissive Cross-domain Security Policy with Untrusted Domains The site's cross-domain access policy is configured too loosely, letting untrusted websites interact with it as if they were trusted.

Environment Secrets Exposed (Unconfirmed)

Port HTTPS · Environment
Secrets Exposed

MEDIUM

WHAT IT IS	The path <code>/.env</code> returns a restricted-access response (401/403) that is genuinely distinct from how this site handles random nonexistent paths — this confirms <i>something</i> is being specially handled at this path, but the response contains no content confirming what it actually is — this exposes credentials or sensitive configuration that should never be public
BUSINESS RISK	Exposed credentials in <code>/.env</code> (API keys, database passwords, AWS access keys) give an attacker direct access to your critical systems — databases, cloud infrastructure, third-party APIs, and internal tools. A single exposed credential can lead to total infrastructure compromise.
REAL-WORLD EXAMPLE	Example: An AWS credentials file exposed at <code>/.env</code> contained a key with full EC2/RDS access. An attacker used it to launch crypto miners on the account, incurring \$50k in charges within 72 hours before discovery. The attacker also captured all database dumps and sold them on underground forums.
HOW TO FIX	1. IMMEDIATELY rotate all exposed credentials (API keys, database passwords, SSH keys). Treat them as fully compromised. 2. Block public access to <code>/.env</code> via web server rules (Nginx: <code>location ~ /\.env { deny all; }</code>). 3. Never commit secrets to version control — use <code>.gitignore</code> and environment variable loaders (python-dotenv, Node's dotenv). 4. Use a secrets manager (AWS Secrets Manager, HashiCorp Vault, Doppler) for production credentials.
URGENCY	Fix within 1 week
REFERENCE	CWE-798 — Use of Hard-coded Credentials A username, password, or key is built directly into the code or configuration, where it can be found and reused by an attacker.

DKIM Not Detected Under Common Selectors

Port DNS · DKIM Record

MEDIUM

WHAT IT IS	No DKIM record found under common selector names — this does not confirm DKIM is unconfigured, only that it wasn't found under any of the selector names checked; many providers use a custom or provider-specific selector this check can't guess
BUSINESS RISK	If DKIM genuinely isn't configured, email providers increasingly use it as a trust signal, and its absence can make legitimate emails more likely to be flagged as suspicious or land in spam. But this check only queries a fixed list of common selector names (default, google, mail, etc.) — many providers assign a random or account-specific selector, so this finding should be treated as 'couldn't confirm DKIM,' not 'DKIM is definitely missing.'
REAL-WORLD EXAMPLE	Example: A legitimate invoice email from the business gets flagged as suspicious or dropped into spam by the recipient's mail provider, simply because there's no DKIM signature to prove the message wasn't altered or forged in transit — this only actually happens if DKIM is truly unconfigured, which this check alone can't confirm.
HOW TO FIX	First confirm whether DKIM is actually configured: check your email provider's admin console (Google Workspace: Admin console > Apps > Gmail > Authenticate email; Microsoft 365: Defender > Email authentication > DKIM) for the exact selector name in use, since it's often not one of the common defaults this scan checks. If it turns out DKIM genuinely isn't enabled, turn it on there and add the resulting TXT record to DNS. If you're not sure how to check, share the selector name your provider gives you and this can be verified directly.
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): assets.restaurants.applebees.com, kit.fontawesome.com, maps.googleapis.com
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 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.

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 — /server-status

Port PATH · Content Discovery

LOW

WHAT IT IS	/server-status returned HTTP 403. This confirms the path exists, but access control prevented access — it was not accessed.
BUSINESS RISK	/server-status 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 54.203.238.30 (hosted by Amazon.com, 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 54.203.238.30 is registered to Amazon.com, Inc. — network: AMAZO-ZPDX6 — block: 54.200.0.0 – 54.203.255.255.

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

robots.txt Present

Port PATH · Content Discovery

INFO

WHAT IT IS /robots.txt is present and was used to seed additional path checks.

BUSINESS RISK /robots.txt is present and was used to seed additional path checks.

HOW TO FIX No action needed — robots.txt is meant to be public.

URGENCY Fix within 1 month

sitemap.xml Present

Port PATH · Content Discovery

INFO

WHAT IT IS /sitemap.xml is present and was used to seed additional path checks.

BUSINESS RISK /sitemap.xml is present and was used to seed additional path checks.

HOW TO FIX No action needed — sitemap.xml is meant to be public.

URGENCY Fix within 1 month

PORT REFERENCE TABLE

PORT	SERVICE	RISK	DESCRIPTION
DNS	SPF Record	HIGH	No SPF record — there's no DNS-level list of which mail servers are authorized to send using this do
DNS	DMARC Record	HIGH	No DMARC record — your domain has zero email authentication enforcement, making it trivial to impersonate

SECRET	Exposed Credential	HIGH	A Google API Key was found exposed in the homepage HTML (AlzaSy...p7Ds) — visible to anyone who views
CORS	CORS Misconfiguration	HIGH	the site reflects any Origin header back in Access-Control-Allow-Origin while also allowing credenti
HTTPS	Environment Secrets Exposed	MEDIUM	The path /.env returns a restricted-access response (401/403) that is genuinely distinct from how th
DNS	DKIM Record	MEDIUM	No DKIM record found under common selector names — this does not confirm DKIM is unconfigured, only
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): assets.restaurants.appleb
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
PATH	Content Discovery	LOW	/.env returned HTTP 403. This confirms the path exists, but access control prevented access — it was
PATH	Content Discovery	LOW	/server-status returned HTTP 403. This confirms the path exists, but access control prevented access
HTTPS	IP Reputation	INFO	Public IP 54.203.238.30 (hosted by Amazon.com, Inc.) has no abuse reports on AbuseIPDB in the last 9
HTTPS	IP WHOIS / Network Ownership	INFO	Public IP 54.203.238.30 is registered to Amazon.com, Inc. — network: AMAZO-ZPDX6 — block: 54.200.0.0
PATH	Content Discovery	INFO	/robots.txt is present and was used to seed additional path checks.
PATH	Content Discovery	INFO	/sitemap.xml is present and was used to seed additional path checks.

NEXT STEPS

→ Start with the 4 HIGH risk finding(s) today — these are your most urgent risks. First, address 'Missing SPF Record — Email Spoofing Possible': Add an SPF TXT record to your DNS listing every service that actually sends mail for this domain — leaving one out will cause its mail to fail SPF and possibly get rejected. Then work through the remaining findings in order of severity.

