

SECURITY REPORT



SCAN TARGET

<https://www.go2kennewick.com>

Open ports detected: 0 · Public IP: 104.16.182.134

A risk score of 10/10 (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 www.go2kennewick.com found 70 issues including 18 high-risk and 5 medium-risk findings. The most critical issue is 'Exposed Google API Key in Page Source' 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 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.
- 2 Confirmed Sensitive-Content Exposure — /currenteventsview.asp: Remove or block public access to /currenteventsview.
- 3 Confirmed Sensitive-Content Exposure — /currenteventsview.aspx: Remove or block public access to /currenteventsview.

DETAILED FINDINGS

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...t784) — 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.

Confirmed Sensitive-Content Exposure — /currenteventsview.asp

Port PATH · Content Discovery

HIGH

WHAT IT IS	/currenteventsview.asp returned HTTP 200 and its content matched a known secret pattern (Environment variable (autoCompleteDebounceKey)). This was not a soft-404 or generic page — real sensitive content was present.
BUSINESS RISK	Whatever secret is exposed here (credentials, keys, or connection strings) can be used immediately by anyone who requests this URL — no further exploitation needed.
REAL-WORLD EXAMPLE	Example: an automated bot that continuously scans for exposed .env/config files finds this one and tests the leaked credentials against the relevant service within minutes.
HOW TO FIX	Remove or block public access to /currenteventsview.asp immediately (web server rule, e.g. Nginx 'location ~ /currenteventsview\..asp { deny all; }'), and rotate every credential that may have been exposed, since it should be treated as compromised.
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.

Confirmed Sensitive-Content Exposure — /currenteventsview.aspx

Port PATH · Content Discovery

HIGH

WHAT IT IS	/currenteventsview.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (autoCompleteDebounceKey)). This was not a soft-404 or generic page — real sensitive content was present.
BUSINESS RISK	Whatever secret is exposed here (credentials, keys, or connection strings) can be used immediately by anyone who requests this URL — no further exploitation needed.
REAL-WORLD EXAMPLE	Example: an automated bot that continuously scans for exposed .env/config files finds this one and tests the leaked credentials against the relevant service within minutes.
HOW TO FIX	Remove or block public access to /currenteventsview.aspx immediately (web server rule, e.g. Nginx 'location ~ /currenteventsview\..aspx { deny all; }'), and rotate every credential that may have been exposed, since it should be treated as compromised.
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.

Confirmed Sensitive-Content Exposure — /CurrentEventsView.aspx

Port PATH · Content Discovery

HIGH

WHAT IT IS	/CurrentEventsView.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (autoCompleteDebounceKey)). This was not a soft-404 or generic page — real sensitive content was present.
BUSINESS RISK	Whatever secret is exposed here (credentials, keys, or connection strings) can be used immediately by anyone who requests this URL — no further exploitation needed.
REAL-WORLD EXAMPLE	Example: an automated bot that continuously scans for exposed .env/config files finds this one and tests the leaked credentials against the relevant service within minutes.
HOW TO FIX	Remove or block public access to /CurrentEventsView.aspx immediately (web server rule, e.g. Nginx 'location ~ /CurrentEventsView\.aspx { deny all; }'), and rotate every credential that may have been exposed, since it should be treated as compromised.
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.

Confirmed Sensitive-Content Exposure — /Search

Port PATH · Content Discovery

HIGH

WHAT IT IS	/Search returned HTTP 200 and its content matched a known secret pattern (Environment variable (autoCompleteDebounceKey)). This was not a soft-404 or generic page — real sensitive content was present.
BUSINESS RISK	Whatever secret is exposed here (credentials, keys, or connection strings) can be used immediately by anyone who requests this URL — no further exploitation needed.
REAL-WORLD EXAMPLE	Example: an automated bot that continuously scans for exposed .env/config files finds this one and tests the leaked credentials against the relevant service within minutes.
HOW TO FIX	Remove or block public access to /Search immediately (web server rule, e.g. Nginx 'location ~ /Search { deny all; }'), and rotate every credential that may have been exposed, since it should be treated as compromised.
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.

Confirmed Sensitive-Content Exposure — /CurrentEventsView.asp

Port PATH · Content Discovery

HIGH

WHAT IT IS	/CurrentEventsView.asp returned HTTP 200 and its content matched a known secret pattern (Environment variable (autoCompleteDebounceKey)). This was not a soft-404 or generic page — real sensitive content was present.
BUSINESS RISK	Whatever secret is exposed here (credentials, keys, or connection strings) can be used immediately by anyone who requests this URL — no further exploitation needed.
REAL-WORLD EXAMPLE	Example: an automated bot that continuously scans for exposed .env/config files finds this one and tests the leaked credentials against the relevant service within minutes.
HOW TO FIX	Remove or block public access to /CurrentEventsView.asp immediately (web server rule, e.g. Nginx 'location ~ /CurrentEventsView\asp { deny all; }'), and rotate every credential that may have been exposed, since it should be treated as compromised.
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.

Confirmed Sensitive-Content Exposure — /error.aspx

Port PATH · Content Discovery

HIGH

WHAT IT IS	/error.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (autoCompleteDebounceKey)). This was not a soft-404 or generic page — real sensitive content was present.
BUSINESS RISK	Whatever secret is exposed here (credentials, keys, or connection strings) can be used immediately by anyone who requests this URL — no further exploitation needed.
REAL-WORLD EXAMPLE	Example: an automated bot that continuously scans for exposed .env/config files finds this one and tests the leaked credentials against the relevant service within minutes.
HOW TO FIX	Remove or block public access to /error.aspx immediately (web server rule, e.g. Nginx 'location ~ /error\.aspx { deny all; }'), and rotate every credential that may have been exposed, since it should be treated as compromised.
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.

Confirmed Sensitive-Content Exposure — /RSS.aspx

Port PATH · Content Discovery

HIGH

WHAT IT IS	/RSS.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (autoCompleteDebounceKey)). This was not a soft-404 or generic page — real sensitive content was present.
BUSINESS RISK	Whatever secret is exposed here (credentials, keys, or connection strings) can be used immediately by anyone who requests this URL — no further exploitation needed.
REAL-WORLD EXAMPLE	Example: an automated bot that continuously scans for exposed .env/config files finds this one and tests the leaked credentials against the relevant service within minutes.
HOW TO FIX	Remove or block public access to /RSS.aspx immediately (web server rule, e.g. Nginx 'location ~ /RSS\.aspx { deny all; }'), and rotate every credential that may have been exposed, since it should be treated as compromised.
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.

Confirmed Sensitive-Content Exposure — /Error.aspx

Port PATH · Content Discovery

HIGH

WHAT IT IS	/Error.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (autoCompleteDebounceKey)). This was not a soft-404 or generic page — real sensitive content was present.
BUSINESS RISK	Whatever secret is exposed here (credentials, keys, or connection strings) can be used immediately by anyone who requests this URL — no further exploitation needed.
REAL-WORLD EXAMPLE	Example: an automated bot that continuously scans for exposed .env/config files finds this one and tests the leaked credentials against the relevant service within minutes.
HOW TO FIX	Remove or block public access to /Error.aspx immediately (web server rule, e.g. Nginx 'location ~ /Error\.aspx { deny all; }'), and rotate every credential that may have been exposed, since it should be treated as compromised.
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.

Confirmed Sensitive-Content Exposure — /rss.aspx

Port PATH · Content Discovery

HIGH

WHAT IT IS	/rss.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (autoCompleteDebounceKey)). This was not a soft-404 or generic page — real sensitive content was present.
BUSINESS RISK	Whatever secret is exposed here (credentials, keys, or connection strings) can be used immediately by anyone who requests this URL — no further exploitation needed.
REAL-WORLD EXAMPLE	Example: an automated bot that continuously scans for exposed .env/config files finds this one and tests the leaked credentials against the relevant service within minutes.
HOW TO FIX	Remove or block public access to /rss.aspx immediately (web server rule, e.g. Nginx 'location ~ /rss\.aspx { deny all; }'), and rotate every credential that may have been exposed, since it should be treated as compromised.
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.

Confirmed Sensitive-Content Exposure — /calendar.aspx

Port PATH · Content Discovery

HIGH

WHAT IT IS	/calendar.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (autoCompleteDebounceKey)). This was not a soft-404 or generic page — real sensitive content was present.
BUSINESS RISK	Whatever secret is exposed here (credentials, keys, or connection strings) can be used immediately by anyone who requests this URL — no further exploitation needed.
REAL-WORLD EXAMPLE	Example: an automated bot that continuously scans for exposed .env/config files finds this one and tests the leaked credentials against the relevant service within minutes.
HOW TO FIX	Remove or block public access to /calendar.aspx immediately (web server rule, e.g. Nginx 'location ~ /calendar\.aspx { deny all; }'), and rotate every credential that may have been exposed, since it should be treated as compromised.
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.

Confirmed Sensitive-Content Exposure — /Bids.aspx

Port PATH · Content Discovery

HIGH

WHAT IT IS	/Bids.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (autoCompleteDebounceKey)). This was not a soft-404 or generic page — real sensitive content was present.
BUSINESS RISK	Whatever secret is exposed here (credentials, keys, or connection strings) can be used immediately by anyone who requests this URL — no further exploitation needed.
REAL-WORLD EXAMPLE	Example: an automated bot that continuously scans for exposed .env/config files finds this one and tests the leaked credentials against the relevant service within minutes.
HOW TO FIX	Remove or block public access to /Bids.aspx immediately (web server rule, e.g. Nginx 'location ~ /Bids\.aspx { deny all; }'), and rotate every credential that may have been exposed, since it should be treated as compromised.
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.

Confirmed Sensitive-Content Exposure — /bids.aspx

Port PATH · Content Discovery

HIGH

WHAT IT IS	/bids.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (autoCompleteDebounceKey)). This was not a soft-404 or generic page — real sensitive content was present.
BUSINESS RISK	Whatever secret is exposed here (credentials, keys, or connection strings) can be used immediately by anyone who requests this URL — no further exploitation needed.
REAL-WORLD EXAMPLE	Example: an automated bot that continuously scans for exposed .env/config files finds this one and tests the leaked credentials against the relevant service within minutes.
HOW TO FIX	Remove or block public access to /bids.aspx immediately (web server rule, e.g. Nginx 'location ~ /bids\.aspx { deny all; }'), and rotate every credential that may have been exposed, since it should be treated as compromised.
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.

Confirmed Sensitive-Content Exposure — /faq.aspx

Port PATH · Content Discovery

HIGH

WHAT IT IS	/faq.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (autoCompleteDebounceKey)). This was not a soft-404 or generic page — real sensitive content was present.
BUSINESS RISK	Whatever secret is exposed here (credentials, keys, or connection strings) can be used immediately by anyone who requests this URL — no further exploitation needed.
REAL-WORLD EXAMPLE	Example: an automated bot that continuously scans for exposed .env/config files finds this one and tests the leaked credentials against the relevant service within minutes.
HOW TO FIX	Remove or block public access to /faq.aspx immediately (web server rule, e.g. Nginx 'location ~ /faq\.aspx { deny all; }'), and rotate every credential that may have been exposed, since it should be treated as compromised.
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.

Confirmed Sensitive-Content Exposure — /CivicAlerts.aspx

Port PATH · Content Discovery

HIGH

WHAT IT IS	/CivicAlerts.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (autoCompleteDebounceKey)). This was not a soft-404 or generic page — real sensitive content was present.
BUSINESS RISK	Whatever secret is exposed here (credentials, keys, or connection strings) can be used immediately by anyone who requests this URL — no further exploitation needed.
REAL-WORLD EXAMPLE	Example: an automated bot that continuously scans for exposed .env/config files finds this one and tests the leaked credentials against the relevant service within minutes.
HOW TO FIX	Remove or block public access to /CivicAlerts.aspx immediately (web server rule, e.g. Nginx 'location ~ /CivicAlerts\.aspx { deny all; }'), and rotate every credential that may have been exposed, since it should be treated as compromised.
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.

Confirmed Sensitive-Content Exposure — /civicalerts.aspx

Port PATH · Content Discovery

HIGH

WHAT IT IS	/civicalerts.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (autoCompleteDebounceKey)). This was not a soft-404 or generic page — real sensitive content was present.
BUSINESS RISK	Whatever secret is exposed here (credentials, keys, or connection strings) can be used immediately by anyone who requests this URL — no further exploitation needed.
REAL-WORLD EXAMPLE	Example: an automated bot that continuously scans for exposed .env/config files finds this one and tests the leaked credentials against the relevant service within minutes.
HOW TO FIX	Remove or block public access to /civicalerts.aspx immediately (web server rule, e.g. Nginx 'location ~ /civicalerts\.aspx { deny all; }'), and rotate every credential that may have been exposed, since it should be treated as compromised.
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.

Confirmed Sensitive-Content Exposure — /FAQ.aspx

Port PATH · Content Discovery

HIGH

WHAT IT IS	/FAQ.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (autoCompleteDebounceKey)). This was not a soft-404 or generic page — real sensitive content was present.
BUSINESS RISK	Whatever secret is exposed here (credentials, keys, or connection strings) can be used immediately by anyone who requests this URL — no further exploitation needed.
REAL-WORLD EXAMPLE	Example: an automated bot that continuously scans for exposed .env/config files finds this one and tests the leaked credentials against the relevant service within minutes.
HOW TO FIX	Remove or block public access to /FAQ.aspx immediately (web server rule, e.g. Nginx 'location ~ /FAQ\.aspx { deny all; }'), and rotate every credential that may have been exposed, since it should be treated as compromised.
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.

Confirmed Sensitive-Content Exposure — /Calendar.aspx

Port PATH · Content Discovery

HIGH

WHAT IT IS	/Calendar.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (autoCompleteDebounceKey)). This was not a soft-404 or generic page — real sensitive content was present.
BUSINESS RISK	Whatever secret is exposed here (credentials, keys, or connection strings) can be used immediately by anyone who requests this URL — no further exploitation needed.
REAL-WORLD EXAMPLE	Example: an automated bot that continuously scans for exposed .env/config files finds this one and tests the leaked credentials against the relevant service within minutes.
HOW TO FIX	Remove or block public access to /Calendar.aspx immediately (web server rule, e.g. Nginx 'location ~ /Calendar\.aspx { deny all; }'), and rotate every credential that may have been exposed, since it should be treated as compromised.
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.

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.

Environment Secrets Exposed (Unconfirmed)

Port HTTPS · Environment
Secrets Exposed

MEDIUM

WHAT IT IS	The path /.env 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 credentials or sensitive configuration that should never be public
BUSINESS RISK	Exposed credentials in /.env (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 /.env 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 /.env via web server rules (Nginx: 'location ~ /\.env { deny all; }'). 3. Never commit secrets to version control — use .gitignore 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.

Exposed .git Directory (Unconfirmed)

Port HTTPS · Exposed .git
Directory

MEDIUM

WHAT IT IS	The path <code>/.git/HEAD</code> 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 <code>/.git/HEAD</code> 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 <code>.git</code> 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 <code>/.git/HEAD</code> entirely — it should never be servable. Nginx: <code>'location ~ /\.git { deny all; }'</code> . Apache: <code>'<DirectoryMatch "\.git"> Require all denied </DirectoryMatch>'</code> . Better yet, don't deploy the <code>.git</code> 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

CSP script-src Too Permissive

Port HTTPS · Content Security Policy

MEDIUM

WHAT IT IS	CSP script-src allows 'unsafe-inline', 'unsafe-eval', * — an injected or hijacked script can still run freely
BUSINESS RISK	This setting defeats most of the protection CSP is meant to provide. If a third-party script you rely on is ever compromised — the exact mechanism behind most 'watering hole' attacks — it will execute without restriction and with no warning.
REAL-WORLD EXAMPLE	Example: A vendor's analytics or chat-widget script gets compromised at the source; because the policy still allows it (or allows inline/eval scripts generally), the malicious code runs exactly as if it belonged on the site, with no warning to anyone.
HOW TO FIX	Tighten script-src to an explicit allowlist of trusted domains and drop 'unsafe-inline'/'unsafe-eval'/wildcards. Move inline scripts to external files or use a nonce/hash. Test changes with https://csp-evaluator.withgoogle.com before deploying.
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): app-script.monsido.com , docaccess.com , www.googletagmanager.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.

Insecure Cookie Flags (CP_IsMobile)

Port HTTPS · Cookie Security

LOW

WHAT IT IS	Cookie 'CP_IsMobile' is missing the HttpOnly, SameSite flag(s). Observed attributes: path=/, secure.
BUSINESS RISK	This cookie's name doesn't clearly indicate it holds session or authentication data, so the practical impact depends on what value it actually stores — anywhere from low-stakes (a UI preference) to more sensitive (tracking or personalization data). Missing these flags means whatever the cookie does hold is more exposed than it needs to be to interception or script access.
REAL-WORLD EXAMPLE	Example: whatever value 'CP_IsMobile' holds could be read by an injected script (no HttpOnly) or intercepted on an unencrypted connection (no Secure) — the actual severity depends on how sensitive that value turns out to be.
HOW TO FIX	Add the missing flag(s) when setting this cookie: Secure (only send over HTTPS), HttpOnly (block JavaScript access), SameSite=Lax or Strict (limit cross-site sending). Most frameworks expose this as a one-line config option — e.g. Flask: <code>app.config['SESSION_COOKIE_SECURE']=True, SESSION_COOKIE_HTTPONLY=True, SESSION_COOKIE_SAMESITE='Lax'.</code>
URGENCY	Fix within 1 month
REFERENCE	CWE-1004, CWE-1275 — Sensitive Cookie Without 'HttpOnly' Flag; Sensitive Cookie with Improper SameSite Attribute The cookie isn't marked HttpOnly, so a malicious script on the page (such as one injected via XSS) can read and steal it. The cookie's SameSite attribute isn't set to a safe value, making it easier for other sites to trick a visitor's browser into sending it in cross-site requests.

Insecure Cookie Flags (CP_IsMobile)

Port HTTPS · Cookie Security

LOW

WHAT IT IS	Cookie 'CP_IsMobile' is missing the HttpOnly, SameSite flag(s). Observed attributes: path=/, secure.
BUSINESS RISK	This cookie's name doesn't clearly indicate it holds session or authentication data, so the practical impact depends on what value it actually stores — anywhere from low-stakes (a UI preference) to more sensitive (tracking or personalization data). Missing these flags means whatever the cookie does hold is more exposed than it needs to be to interception or script access.
REAL-WORLD EXAMPLE	Example: whatever value 'CP_IsMobile' holds could be read by an injected script (no HttpOnly) or intercepted on an unencrypted connection (no Secure) — the actual severity depends on how sensitive that value turns out to be.
HOW TO FIX	Add the missing flag(s) when setting this cookie: Secure (only send over HTTPS), HttpOnly (block JavaScript access), SameSite=Lax or Strict (limit cross-site sending). Most frameworks expose this as a one-line config option — e.g. Flask: <code>app.config['SESSION_COOKIE_SECURE']=True, SESSION_COOKIE_HTTPONLY=True, SESSION_COOKIE_SAMESITE='Lax'.</code>
URGENCY	Fix within 1 month
REFERENCE	CWE-1004, CWE-1275 — Sensitive Cookie Without 'HttpOnly' Flag; Sensitive Cookie with Improper SameSite Attribute The cookie isn't marked HttpOnly, so a malicious script on the page (such as one injected via XSS) can read and steal it. The cookie's SameSite attribute isn't set to a safe value, making it easier for other sites to trick a visitor's browser into sending it in cross-site requests.

SSH Private Key Exposed (Unconfirmed)

Port HTTPS · SSH Private Key
Exposed

LOW

WHAT IT IS	The path <code>/.id_rsa</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>/.id_rsa</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>/.id_rsa</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>/.id_rsa</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 month
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.

Missing SPF Record — Email Spoofing Possible

Port DNS · SPF Record

LOW

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 (note: this domain publishes no MX records, so it may not send email at all — adding SPF/DMARC here is preventative, to stop spoofing of a non-sending domain, and is low priority)
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 within 1 month
REFERENCE	CWE-290 — Authentication Bypass by Spoofing A system trusts information that can be forged, so an attacker can impersonate a trusted party — for email, missing SPF/DMARC/DKIM lets a sender fake your domain.

Missing DMARC Record — Email Impersonation Risk

Port DNS · DMARC Record

LOW

WHAT IT IS	No DMARC record — your domain has no DMARC email-authentication policy, which makes it easier to impersonate your business in email (note: this domain publishes no MX records, so it may not send email at all — this is preventative hardening and low priority)
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 within 1 month
REFERENCE	CWE-290 — Authentication Bypass by Spoofing A system trusts information that can be forged, so an attacker can impersonate a trusted party — for email, missing SPF/DMARC/DKIM lets a sender fake your domain.

Protected Path — /.env

Port PATH · Content Discovery

LOW

WHAT IT IS	/.env returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't 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 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't 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 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't 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 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't 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 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't 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 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /id_rsa

Port PATH · Content Discovery

LOW

WHAT IT IS	/id_rsa returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /phpinfo.php

Port PATH · Content Discovery

LOW

WHAT IT IS	/phpinfo.php returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /web.config

Port PATH · Content Discovery

LOW

WHAT IT IS	/web.config returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /web.config.bak

Port PATH · Content Discovery

LOW

WHAT IT IS	/web.config.bak returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /.idea/workspace.xml

Port PATH · Content Discovery

LOW

WHAT IT IS	/.idea/workspace.xml returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /backup.bak

Port PATH · Content Discovery

LOW

WHAT IT IS	/backup.bak returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /error_log

Port PATH · Content Discovery

LOW

WHAT IT IS	/error_log returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /.DS_Store

Port PATH · Content Discovery

LOW

WHAT IT IS	/.DS_Store returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /backup.old

Port PATH · Content Discovery

LOW

WHAT IT IS	/backup.old returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /site.bak

Port PATH · Content Discovery

LOW

WHAT IT IS	/site.bak returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /site.old

Port PATH · Content Discovery

LOW

WHAT IT IS	/site.old returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /www.bak

Port PATH · Content Discovery

LOW

WHAT IT IS	/www.bak returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /database.bak

Port PATH · Content Discovery

LOW

WHAT IT IS	/database.bak returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /db.bak

Port PATH · Content Discovery

LOW

WHAT IT IS	/db.bak returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /www.old

Port PATH · Content Discovery

LOW

WHAT IT IS	/www.old returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /database.old

Port PATH · Content Discovery

LOW

WHAT IT IS	/database.old returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /db.old

Port PATH · Content Discovery

LOW

WHAT IT IS	/db.old returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /config.old

Port PATH · Content Discovery

LOW

WHAT IT IS	/config.old returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /app.bak

Port PATH · Content Discovery

LOW

WHAT IT IS	/app.bak returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /config.bak

Port PATH · Content Discovery

LOW

WHAT IT IS	/config.bak returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /app.old

Port PATH · Content Discovery

LOW

WHAT IT IS	/app.old returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /old.bak

Port PATH · Content Discovery

LOW

WHAT IT IS	/old.bak returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /old.old

Port PATH · Content Discovery

LOW

WHAT IT IS	/old.old returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /common/admin/

Port PATH · Content Discovery

LOW

WHAT IT IS	/common/admin/ returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /OJA

Port PATH · Content Discovery

LOW

WHAT IT IS	/OJA returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /Support

Port PATH · Content Discovery

LOW

WHAT IT IS	/Support returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /support

Port PATH · Content Discovery

LOW

WHAT IT IS	/support returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /common/modules/debug/

Port PATH · Content Discovery

LOW

WHAT IT IS	/common/modules/debug/ returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't left at a default.
URGENCY	Fix within 1 month
REFERENCE	CWE-284

Protected Path — /common/modules/icalendar/

Port PATH · Content Discovery

LOW

WHAT IT IS	/common/modules/icalendar/ returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 like this is often a generic WAF / default-deny rule rather than proof a real file exists at this path — verify manually before treating it as a real resource.
HOW TO FIX	No action needed if this is intentionally restricted or is just a WAF default-deny. If it maps to a real protected resource, confirm the protection is deliberate and any credentials aren't 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 104.16.182.134 (hosted by Cloudflare, 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 104.16.182.134 is registered to Cloudflare, Inc. — network: CLOUDFLARENET — block: 104.16.0.0 – 104.31.255.255.
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

DKIM Not Detected Under Common Selectors

Port DNS · DKIM Record

INFO

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 month
REFERENCE	CWE-290 — Authentication Bypass by Spoofing A system trusts information that can be forged, so an attacker can impersonate a trusted party — for email, missing SPF/DMARC/DKIM lets a sender fake your domain.

Detected Technology Stack

Port HTTPS · Technology
Fingerprint

INFO

WHAT IT IS	Detected technology stack: JQuery 3.7.1, JQuery-Migrate 3.6.0.
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

robots.txt Present

Port PATH · Content Discovery

INFO

WHAT IT IS /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.

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

URGENCY Fix within 1 month

3 Additional Paths Responded (site map)

Port PATH · Content Discovery

INFO

WHAT IT IS These same-origin paths returned a real response rather than the site's 404 page. They are usually ordinary application routes and are grouped here for a quick manual skim instead of as individual findings — a path merely existing is not a vulnerability.

HOW TO FIX Skim the list and confirm none are leftover debug/test/admin tooling that should be removed or access-restricted. Routine public routes need no action.

URGENCY Fix within 1 month

PORT REFERENCE TABLE

PORT	SERVICE	RISK	DESCRIPTION
SECRET	Exposed Credential	HIGH	A Google API Key was found exposed in the homepage HTML (AlzaSy...t784) — visible to anyone who views
PATH	Content Discovery	HIGH	/currenteventsview.asp returned HTTP 200 and its content matched a known secret pattern (Environment
PATH	Content Discovery	HIGH	/currenteventsview.aspx returned HTTP 200 and its content matched a known secret pattern (Environmen
PATH	Content Discovery	HIGH	/CurrentEventsView.aspx returned HTTP 200 and its content matched a known secret pattern (Environmen
PATH	Content Discovery	HIGH	/Search returned HTTP 200 and its content matched a known secret pattern (Environment variable (auto

PATH	Content Discovery	HIGH	/CurrentEventsView.asp returned HTTP 200 and its content matched a known secret pattern (Environment
PATH	Content Discovery	HIGH	/error.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (
PATH	Content Discovery	HIGH	/RSS.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (au
PATH	Content Discovery	HIGH	/Error.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (
PATH	Content Discovery	HIGH	/rss.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (au
PATH	Content Discovery	HIGH	/calendar.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variabl
PATH	Content Discovery	HIGH	/Bids.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (a
PATH	Content Discovery	HIGH	/bids.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (a
PATH	Content Discovery	HIGH	/faq.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (au
PATH	Content Discovery	HIGH	/CivicAlerts.aspx returned HTTP 200 and its content matched a known secret pattern (Environment vari
PATH	Content Discovery	HIGH	/civicalerts.aspx returned HTTP 200 and its content matched a known secret pattern (Environment vari
PATH	Content Discovery	HIGH	/FAQ.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variable (au
PATH	Content Discovery	HIGH	/Calendar.aspx returned HTTP 200 and its content matched a known secret pattern (Environment variabl
HTTPS	HSTS Header	MEDIUM	Missing Strict-Transport-Security (HSTS) — browsers aren't forced to always use HTTPS, leaving visit
HTTPS	Environment Secrets Exposed	MEDIUM	The path /.env returns a restricted-access response (401/403) that is genuinely distinct from how th
HTTPS	Exposed .git Directory	MEDIUM	The path /.git/HEAD returns a restricted-access response (401/403) that is genuinely distinct from h
HTTPS	Content Security Policy	MEDIUM	CSP script-src allows 'unsafe-inline', 'unsafe-eval', * — an injected or hijacked script can still r
HTTPS	Third-Party Script Loading	MEDIUM	3 third-party script source(s) loaded without Subresource Integrity (SRI): app-script.monsido.com, d

HTTPS	Cookie Security	LOW	Cookie 'CP_IsMobile' is missing the HttpOnly, SameSite flag(s). Observed attributes: path=/, secure.
HTTPS	Cookie Security	LOW	Cookie 'CP_IsMobile' is missing the HttpOnly, SameSite flag(s). Observed attributes: path=/, secure.
HTTPS	SSH Private Key Exposed	LOW	The path /.id_rsa returns a restricted-access response (401/403) that is genuinely distinct from how
DNS	SPF Record	LOW	No SPF record — there's no DNS-level list of which mail servers are authorized to send using this do
DNS	DMARC Record	LOW	No DMARC record — your domain has no DMARC email-authentication policy, which makes it easier to imp
PATH	Content Discovery	LOW	/.env returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 I
PATH	Content Discovery	LOW	/.git/HEAD returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a
PATH	Content Discovery	LOW	/wp-config.php.bak returned HTTP 403 — access control blocked the request and nothing was accessed.
PATH	Content Discovery	LOW	/.git/config returned HTTP 403 — access control blocked the request and nothing was accessed. Note:
PATH	Content Discovery	LOW	/.svn/entries returned HTTP 403 — access control blocked the request and nothing was accessed. Note:
PATH	Content Discovery	LOW	/.htpasswd returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a
PATH	Content Discovery	LOW	/.id_rsa returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403
PATH	Content Discovery	LOW	/phpinfo.php returned HTTP 403 — access control blocked the request and nothing was accessed. Note:
PATH	Content Discovery	LOW	/web.config returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a
PATH	Content Discovery	LOW	/web.config.bak returned HTTP 403 — access control blocked the request and nothing was accessed. Not
PATH	Content Discovery	LOW	/.idea/workspace.xml returned HTTP 403 — access control blocked the request and nothing was accessed
PATH	Content Discovery	LOW	/backup.bak returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a
PATH	Content Discovery	LOW	/error_log returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a

PATH	Content Discovery	LOW	/DS_Store returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a
PATH	Content Discovery	LOW	/backup.old returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a
PATH	Content Discovery	LOW	/site.bak returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 4
PATH	Content Discovery	LOW	/site.old returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 4
PATH	Content Discovery	LOW	/www.bak returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 40
PATH	Content Discovery	LOW	/database.bak returned HTTP 403 — access control blocked the request and nothing was accessed. Note:
PATH	Content Discovery	LOW	/db.bak returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403
PATH	Content Discovery	LOW	/www.old returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 40
PATH	Content Discovery	LOW	/database.old returned HTTP 403 — access control blocked the request and nothing was accessed. Note:
PATH	Content Discovery	LOW	/db.old returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403
PATH	Content Discovery	LOW	/config.old returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a
PATH	Content Discovery	LOW	/app.bak returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 40
PATH	Content Discovery	LOW	/config.bak returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a
PATH	Content Discovery	LOW	/app.old returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 40
PATH	Content Discovery	LOW	/old.bak returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 40
PATH	Content Discovery	LOW	/old.old returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 40
PATH	Content Discovery	LOW	/common/admin/ returned HTTP 403 — access control blocked the request and nothing was accessed. Note
PATH	Content Discovery	LOW	/OJA returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 403 li

PATH	Content Discovery	LOW	/Support returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 40
PATH	Content Discovery	LOW	/support returned HTTP 403 — access control blocked the request and nothing was accessed. Note: a 40
PATH	Content Discovery	LOW	/common/modules/debug/ returned HTTP 403 — access control blocked the request and nothing was access
PATH	Content Discovery	LOW	/common/modules/icalendar/ returned HTTP 403 — access control blocked the request and nothing was ac
HTTPS	IP Reputation	INFO	Public IP 104.16.182.134 (hosted by Cloudflare, Inc.) has no abuse reports on AbuseIPDB in the last
HTTPS	IP WHOIS / Network Ownership	INFO	Public IP 104.16.182.134 is registered to Cloudflare, Inc. — network: CLOUDFLARENET — block: 104.16.
DNS	DKIM Record	INFO	No DKIM record found under common selector names — this does not confirm DKIM is unconfigured, only
HTTPS	Technology Fingerprint	INFO	Detected technology stack: JQuery 3.7.1, JQuery-Migrate 3.6.0.
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.
PATH	Content Discovery	INFO	These same-origin paths returned a real response rather than the site's 404 page. They are usually o

NEXT STEPS

→ Start with the 18 HIGH risk finding(s) today — these are your most urgent risks. First, address '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. Then work through the remaining findings in order of severity.

This report is for informational purposes only and represents a point-in-time automated scan. It is not a substitute for a professional penetration test.