

SECURITY REPORT



SCAN TARGET

target.com

Open ports detected: 0 · Public IP: **151.101.194.187**

A risk score of 4/10 (MEDIUM) indicates your business has security issues that need attention.

The higher the score, the greater the chance of a breach, ransomware, or data theft.

EXECUTIVE SUMMARY

The security scan of target.com found 29 medium and low severity issues. No critical vulnerabilities were detected, but 8 issues should be addressed this week to harden your security posture. Detailed fix instructions are included for each finding.

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

TOP RECOMMENDATIONS

- 1 Insecure Cookie Flags (idToken): 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).
- 2 Insecure Cookie Flags (TealeafAkaSid): 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).
- 3 Weak DKIM Key — 1024-bit RSA (selector: k1): Upgrade the DKIM key for selector 'k1' to 2048 bits.

DETAILED FINDINGS

Insecure Cookie Flags (idToken)

Port HTTPS · Cookie Security

MEDIUM

WHAT IT IS	Cookie 'idToken' is missing the HttpOnly flag(s). Observed attributes: Domain=target.com, Path=/, SameSite=Lax, Secure.
BUSINESS RISK	This cookie's name suggests it may hold a session or authentication token, though its actual contents weren't inspected. If it does, a cookie missing these flags is easier to steal through cross-site scripting or to intercept over an unencrypted connection — and a stolen session cookie can let an attacker impersonate that logged-in user without ever needing their password.
REAL-WORLD EXAMPLE	Example: A visitor on public wifi has their session cookie intercepted because it wasn't marked Secure, or a malicious ad script reads it directly because it wasn't marked HttpOnly — either way, the attacker is now logged in as that user without ever seeing their password.
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 week
REFERENCE	CWE-1004 — Sensitive Cookie Without 'HttpOnly' Flag The cookie isn't marked HttpOnly, so a malicious script on the page (such as one injected via XSS) can read and steal it.

WHAT IT IS	Cookie 'TealeafAkaSid' is missing the Secure, HttpOnly, SameSite flag(s). Observed attributes: Expires=Thu, 15 Oct 2026 05:55:02 GMT, Path=/, Domain=target.com.
BUSINESS RISK	This cookie's name suggests it may hold a session or authentication token, though its actual contents weren't inspected. If it does, a cookie missing these flags is easier to steal through cross-site scripting or to intercept over an unencrypted connection — and a stolen session cookie can let an attacker impersonate that logged-in user without ever needing their password.
REAL-WORLD EXAMPLE	Example: A visitor on public wifi has their session cookie intercepted because it wasn't marked Secure, or a malicious ad script reads it directly because it wasn't marked HttpOnly — either way, the attacker is now logged in as that user without ever seeing their password.
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 week
REFERENCE	CWE-614, CWE-1004, CWE-1275 — Sensitive Cookie in HTTPS Session Without 'Secure' Attribute; Sensitive Cookie Without 'HttpOnly' Flag; Sensitive Cookie with Improper SameSite Attribute The cookie isn't marked Secure, so a browser could send it over an unencrypted connection where it can be intercepted. 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.

Weak DKIM Key — 1024-bit RSA (selector: k1)

Port DNS · DKIM Key Strength

MEDIUM

WHAT IT IS	DKIM key for selector 'k1' appears to be approximately 1024 bits — 1024-bit RSA keys are considered weak by modern standards and NIST has deprecated them. A well-resourced attacker could factor this key, breaking your email authentication.
BUSINESS RISK	A forged DKIM signature lets an attacker send phishing or fraud emails that cryptographically appear to come from your domain — bypassing email filters that rely on DKIM as a trust signal, and making impersonation emails indistinguishable from your real ones.
REAL-WORLD EXAMPLE	Example: A security researcher demonstrated in a published study that 512-bit DKIM keys could be factored in under 72 hours using cloud computing resources costing less than \$100 — allowing anyone with that capability to forge valid email signatures for the affected domain.
HOW TO FIX	Upgrade the DKIM key for selector 'k1' to 2048 bits. Generate a new key through your email provider, update the DNS TXT record to the new public key, and retire the old 1024-bit key.
URGENCY	Fix within 1 week
REFERENCE	CWE-326

Weak DKIM Key — 1024-bit RSA (selector: selector1)

Port DNS · DKIM Key Strength

MEDIUM

WHAT IT IS

DKIM key for selector 'selector1' appears to be approximately 1024 bits — 1024-bit RSA keys are considered weak by modern standards and NIST has deprecated them. A well-resourced attacker could factor this key, breaking your email authentication.

BUSINESS RISK

A forged DKIM signature lets an attacker send phishing or fraud emails that cryptographically appear to come from your domain — bypassing email filters that rely on DKIM as a trust signal, and making impersonation emails indistinguishable from your real ones.

REAL-WORLD EXAMPLE

Example: A security researcher demonstrated in a published study that 512-bit DKIM keys could be factored in under 72 hours using cloud computing resources costing less than \$100 — allowing anyone with that capability to forge valid email signatures for the affected domain.

HOW TO FIX

Upgrade the DKIM key for selector 'selector1' to 2048 bits. Generate a new key through your email provider, update the DNS TXT record to the new public key, and retire the old 1024-bit key.

URGENCY

Fix within 1 week

REFERENCE

CWE-326

Weak DKIM Key — 1024-bit RSA (selector: s2)

Port DNS · DKIM Key Strength

MEDIUM

WHAT IT IS	DKIM key for selector 's2' appears to be approximately 1024 bits — 1024-bit RSA keys are considered weak by modern standards and NIST has deprecated them. A well-resourced attacker could factor this key, breaking your email authentication.
BUSINESS RISK	A forged DKIM signature lets an attacker send phishing or fraud emails that cryptographically appear to come from your domain — bypassing email filters that rely on DKIM as a trust signal, and making impersonation emails indistinguishable from your real ones.
REAL-WORLD EXAMPLE	Example: A security researcher demonstrated in a published study that 512-bit DKIM keys could be factored in under 72 hours using cloud computing resources costing less than \$100 — allowing anyone with that capability to forge valid email signatures for the affected domain.
HOW TO FIX	Upgrade the DKIM key for selector 's2' to 2048 bits. Generate a new key through your email provider, update the DNS TXT record to the new public key, and retire the old 1024-bit key.
URGENCY	Fix within 1 week
REFERENCE	CWE-326

Weak DKIM Key — 1024-bit RSA (selector: mandrill)

Port DNS · DKIM Key Strength

MEDIUM

WHAT IT IS	DKIM key for selector 'mandrill' appears to be approximately 1024 bits — 1024-bit RSA keys are considered weak by modern standards and NIST has deprecated them. A well-resourced attacker could factor this key, breaking your email authentication.
BUSINESS RISK	A forged DKIM signature lets an attacker send phishing or fraud emails that cryptographically appear to come from your domain — bypassing email filters that rely on DKIM as a trust signal, and making impersonation emails indistinguishable from your real ones.
REAL-WORLD EXAMPLE	Example: A security researcher demonstrated in a published study that 512-bit DKIM keys could be factored in under 72 hours using cloud computing resources costing less than \$100 — allowing anyone with that capability to forge valid email signatures for the affected domain.
HOW TO FIX	Upgrade the DKIM key for selector 'mandrill' to 2048 bits. Generate a new key through your email provider, update the DNS TXT record to the new public key, and retire the old 1024-bit key.
URGENCY	Fix within 1 week
REFERENCE	CWE-326

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	2 third-party script source(s) loaded without Subresource Integrity (SRI): assets.targetimg1.com, client.px-cloud.net
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 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.

Insecure Cookie Flags (adScriptData)

Port HTTPS · Cookie Security

LOW

WHAT IT IS	Cookie 'adScriptData' is missing the HttpOnly flag(s). Observed attributes: Domain=target.com, Path=/, SameSite=Lax, Secure, Max-Age=86400.
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 'adScriptData' 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 — Sensitive Cookie Without 'HttpOnly' Flag The cookie isn't marked HttpOnly, so a malicious script on the page (such as one injected via XSS) can read and steal it.

Insecure Cookie Flags (sapphire)

Port HTTPS · Cookie Security

LOW

WHAT IT IS	Cookie 'sapphire' is missing the HttpOnly flag(s). Observed attributes: Max-Age=2629746, Expires=Thu, 15 Oct 2026 16:24:07 GMT, Path=/, Domain=.target.com, Secure, SameSite=None.
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 'sapphire' 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 — Sensitive Cookie Without 'HttpOnly' Flag The cookie isn't marked HttpOnly, so a malicious script on the page (such as one injected via XSS) can read and steal it.

Insecure Cookie Flags (onboardingGuest)

Port HTTPS · Cookie Security

LOW

WHAT IT IS	Cookie 'onboardingGuest' is missing the HttpOnly flag(s). Observed attributes: Domain=.target.com, Path=/, Secure, SameSite=Lax.
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 'onboardingGuest' 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 — Sensitive Cookie Without 'HttpOnly' Flag The cookie isn't marked HttpOnly, so a malicious script on the page (such as one injected via XSS) can read and steal it.

Insecure Cookie Flags (fiatsCookie)

Port HTTPS · Cookie Security

LOW

WHAT IT IS	Cookie 'fiatsCookie' is missing the HttpOnly flag(s). Observed attributes: Domain=.target.com, Path=/, Secure, SameSite=Lax.
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 'fiatsCookie' 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 — Sensitive Cookie Without 'HttpOnly' Flag The cookie isn't marked HttpOnly, so a malicious script on the page (such as one injected via XSS) can read and steal it.

Insecure Cookie Flags (sddStore)

Port HTTPS · Cookie Security

LOW

WHAT IT IS	Cookie 'sddStore' is missing the Secure, HttpOnly, SameSite flag(s). Observed attributes: Domain=.target.com, Path=/.
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 'sddStore' 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-614, CWE-1004, CWE-1275 — Sensitive Cookie in HTTPS Session Without 'Secure' Attribute; Sensitive Cookie Without 'HttpOnly' Flag; Sensitive Cookie with Improper SameSite Attribute The cookie isn't marked Secure, so a browser could send it over an unencrypted connection where it can be intercepted. 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 (GuestLocation)

Port HTTPS · Cookie Security

LOW

WHAT IT IS	Cookie 'GuestLocation' is missing the HttpOnly, SameSite flag(s). Observed attributes: Path=/, Secure, Max-Age=86400.
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 'GuestLocation' 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 (visitorId)

Port HTTPS · Cookie Security

LOW

WHAT IT IS	Cookie 'visitorId' is missing the HttpOnly flag(s). Observed attributes: Max-Age=63072000, Expires=Thu, 14 Sep 2028 05:55:01 GMT, Path=/, Domain=.target.com, Secure, SameSite=None.
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 'visitorId' 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 — Sensitive Cookie Without 'HttpOnly' Flag The cookie isn't marked HttpOnly, so a malicious script on the page (such as one injected via XSS) can read and steal it.

Insecure Cookie Flags (_pxhd)

Port HTTPS · Cookie Security

LOW

WHAT IT IS	Cookie '_pxhd' is missing the Secure, HttpOnly flag(s). Observed attributes: Max-Age=31536000, SameSite=Lax, Path=/, Domain=.target.com.
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 '_pxhd' 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-614, CWE-1004 — Sensitive Cookie in HTTPS Session Without 'Secure' Attribute; Sensitive Cookie Without 'HttpOnly' Flag The cookie isn't marked Secure, so a browser could send it over an unencrypted connection where it can be intercepted. The cookie isn't marked HttpOnly, so a malicious script on the page (such as one injected via XSS) can read and steal it.

Login/Admin Path Discovered — /login/

Port PATH · Content Discovery

LOW

WHAT IT IS	/login/ returned HTTP 200 with no authentication challenge. This confirms a login or admin entry point exists at this path.
BUSINESS RISK	/login/ returned HTTP 200 with no authentication challenge. This confirms a login or admin entry point exists at this path.
HOW TO FIX	Confirm this login is protected by a strong password and, ideally, multi-factor authentication. Consider IP-restricting it if it's only used by internal staff.
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 151.101.194.187 (hosted by Fastly, 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 151.101.194.187 is registered to Fastly, Inc. — network: SKYCA-3 — block: 151.101.0.0 – 151.101.255.255.
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

BIMI Opportunity — Verified Logo Not Enabled

Port DNS · BIMI Not Configured

INFO

WHAT IT IS	DMARC is already enforcing, so this domain qualifies for BIMI — a verified logo shown next to emails in Gmail, Yahoo, and Apple Mail — but no BIMI record is set up
BUSINESS RISK	This isn't a vulnerability — it's a missed brand-trust opportunity. Since DMARC is already enforcing (a prerequisite most mailbox providers require), this domain could show a verified logo next to its emails in supporting inboxes, which can measurably improve open rates and make phishing look more out of place by comparison.
REAL-WORLD EXAMPLE	Example: two otherwise-identical marketing emails land in a Gmail inbox — one from a sender with BIMI configured shows a verified brand logo next to it, the other shows a generic default icon indistinguishable from a spoofed sender.
HOW TO FIX	Add a TXT record for 'default._bimi' with a value pointing to a hosted SVG logo, e.g. 'v=BIMI1; l=https://yourdomain.com/logo.svg'. Some mailbox providers (notably Gmail) also require a Verified Mark Certificate (VMC) — a paid trademark-backed certificate — before they'll display the logo; Apple Mail and Yahoo currently do not require one. This is optional and lower-priority than any HIGH/MEDIUM finding above.
URGENCY	Fix within 1 month

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

Path Discovered — /bp/c/

Port PATH · Content Discovery

INFO

WHAT IT IS	/bp/c/ returned HTTP 200, a real distinct response rather than the site's soft-404 page. This path isn't one a typical visitor would stumble onto by browsing the site normally, which is why it's worth a quick manual look — but its existence alone isn't a vulnerability.
BUSINESS RISK	/bp/c/ returned HTTP 200, a real distinct response rather than the site's soft-404 page. This path isn't one a typical visitor would stumble onto by browsing the site normally, which is why it's worth a quick manual look — but its existence alone isn't a vulnerability.
HOW TO FIX	Open the path yourself and confirm it's meant to be public. If it's leftover debug/test/admin tooling that should have been removed, take it down or restrict it by IP allowlist. If it's intentional, no action needed.
URGENCY	Fix within 1 month

Path Discovered — /bp/p/

Port PATH · Content Discovery

INFO

WHAT IT IS	/bp/p/ returned HTTP 200, a real distinct response rather than the site's soft-404 page. This path isn't one a typical visitor would stumble onto by browsing the site normally, which is why it's worth a quick manual look — but its existence alone isn't a vulnerability.
BUSINESS RISK	/bp/p/ returned HTTP 200, a real distinct response rather than the site's soft-404 page. This path isn't one a typical visitor would stumble onto by browsing the site normally, which is why it's worth a quick manual look — but its existence alone isn't a vulnerability.
HOW TO FIX	Open the path yourself and confirm it's meant to be public. If it's leftover debug/test/admin tooling that should have been removed, take it down or restrict it by IP allowlist. If it's intentional, no action needed.
URGENCY	Fix within 1 month

Path Discovered — /deals/

Port PATH · Content Discovery

INFO

WHAT IT IS	/deals/ returned HTTP 200, a real distinct response rather than the site's soft-404 page. This path isn't one a typical visitor would stumble onto by browsing the site normally, which is why it's worth a quick manual look — but its existence alone isn't a vulnerability.
BUSINESS RISK	/deals/ returned HTTP 200, a real distinct response rather than the site's soft-404 page. This path isn't one a typical visitor would stumble onto by browsing the site normally, which is why it's worth a quick manual look — but its existence alone isn't a vulnerability.
HOW TO FIX	Open the path yourself and confirm it's meant to be public. If it's leftover debug/test/admin tooling that should have been removed, take it down or restrict it by IP allowlist. If it's intentional, no action needed.
URGENCY	Fix within 1 month

Path Discovered — /gc?k

Port PATH · Content Discovery

INFO

WHAT IT IS	/gc?k returned HTTP 200, a real distinct response rather than the site's soft-404 page. This path isn't one a typical visitor would stumble onto by browsing the site normally, which is why it's worth a quick manual look — but its existence alone isn't a vulnerability.
BUSINESS RISK	/gc?k returned HTTP 200, a real distinct response rather than the site's soft-404 page. This path isn't one a typical visitor would stumble onto by browsing the site normally, which is why it's worth a quick manual look — but its existence alone isn't a vulnerability.
HOW TO FIX	Open the path yourself and confirm it's meant to be public. If it's leftover debug/test/admin tooling that should have been removed, take it down or restrict it by IP allowlist. If it's intentional, no action needed.
URGENCY	Fix within 1 month

Path Discovered — /cart

Port PATH · Content Discovery

INFO

WHAT IT IS	/cart returned HTTP 200, a real distinct response rather than the site's soft-404 page. This path isn't one a typical visitor would stumble onto by browsing the site normally, which is why it's worth a quick manual look — but its existence alone isn't a vulnerability.
BUSINESS RISK	/cart returned HTTP 200, a real distinct response rather than the site's soft-404 page. This path isn't one a typical visitor would stumble onto by browsing the site normally, which is why it's worth a quick manual look — but its existence alone isn't a vulnerability.
HOW TO FIX	Open the path yourself and confirm it's meant to be public. If it's leftover debug/test/admin tooling that should have been removed, take it down or restrict it by IP allowlist. If it's intentional, no action needed.
URGENCY	Fix within 1 month

Path Discovered — /yr

Port PATH · Content Discovery

INFO

WHAT IT IS	/yr returned HTTP 200, a real distinct response rather than the site's soft-404 page. This path isn't one a typical visitor would stumble onto by browsing the site normally, which is why it's worth a quick manual look — but its existence alone isn't a vulnerability.
BUSINESS RISK	/yr returned HTTP 200, a real distinct response rather than the site's soft-404 page. This path isn't one a typical visitor would stumble onto by browsing the site normally, which is why it's worth a quick manual look — but its existence alone isn't a vulnerability.
HOW TO FIX	Open the path yourself and confirm it's meant to be public. If it's leftover debug/test/admin tooling that should have been removed, take it down or restrict it by IP allowlist. If it's intentional, no action needed.
URGENCY	Fix within 1 month

WHAT IT IS	/account/ returned HTTP 200, a real distinct response rather than the site's soft-404 page. This path isn't one a typical visitor would stumble onto by browsing the site normally, which is why it's worth a quick manual look — but its existence alone isn't a vulnerability.
BUSINESS RISK	/account/ returned HTTP 200, a real distinct response rather than the site's soft-404 page. This path isn't one a typical visitor would stumble onto by browsing the site normally, which is why it's worth a quick manual look — but its existence alone isn't a vulnerability.
HOW TO FIX	Open the path yourself and confirm it's meant to be public. If it's leftover debug/test/admin tooling that should have been removed, take it down or restrict it by IP allowlist. If it's intentional, no action needed.
URGENCY	Fix within 1 month

PORT REFERENCE TABLE

PORT	SERVICE	RISK	DESCRIPTION
HTTPS	Cookie Security	MEDIUM	Cookie 'idToken' is missing the HttpOnly flag(s). Observed attributes: Domain=target.com, Path=/, Sa
HTTPS	Cookie Security	MEDIUM	Cookie 'TealeafAkaSid' is missing the Secure, HttpOnly, SameSite flag(s). Observed attributes: Expir
DNS	DKIM Key Strength	MEDIUM	DKIM key for selector 'k1' appears to be approximately 1024 bits — 1024-bit RSA keys are considered
DNS	DKIM Key Strength	MEDIUM	DKIM key for selector 'selector1' appears to be approximately 1024 bits — 1024-bit RSA keys are cons
DNS	DKIM Key Strength	MEDIUM	DKIM key for selector 's2' appears to be approximately 1024 bits — 1024-bit RSA keys are considered
DNS	DKIM Key Strength	MEDIUM	DKIM key for selector 'mandrill' appears to be approximately 1024 bits — 1024-bit RSA keys are consi
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	2 third-party script source(s) loaded without Subresource Integrity (SRI): assets.targeting1.com, cl
HTTPS	Permissions Policy	LOW	Missing Permissions-Policy — browser features like camera, microphone, and location aren't restricte
HTTPS	Cookie Security	LOW	Cookie 'adScriptData' is missing the HttpOnly flag(s). Observed attributes: Domain=target.com, Path=

HTTPS	Cookie Security	LOW	Cookie 'sapphire' is missing the HttpOnly flag(s). Observed attributes: Max-Age=2629746, Expires=Thu
HTTPS	Cookie Security	LOW	Cookie 'onboardingGuest' is missing the HttpOnly flag(s). Observed attributes: Domain=.target.com, P
HTTPS	Cookie Security	LOW	Cookie 'fiatsCookie' is missing the HttpOnly flag(s). Observed attributes: Domain=.target.com, Path=
HTTPS	Cookie Security	LOW	Cookie 'sddStore' is missing the Secure, HttpOnly, SameSite flag(s). Observed attributes: Domain=.ta
HTTPS	Cookie Security	LOW	Cookie 'GuestLocation' is missing the HttpOnly, SameSite flag(s). Observed attributes: Path=/, Secur
HTTPS	Cookie Security	LOW	Cookie 'visitorId' is missing the HttpOnly flag(s). Observed attributes: Max-Age=63072000, Expires=T
HTTPS	Cookie Security	LOW	Cookie '_pxhd' is missing the Secure, HttpOnly flag(s). Observed attributes: Max-Age=31536000, SameS
PATH	Content Discovery	LOW	/login/ returned HTTP 200 with no authentication challenge. This confirms a login or admin entry poi
HTTPS	IP Reputation	INFO	Public IP 151.101.194.187 (hosted by Fastly, Inc.) has no abuse reports on AbuseIPDB in the last 90
HTTPS	IP WHOIS / Network Ownership	INFO	Public IP 151.101.194.187 is registered to Fastly, Inc. — network: SKYCA-3 — block: 151.101.0.0 – 15
DNS	BIMI Not Configured	INFO	DMARC is already enforcing, so this domain qualifies for BIMI — a verified logo shown next to emails
PATH	Content Discovery	INFO	/robots.txt is present and was used to seed additional path checks.
PATH	Content Discovery	INFO	/bp/c/ returned HTTP 200, a real distinct response rather than the site's soft-404 page. This path i
PATH	Content Discovery	INFO	/bp/p/ returned HTTP 200, a real distinct response rather than the site's soft-404 page. This path i
PATH	Content Discovery	INFO	/deals/ returned HTTP 200, a real distinct response rather than the site's soft-404 page. This path
PATH	Content Discovery	INFO	/gc?k returned HTTP 200, a real distinct response rather than the site's soft-404 page. This path is
PATH	Content Discovery	INFO	/cart returned HTTP 200, a real distinct response rather than the site's soft-404 page. This path is
PATH	Content Discovery	INFO	/yr returned HTTP 200, a real distinct response rather than the site's soft-404 page. This path isn'

PATH	Content Discovery	INFO	/account/ returned HTTP 200, a real distinct response rather than the site's soft-404 page. This pat
------	----------------------	------	--

NEXT STEPS

→ Address the 8 MEDIUM risk finding(s) this week. Start with 'Insecure Cookie Flags (idToken)': 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).

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.