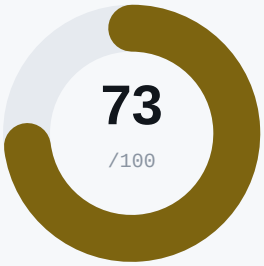
Sample report — a worked example for a fictional store, not a real assessment.

Threnda

Security Report

northbound-supply.com

Assessed 22 August 2026 · Report version 0.1.0



Risk level	**Medium**
Grade	**C**
Scored findings	**3**
Platform	**Shopify** (100% confidence)

Critical	1
High	0
Medium	1
Low	1
Info	2

Executive summary

northbound-supply.com scores 73/100 (grade C), a medium risk level. The assessment found 1 critical, 1 medium and 1 low issues across 3 areas. HEADERS accounts for the largest share of the deduction. The storefront runs on Shopify. The highest-impact fix is to implement a Content-Security-Policy header to reduce XSS and skimming risk.

What to do first

In priority order. The first item removes the most exposure for the least work.

1. Implement a Content-Security-Policy header to reduce XSS and skimming risk.
2. Set the Secure flag on every cookie the store issues.
3. Accept the exposure but harden admin access, which is what it points at.
4. Harden the parts of Shopify that remain your responsibility.
5. Reduce and constrain third-party scripts running on the storefront.

At a glance

#	FINDING	SEVERITY	AREA
1	Missing Content-Security-Policy header	Critical	HTTP Headers
2	Cookies set without the Secure flag	Medium	Cookies
3	Internal myshopify.com domain exposed in page source	Low	Exposure
4	Shopify storefront detected (100% confidence)	Info	Platform
5	Third-party app risk: 4 external script sources	Info	Platform

Findings in detail

Each finding lists what was observed, why it matters to the business, and the exact steps to close it.

Critical · 1

1. Missing Content-Security-Policy header

CRITICAL

WHAT
The response does not set Content-Security-Policy.

EVIDENCE
`Header absent from https://northbound-supply.com/`

BUSINESS IMPACT
Without a CSP, any injected script - from a compromised third-party app, a malicious theme edit, or a stored XSS bug - can skim card details and customer data straight off the checkout page.

HOW TO FIX

Implement a Content-Security-Policy header to reduce XSS and skimming risk.

1. Deploy the policy in Content-Security-Policy-Report-Only mode first and collect violations.
2. Enumerate the script hosts your store legitimately needs (payments, analytics, apps).
3. Tighten to an allow-list, remove 'unsafe-inline', then enforce.
4. Add frame-ancestors to cover clickjacking at the same time.
5. Shopify note: On Shopify, storefront response headers are set by the platform; use Shopify's checkout/CSP controls or a Shopify Plus proxy - theme.liquid meta tags cannot set frame-ancestors or a full policy.

```
Content-Security-Policy: default-src 'self'; script-src 'self' https://cdn.shopify.com;
frame-ancestors 'none'; base-uri 'self'
```

headers.missing-content-security-policy · confidence: confirmed

Medium · 1

2. Cookies set without the Secure flag

MEDIUM

WHAT
Cookies without Secure are transmitted over plain HTTP.

EVIDENCE
`Affected: cart_currency`

BUSINESS IMPACT
These cookies leak over plain HTTP; impact is limited because none of them appear to carry a session.

HOW TO FIX

Set the Secure flag on every cookie the store issues.

1. Add Secure to all Set-Cookie headers, starting with session cookies.
2. Serve the site exclusively over HTTPS so nothing breaks.
3. Shopify note: Shopify sets its own storefront cookies; focus on cookies added by your theme, apps or custom pixels.

```
Set-Cookie: session=...; Secure; HttpOnly; SameSite=Lax
```

cookies.missing-secure · confidence: confirmed

Low · 1

3. Internal myshopify.com domain exposed in page source

LOW

WHAT The store's permanent Shopify domain is visible to anyone reading the HTML.

EVIDENCE `northbound-supply.myshopify.com`

BUSINESS IMPACT Gives attackers a stable target for credential-stuffing and phishing against the store's admin login.

HOW TO FIX

Accept the exposure but harden admin access, which is what it points at.

1. Enforce MFA on all staff accounts.
2. Review the login audit log for unfamiliar access.

platform.shopify-internal-domain-exposed · confidence: confirmed

Info · 2

4. Shopify storefront detected (100% confidence)

INFO

WHAT The store runs on Shopify's hosted platform.

EVIDENCE `header x-shopid: 48213307; header x-shopify-stage: production; Powered-By: Shopify; Shopify CDN asset host`

BUSINESS IMPACT Shopify handles infrastructure patching, but theme code, installed apps and account access remain the merchant's responsibility.

HOW TO FIX

Harden the parts of Shopify that remain your responsibility.

1. Require multi-factor authentication for every staff account.
2. Review staff permissions and remove dormant accounts.
3. Audit installed apps and uninstall anything unused.
4. Restrict who can edit theme code and publish themes.

platform.shopify-detected · confidence: confirmed

5. Third-party app risk: 4 external script sources

INFO

WHAT Shopify apps and marketing tags load JavaScript from 4 external hosts. Each one can execute arbitrary code on the storefront.

EVIDENCE `cdn.reviews-vendor.example, connect.facebook.net, static.klaviyo.com, www.googletagmanager.com`

BUSINESS IMPACT This is the standard Magecart attack path: compromise one third-party script and skim card data from every checkout, without ever touching the store's own systems.

HOW TO FIX

Reduce and constrain third-party scripts running on the storefront.

1. Inventory every installed app and marketing tag; remove what is unused.
2. Check each remaining vendor's security posture and breach history.
3. Constrain the survivors with a CSP script-src allow-list.
4. Add Subresource Integrity to any script served from a fixed URL.

platform.third-party-script-risk · confidence: confirmed

Scope and method

This is a passive, external assessment: Threnda requested public pages over HTTP and HTTPS, completed a TLS handshake, and queried public DNS records. No authentication was used, no vulnerability was exploited, and no customer data was accessed. Findings describe observable configuration, not proven exploitability.

DETECTED TECHNOLOGY

Shopify, Google Tag Manager, Cloudflare, Klaviyo, Meta Pixel