VULNIFY

Comprehensive Pentest Report

CONFIDENTIAL

August 17, 2026

TARGET

https://shop.example.com

Hostname shop.example.com · Suite comprehensive_core · Tier comprehensive

0	4	3	0	0
CRITICAL	HIGH	MEDIUM	LOW	INFO

High residual risk

Job UUID sample-comprehensive-pentest-report

Included retests: 2 · AI status: sample

DISCLOSURE

This is an automated, evidence-backed Vulnify assessment. It is not a PCI DSS, SOC 2, ISO 27001, or certified penetration test, and it is not a human consultant letter or attestation.

This automated evidence pack can support your auditor's review. It is not a PCI DSS, SOC 2, or ISO 27001 certification, and it is not a certified or human-signed penetration test.

1. Executive summary

Automated penetration test identified 7 finding(s) on shop.example.com

Vulnify ran the comprehensive_core suite against https://shop.example.com. Findings by severity: Critical 0, High 4, Medium 3, Low 0, Info 0. Check matrix: 4 passed, 6 failed, 6 not applicable.

Leadership remediation this week

Fund XSS encoding and JWT cookie storage this week on shop.example.com, then disable GraphQL introspection on api.shop.example.com.

This is an automated, evidence-backed Vulnify assessment. It is not a PCI DSS, SOC 2, ISO 27001, or certified penetration test, and it is not a human consultant letter or attestation.

2. Do this week

The highest-severity fixes, in order, with the first remediation step and the URL to change.

1. **[HIGH] Reflected XSS on search**

   URL: /search?q=
   First step: HTML-encode the recorded parameter on output for the context (HTML body, attribute, or JS).

2. **[HIGH] Missing anti-CSRF tokens (4 paths)**

   URL: /account/email
   First step: Require a per-session anti-CSRF token on every state-changing form and JSON POST.

3. **[HIGH] Path traversal on file download**

   URL: /file?path=
   First step: Resolve the file against an allowlisted directory and reject .. segments after normalization.

4. **[HIGH] JWT in first-party HTML**

   URL: /
   First step: Apply this fix on host shop.example.com.

5. **[MEDIUM] Missing security headers (3)**

   URL: /
   First step: Add Content-Security-Policy with a default-src of self and explicit script/style sources. Avoid unsafe-inline once scripts are nonced.

- 4 extra hosts found, 3 verified and tested, 1 left unverified or skipped.
- This is an automated, evidence-backed Vulnify assessment. It is not a PCI DSS, SOC 2, ISO 27001, or certified penetration test, and it is not a human consultant letter or attestation.
- Comprehensive Pentest: a second auth profile was not supplied, so multi-role and BOLA-across-roles did not run.
- No OpenAPI spec was uploaded, so API and BOLA checks used crawl and hint discovery only.
- SQL injection probes ran and did not confirm a finding. Unconfirmed SQL injection is not reported as a vulnerability.
- Stored XSS was not confirmed. No persistent sink was in scope or the check was not applicable.
- DOM XSS was not confirmed. No client-side sink was in scope or the check was not applicable.
- Authenticated crawl did not run, so in-app pages behind login were not mapped.
- BOLA / object-level API access was not tested. That check needs object IDs and a login session.
- Privilege escalation was not tested. That check needs a working authenticated role.
- No repository zip was uploaded, so secret scanning did not run.

4. Attack surface appendix

3 extra hosts found, 2 verified and tested, 1 left unverified or skipped.

Hostname	Status	Role	Notes
shop.example.com	tested	primary	primary
api.shop.example.com	tested	extra	inventory
admin.shop.example.com	tested	extra	inventory
staging.shop.example.com	out_of_scope	extra	Not selected

Last pentest comparison

Compared with prior job sample-pentest-report: 2 new, 4 still open, 0 resolved.

Prior job: sample-pentest-report

5. Authorization matrix

A second auth profile was not supplied, so role B columns are Not tested.

URL / object	Unauth	Role A	Role B	Outcome
https://api.shop.example.com/orders/1001	401	200	not_tested	role_a_ok

6. Scope and methodology

Target hostname shop.example.com was tested under exact-hostname scope lock. Suite comprehensive_core executed 16 catalog checks. Engines and Vulnify modules invoked: must_checks. This run was unauthenticated. Authenticated authz and business-logic checks are not applicable until a healthy auth profile is supplied. Checks marked not_applicable were skipped because preconditions were missing (for example auth, object IDs, or optional binaries). Findings are never invented to fill those gaps. This Comprehensive Pentest includes two targeted retest cycles. Extra sibling hosts are tested only after they are verified (cap 5).

Engines

- must_checks

Nuclei and ZAP run allowlisted policies only. Destructive DoS classes are denied.

This automated evidence pack can support your auditor's review. It is not a PCI DSS, SOC 2, or ISO 27001 certification, and it is not a certified or human-signed penetration test.

Scope letter

- Requester: Authorized buyer
- Consent: Recorded with the order
- Terms: current Vulnify Terms of Service
- Verified hostnames: shop.example.com, api.shop.example.com, admin.shop.example.com
- Window: see job timestamps to see job timestamps
- Suite: comprehensive_core · Auth: cookie
- Engines: Allowlisted Nuclei, ZAP, SQLMap, httpx, and Vulnify modules

Methodology map

baseline.tls_transport — WSTG-CRYP-01 · PTES Vulnerability Analysis · NIST SP 800-115 3.2 Review Techniques · ASVS V9.1
baseline.security_headers — WSTG-CONF-12 · PTES Vulnerability Analysis · NIST SP 800-115 3.2 Review Techniques · ASVS V14.4
web.xss.reflected — WSTG-INPV-01 · PTES Exploitation · NIST SP 800-115 3.4 Target Vulnerability Validation · ASVS V5.2
web.xss.stored — WSTG-INPV-02 · PTES Exploitation · NIST SP 800-115 3.4 Target Vulnerability Validation · ASVS V5.2
web.xss.dom — WSTG-CLNT-01 · PTES Exploitation · NIST SP 800-115 3.4 Target Vulnerability Validation · ASVS V5.2
web.csrf.state_changing — WSTG-SESS-05 · PTES Vulnerability Analysis · NIST SP 800-115 3.4 Target Vulnerability Validation · ASVS V4.2
web.lfi_path_traversal — WSTG-ATHZ-01 · PTES Exploitation · NIST SP 800-115 3.4 Target Vulnerability Validation · ASVS V12.3
web.sqli.reflect_or_blind — WSTG-INPV-05 · PTES Exploitation · NIST SP 800-115 3.4 Target Vulnerability Validation · ASVS V5.3
discover.crawl_auth — WSTG-INFO-06 · PTES Intelligence Gathering · NIST SP 800-115 3.3 Target Identification and Analysis · ASVS V1.2
api.bola_object_id — WSTG-ATHZ-04 · PTES Exploitation · NIST SP 800-115 3.4 Target Vulnerability Validation · ASVS V4.1
logic.priv_escalation — WSTG-ATHZ-03 · PTES Exploitation · NIST SP 800-115 3.4 Target Vulnerability Validation · ASVS V4.2
recon.subdomain_inventory — WSTG-INFO-01 · PTES Intelligence Gathering · NIST SP 800-115 3.3 Target Identification and Analysis · ASVS V14.1
api.graphql.introspection — WSTG-INFO-10 · PTES Intelligence Gathering · NIST SP 800-115 3.3 Target Identification and Analysis · ASVS V13.4
session.jwt_exposure — WSTG-SESS-02 · PTES Vulnerability Analysis · NIST SP 800-115 3.2 Review Techniques · ASVS V3.5
session.jwt_alg_none — WSTG-CRYP-04 · PTES Exploitation · NIST SP 800-115 3.4 Target Vulnerability Validation · ASVS V3.5
repo.secrets.gitleaks — WSTG-CONF-05 · PTES Intelligence Gathering · NIST SP 800-115 3.2 Review Techniques · ASVS V2.10

Retest proof

Entitlement: 2 included retest(s). Used: 0. Remaining: 2.

8. Findings

7 confirmed finding card(s). Remediation is numbered; it is not collapsed to a one-liner.

HIGH

Reflected XSS on search

Check web.xss.reflected · /search?q= · Host shop.example.com · Parameter q · Engines must_checks

How to reproduce

Send GET to /search?q=, observe HTTP 200, using parameter q.

Business impact

Reflected XSS steals sessions, injects UI, and is a common phishing primitive on the same origin.

Remediation · Owner application · Effort medium

- 1. HTML-encode the recorded parameter on output for the context (HTML body, attribute, or JS).
- 2. Add or tighten Content-Security-Policy so inline script cannot run.
- 3. Prefer framework auto-escaping and avoid inserting unsanitized HTML.
- 4. If the stack was not identified, apply the generic application-layer steps and say so in the ticket.

Verify before retest

- Replay the recorded payload and confirm it is encoded, not executed.
- Confirm CSP is present on that response.

After the change is live, use the included targeted retest from the workspace. Do not mark the finding closed until the retest records a before/after result.

Evidence

GET · /search?q= · HTTP 200

Path hits: /search?q=, /search?q=

Browser proof: no browser proof

HIGH

Missing anti-CSRF tokens (4 paths)

Check web.csrf.state_changing · /account/email · Host shop.example.com · Engines must_checks

How to reproduce

Send POST to /account/email, observe HTTP 200.

Business impact

Without anti-CSRF controls, a foreign site can change email, password, or money-moving actions as the victim.

Remediation · Owner application · Effort medium

- 1. Require a per-session anti-CSRF token on every state-changing form and JSON POST.
- 2. Verify Origin or Referer as a second check.
- 3. Set SameSite=Strict or Lax on session cookies.
- 4. If the stack was not identified, apply the generic application-layer steps and say so in the ticket.

Verify before retest

- POST the recorded action without a token and confirm rejection.
- Confirm SameSite on the session cookie.

After the change is live, use the included targeted retest from the workspace. Do not mark the finding closed until the retest records a before/after result.

Evidence

POST · /account/email · HTTP 200

Path hits: /account/email, /password/reset, /account/email, /account/email

Browser proof: no browser proof

HIGH

Path traversal on file download

Check web.lfi_path_traversal · /file?path= · Host shop.example.com · Parameter path · Engines must_checks

How to reproduce

Send GET to /file?path=, observe HTTP 200, using parameter path.

Business impact

Path traversal reads secrets, source, and sometimes credentials off disk.

Remediation · Owner application · Effort medium

- 1. Resolve the file against an allowlisted directory and reject .. segments after normalization.
- 2. Do not concatenate user paths onto filesystem roots.
- 3. Serve downloads by opaque IDs, not filenames.
- 4. If the stack was not identified, apply the generic application-layer steps and say so in the ticket.

Verify before retest

- Replay the recorded path parameter and confirm the marker file is not returned.

After the change is live, use the included targeted retest from the workspace. Do not mark the finding closed until the retest records a before/after result.

Evidence

GET · /file?path= · HTTP 200

Path hits: /file?path=, /file?path=

Browser proof: no browser proof

MEDIUM

Missing security headers (3)

Check baseline.security_headers · / · Host shop.example.com · Engines http_probe

How to reproduce

Open / and confirm the recorded evidence still appears.

Business impact

Missing security headers make XSS, clickjacking, and MIME confusion easier. Auditors look for these on every HTTPS response.

Remediation · Owner platform · Effort small

- 1. Add Content-Security-Policy with a default-src of self and explicit script/style sources. Avoid unsafe-inline once scripts are nonced.
- 2. Add Strict-Transport-Security, X-Content-Type-Options: nosniff, Referrer-Policy: strict-origin-when-cross-origin, and a frame-ancestors policy (or X-Frame-Options).
- 3. Apply headers on every HTML and API error page, not only the homepage.
- 4. If the stack was not identified, apply the generic application-layer steps and say so in the ticket.

Verify before retest

- Fetch the recorded URL and confirm each missing header from the finding evidence is now present.
- Confirm error pages (404/500) also send the same headers.

After the change is live, use the included targeted retest from the workspace. Do not mark the finding closed until the retest records a before/after result.

Evidence

Missing headers: content-security-policy, strict-transport-security, x-content-type-options

Path hits: /, /

Browser proof: no browser proof

MEDIUM

Cookie security flags incomplete

Check session.cookie_flags_fixation · / · Host shop.example.com · Engines must_checks

How to reproduce

Open / and confirm the recorded evidence still appears.

Business impact

Missing cookie flags or session fixation lets an attacker keep a stolen or planted session.

Remediation · Owner application · Effort medium

- 1. Set Secure, HttpOnly, and SameSite on session cookies.
- 2. Rotate the session ID at login.
- 3. Bind the session to a User-Agent or IP only if that matches your product threat model.
- 4. If the stack was not identified, apply the generic application-layer steps and say so in the ticket.

Verify before retest

- Log in and confirm a new session cookie with flags set.

After the change is live, use the included targeted retest from the workspace. Do not mark the finding closed until the retest records a before/after result.

Evidence

Path hits: /, /

Browser proof: no browser proof

MEDIUM

GraphQL introspection enabled on api.shop.example.com

Check api.graphql.introspection · /graphql · Host api.shop.example.com · Engines must_checks

How to reproduce

Send POST to /graphql, observe HTTP 200.

Business impact

Public introspection hands attackers the full schema, including internal mutations.

Remediation · Owner application · Effort small

- 1. Apply this fix on host api.shop.example.com.
- 2. Disable introspection in production.
- 3. Put GraphQL behind authentication if it is not a public catalog.
- 4. If the stack was not identified, apply the generic application-layer steps and say so in the ticket.

Verify before retest

- POST an introspection query anonymously and confirm it is rejected.

After the change is live, use the included targeted retest from the workspace. Do not mark the finding closed until the retest records a before/after result.

Evidence

POST · /graphql · HTTP 200

Path hits: /graphql

Browser proof: no browser proof

HIGH

JWT in first-party HTML

Check session.jwt_exposure · / · Host shop.example.com · Engines must_checks

How to reproduce

Send GET to /, observe HTTP 200.

Business impact

JWTs in HTML or localStorage are stolen by XSS and used as bearer sessions.

Remediation · Owner application · Effort medium

- 1. Apply this fix on host shop.example.com.
- 2. Prefer HttpOnly cookies over localStorage for session JWTs.
- 3. Shorten TTL and rotate refresh tokens.
- 4. Do not embed JWTs in first-party HTML.
- 5. If the stack was not identified, apply the generic application-layer steps and say so in the ticket.

Verify before retest

- Confirm the recorded page no longer contains a live JWT.

After the change is live, use the included targeted retest from the workspace. Do not mark the finding closed until the retest records a before/after result.

Evidence

GET · / · HTTP 200

Path hits: /

eyJhbGciOi...[REDACTED]

Browser proof: no browser proof

Prioritize CISA KEV and high-EPSS findings first, then Critical/High severity. Retest after each remediation batch.

- **#1 [HIGH] Reflected XSS on search**
  Start at /search?q=
- **#2 [HIGH] Missing anti-CSRF tokens (4 paths)**
  Start at /account/email
- **#3 [HIGH] Path traversal on file download**
  Start at /file?path=
- **#4 [HIGH] JWT in first-party HTML**
  Start at /
- **#5 [MEDIUM] Missing security headers (3)**
  Start at /
- **#6 [MEDIUM] Cookie security flags incomplete**
  Start at /
- **#7 [MEDIUM] GraphQL introspection enabled on api.shop.example.com**
  Start at /graphql

10. Likely attack paths

Identified 2 combined path(s) from related findings. Validate and remediate path members together.

Session theft path

Weak session cookies combined with missing transport hardening and script injection risk can enable session hijacking.

Injection impact path

Confirmed injection near sensitive or administrative surfaces increases the chance of data theft or privilege abuse.

11. Check matrix

PASSED	baseline.tls_transport — TLS 1.2+ served on the sample host
FAILED	baseline.security_headers — CSP, HSTS, and X-Content-Type-Options missing
FAILED	web.xss.reflected — Reflected XSS confirmed on /search
NOT_APPL	web.xss.stored — No persistent sink in sample scope
NOT_APPL	web.xss.dom — No DOM sink in sample scope
FAILED	web.csrf.state_changing — State-changing forms lacked CSRF tokens
FAILED	web.lfi_path_traversal — Path parameter read a local file
PASSED	web.sqli.reflect_or_blind — SQL injection was not confirmed
NOT_APPL	discover.crawl_auth — Sample report is unauthenticated
NOT_APPL	api.bola_object_id — No authenticated object IDs in sample
NOT_APPL	logic.priv_escalation — No login role in sample
PASSED	recon.subdomain_inventory — 3 extra hosts found; 2 verified and tested
FAILED	api.graphql.introspection — Introspection open on /graphql
FAILED	session.jwt_exposure — Replayable JWT in page

PASSED session.jwt_alg_none — alg=none rejected

NOT_APPL repo.secrets.gitleaks — No repository zip uploaded

12. Document control

- Document: Vulnify Comprehensive Pentest Report
- Classification: Confidential
- Generated: August 17, 2026
- Job UUID: sample-comprehensive-pentest-report
- Hostname: shop.example.com
- Suite / tier: comprehensive_core / comprehensive
- Retest entitlement: 2
- AI narrative status: sample
- This automated evidence pack can support your auditor's review. It is not a PCI DSS, SOC 2, or ISO 27001 certification, and it is not a certified or human-signed penetration test.
- This is an automated, evidence-backed Vulnify assessment. It is not a PCI DSS, SOC 2, ISO 27001, or certified penetration test, and it is not a human consultant letter or attestation.