Saqr صقر

Confidential

Technical report for developers

Prepared for

shop.yourcompany.com

Security Assessment

Testing window	Report date	Report ID	Version
4 to 23 November 2026	23 November 2026	SQR-0001-01	1

Document control

About this report

Report	Technical report for developers
Prepared for	shop.yourcompany.com
Engagement	Security Assessment
Tasks included	Quick task on shop.yourcompany.com, Full task on shop.yourcompany.com, Quick task on shop.yourcompany.com, Quick task on shop.yourcompany.com, Full task on shop.yourcompany.com (5 tasks)
Prepared by	Security Assessment Team
Report ID	SQR-0001-01
Version and date	Version 1, 23 November 2026
Classification	Confidential
Distribution	The people responsible for shop.yourcompany.com only. Do not forward outside the distribution without the owner's approval.

Contents

1 Summary 3

2 Scope and rules of engagement 4

3 Methodology and standards 5

4 Findings index 6

5 Findings 7

A. Asset inventory 28

B. Coverage 28

C. Tools used 28

D. Timeline 28

E. Integrity 29

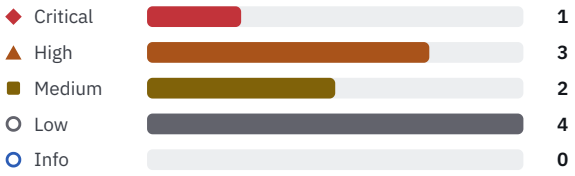
Section 1

Summary

Testing of shop.yourcompany.com (4 to 23 November 2026, 5 tasks) found 10 security findings: 1 critical, 3 high, 2 medium and 4 low. The overall risk is high. The most serious is invoice PDFs can be downloaded without signing in (F-02). 1 finding has already been fixed.

Overall risk	Findings	Critical	High	Medium
▲ High	10	◆ 1	▲ 3	■ 2

Findings by severity



Remediation status



Top risks

Ref	Severity	Risk	Business impact
F-02	▲ High	**Invoice PDFs can be downloaded without signing in**	Anyone who guesses or counts through invoice numbers can download every customer's invoice: name, billing address and what was bought.
F-03	▲ High	**Another customer's order can be read by changing the order id**	Any customer can read the orders of all other customers: full name, email address, phone number, shipping address and what they bought.
F-04	▲ High	**Stored script injection in the profile display name**	A customer can save a name that contains script.
F-05	■ Medium	**Error pages reveal stack traces and file paths**	The details make other issues easier to find: table and column names, file layout and exact library versions.
F-06	■ Medium	**No rate limit on the sign-in endpoint**	Passwords can be guessed at scale, and passwords leaked from other sites can be tried against all customer emails.

Section 2

Scope and rules of engagement

Target	`shop.yourcompany.com`
Testing window	4 to 23 November 2026

Tasks included

Task	Window	Status
Quick task on shop.yourcompany.com	4 November 2026	Completed
Full task on shop.yourcompany.com	6 November 2026	Completed
Quick task on shop.yourcompany.com	12 November 2026	Completed
Quick task on shop.yourcompany.com	19 November 2026	Completed
Full task on shop.yourcompany.com	From 23 November 2026	Running

Authorised scope

Type	Target	Why it is in scope
Domain	`shop.yourcompany.com`	The target of this report.

Rules of engagement

- Testing was limited to the authorised scope listed above. Requests to any other host were blocked by the platform.
- Actions with a risk of changing data or service availability waited for a person's approval before they ran.
- Government, military and education networks are blocked by default and were not tested.

Section 3

Methodology and standards

Testing followed the phases of NIST SP 800-115 (planning, discovery, attack and reporting) and the test cases of the OWASP Web Security Testing Guide. Each finding is mapped to the references below so engineers can look up the weakness, the test and the verification requirement.

Standard	How this report uses it
OWASP WSTG	The test case that found or confirms each finding (for example WSTG-INPV-05).
OWASP ASVS	The verification chapter to check the fix against.
OWASP Top 10 (2021)	The risk category of each finding.
CWE	The weakness type; a CWE stated by a scanner is kept as recorded.
CVSS v4.0	The base score of each finding. A vector written in the finding is used as stated; otherwise the score is the baseline of the weakness class and should be confirmed for the client's environment.
MITRE ATT&CK	The technique someone could use, where one applies.
NIST SP 800-115	The phases of the engagement and the evidence practice.

Severity and proof

Severity	Meaning
◆ Critical	Exploitable remotely with little effort, leading to a serious breach of data or control. Fix immediately.
▲ High	Could give someone access to data or functions they should not have. Fix within 30 days.
■ Medium	Needs specific conditions or has a limited effect on its own. Fix within 90 days.
○ Low	Low impact on its own; fix in normal maintenance.
○ Info	No direct risk; noted for hardening.

Proof level is how far exploitation was taken, from 0 (observed) to 4 (impact demonstrated). Verified means the platform re-checked the claim against recorded traffic.

Section 4

Findings index

All 10 findings, most severe first. The page column points to each finding's detail.

Ref	Severity	Finding	CVSS	Status	Page
F-01	◆ Critical	Secret API key exposed in the public JavaScript bundle	9.3	Fixed	7
F-02	▲ High	Invoice PDFs can be downloaded without signing in	8.8	Open	9
F-03	▲ High	Another customer's order can be read by changing the order id	7.1	Open	11
F-04	▲ High	Stored script injection in the profile display name	7.1	Open	14
F-05	■ Medium	Error pages reveal stack traces and file paths	6.9	Open	16
F-06	■ Medium	No rate limit on the sign-in endpoint	6.3	Open	18
F-07	○ Low	Security headers are missing on every page	2.3	Open	20
F-08	○ Low	Session cookie is set without the Secure and HttpOnly flags	2.1	Open	22
F-09	○ Low	Source map files are publicly downloadable	2.1	Risk accepted	24
F-10	○ Low	The uploads directory lists its files	2.1	False positive	26

Finding F-01

Secret API key exposed in the public JavaScript bundle

◆ Critical CVSS v4.0 **9.3** ● Fixed Proof level 3 · Verified Traffic evidence: 1

CVSS v4.0 vector (severity baseline)
`CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:N/SC:N/SI:N/SA:N`

CWE	OWASP Top 10
CWE-200 Exposure of Sensitive Information to an Unauthorized Actor	A01:2021 Broken Access Control
Test (WSTG)	**MITRE ATT&CK**
WSTG-INFO-05	No technique applies

Affected asset
`shop.yourcompany.com/assets/index-8f3c1a.js`

Description

The shop's main JavaScript bundle, `/assets/index-8f3c1a.js`, is downloaded by every visitor. It contains the secret API key of the payment service (`PAYMENT_SECRET_KEY`, prefix `sk_live_`). Secret keys are meant to be used only by server code.

Business impact

Anyone can read the key from the page source without signing in.

With a secret key a person can act as the shop at the payment service: list customers and payments, issue refunds and create charges. This can lead to direct financial loss and to exposure of customer payment records.

Evidence

Request

```
GET /assets/index-8f3c1a.js HTTP/1.1
Host: shop.yourcompany.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/131.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
```

Response

```
HTTP/1.1 200 OK
Date: Tue, 03 Nov 2026 22:02:56 GMT
Content-Type: application/javascript; charset=utf-8
Content-Length: 428
Connection: keep-alive
X-Powered-By: Express
Cache-Control: public, max-age=31536000, immutable

/* index-8f3c1a.js */
const cfg=
{API_BASE:"/api",PAYMENT_PUBLISHABLE_KEY:"pk_live_51NqT8uKc...REDACTED...4hTz",PAYMENT_SECRET_KEY:
"sk_live_51NqT8uKc...REDACTED...x9Qe",SENTRY_ENV:"production"};
export async function createCharge(order){return fetch("https://api.payments.example/v1/charges",
{method:"POST",headers:{Authorization:"Bearer "+cfg.PAYMENT_SECRET_KEY},body:new
```

```
URLSearchParams({amount:order.total,currency:"usd"})})}
```

Reproduction steps

1. Open `https://shop.yourcompany.com/` in a browser and view the network requests.
2. Open `/assets/index-8f3c1a.js` and search for `sk_live_`.
3. The value of `PAYMENT_SECRET_KEY` is in clear text.

```
GET /assets/index-8f3c1a.js HTTP/1.1
Host: shop.yourcompany.com
```

Remediation

Revoke this key in the payment service dashboard today and create a new one. Keep the new key only in server-side environment settings, and move the charge request from the browser to a server route. Only the publishable key (`pk_live_`) belongs in browser code. Check the payment service logs for calls you do not recognise since the key was first published.

References

CWE-200 Exposure of Sensitive Information to an Unauthorized Actor · `https://cwe.mitre.org/data/definitions/200.html`
OWASP Web Security Testing Guide WSTG-INFO-05 · `https://owasp.org/www-project-web-security-testing-guide/`
OWASP Top 10 A01:2021 Broken Access Control · `https://owasp.org/Top10/`

Verification and retest

Verified, proof level 3. Retest: Not retested.

Finding F-02

Invoice PDFs can be downloaded without signing in

▲ High CVSS v4.0 **8.8** ● Open Proof level 3 · Verified Traffic evidence: 1

CVSS v4.0 vector (weakness class baseline)

`CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:L/VA:N/SC:N/SI:N/SA:N`

CWE	OWASP Top 10
CWE-862 Missing Authorization	A01:2021 Broken Access Control
Test (WSTG)	**MITRE ATT&CK**
WSTG-ATHZ-02	No technique applies

Affected asset

`shop.yourcompany.com/invoices/INV-2026-1043.pdf`

Description

Invoices are generated into the public folder `/invoices/` and named after the order number. The files are served to anyone, signed in or not.

Business impact

Anyone who guesses or counts through invoice numbers can download every customer's invoice: name, billing address and what was bought.

Evidence

Request

```
GET /invoices/INV-2026-1043.pdf HTTP/1.1
Host: shop.yourcompany.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/131.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
```

Response

```
HTTP/1.1 200 OK
Date: Sun, 22 Nov 2026 21:46:56 GMT
Content-Type: application/pdf
Content-Length: 47
Connection: keep-alive
X-Powered-By: Express
Content-Disposition: inline; filename="INV-2026-1043.pdf"

%PDF-1.7
[binary PDF, 48,112 bytes, not shown]
```

Reproduction steps

1. In a private browser window, with no session, request `https://shop.yourcompany.com/invoices/INV-2026-1043.pdf`.
2. The PDF is returned.

Remediation

Move the invoice files out of the public folder. Serve them through a route that checks the session and that the invoice belongs to the signed-in customer, or use short-lived signed links. Rename existing files so the old addresses stop working.

References

CWE-862 Missing Authorization · `https://cwe.mitre.org/data/definitions/862.html`
OWASP Web Security Testing Guide WSTG-ATHZ-02 · `https://owasp.org/www-project-web-security-testing-guide/`
OWASP Top 10 A01:2021 Broken Access Control · `https://owasp.org/Top10/`

Verification and retest

Verified, proof level 3. Retest: Not retested.

Finding F-03

Another customer's order can be read by changing the order id

▲ High CVSS v4.0 **7.1** ● Open Proof level 3 · Verified Traffic evidence: 2

CVSS v4.0 vector (weakness class baseline)

`CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:N/VA:N/SC:N/SI:N/SA:N`

CWE	OWASP Top 10
CWE-639 Authorization Bypass Through User-Controlled Key	A01:2021 Broken Access Control
Test (WSTG)	**MITRE ATT&CK**
WSTG-ATHZ-04	No technique applies

Affected assets

`shop.yourcompany.com/api/orders/1042` `shop.yourcompany.com/api/orders/1043`

Description

The endpoint `GET /api/orders/{id}` checks that the caller is signed in, but not that the order belongs to the caller. Order ids are sequential numbers, so every order in the shop can be read one by one.

Business impact

Any customer can read the orders of all other customers: full name, email address, phone number, shipping address and what they bought.

This is a personal data exposure that may need to be reported under data protection rules.

Evidence

Request

```
GET /api/orders/1042 HTTP/1.1
Host: shop.yourcompany.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/131.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
Cookie: sid=s%3A7J... (shortened)
```

Response

```
HTTP/1.1 200 OK
Date: Thu, 05 Nov 2026 21:54:56 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 297
Connection: keep-alive
X-Powered-By: Express

{"id":1042,"customer_id":318,"email":"alex@yourcompany.com","status":"shipped","total":"84.00","it
ems":[{"sku":"MUG-BLUE","qty":2,"price":"12.00"},{"sku":"TOTE-
01","qty":1,"price":"60.00"}],"shipping_address":{"name":"Alex Tester","line1":"12 Test
Street","city":"Springfield","postcode":"00012"}}
```

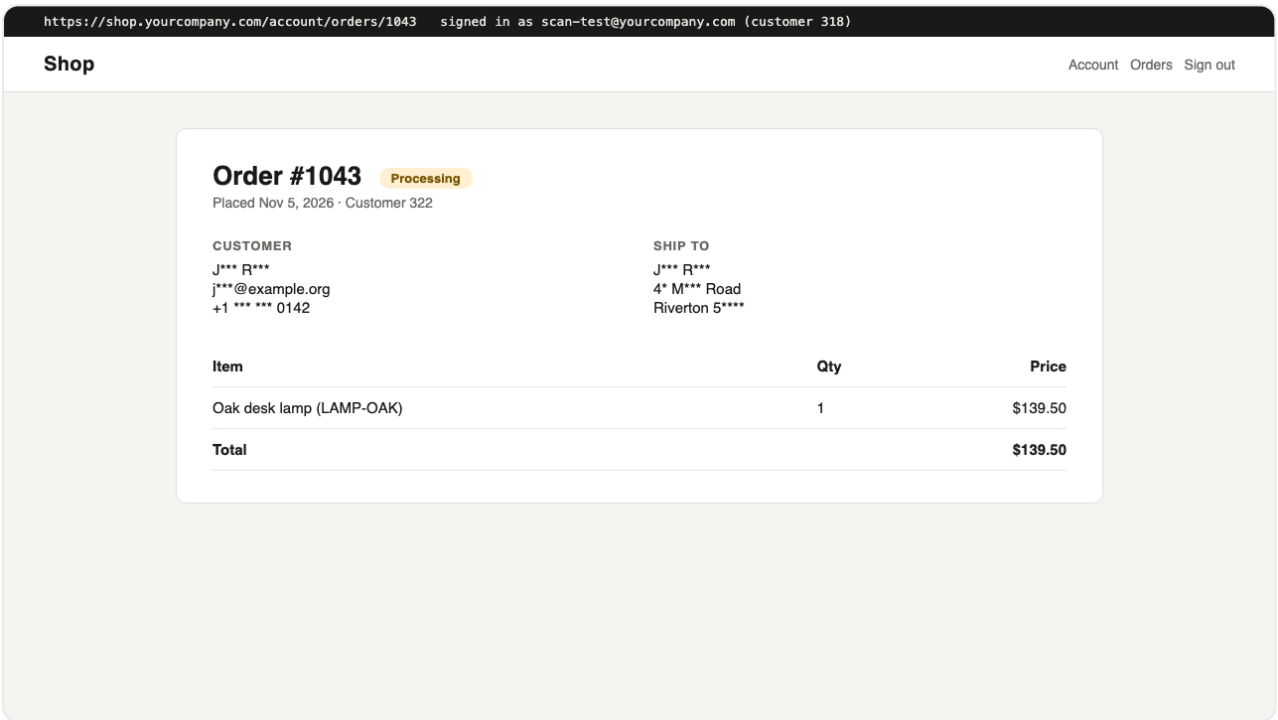
Request

```
GET /api/orders/1043 HTTP/1.1
Host: shop.yourcompany.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/131.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
Cookie: sid=s%3A7J... (shortened)
```

Response

```
HTTP/1.1 200 OK
Date: Thu, 05 Nov 2026 21:55:56 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 275
Connection: keep-alive
X-Powered-By: Express

{"id":1043,"customer_id":322,"email":"j***@example.org","status":"processing","total":"139.50","it
ems":[{"sku":"LAMP-OAK","qty":1,"price":"139.50"}],"shipping_address":{"name":"J***
R***","line1":"4* M*** Road","city":"Riverton","postcode":"5****"},"phone":"+1 *** *** 0142"}
```



Browser screenshot screenshot-order-1043.png, SHA-256 5b0c3b83a5ae6e06f928dc9ea75e2a427bb37bdc3e53b5fcbfbb7004fa053214

Reproduction steps

1. Sign in as any customer and open one of your own orders, for example `/api/orders/1042`.
2. Request `/api/orders/1043` with the same session.
3. The response is `200 OK` with `customer_id: 322`, which is a different customer.

```
GET /api/orders/1043 HTTP/1.1
Host: shop.yourcompany.com
Cookie: sid=<sessi... (shortened) of customer 318>
```

Remediation

Check ownership on the server for every order route. Load the order with both conditions in the query, for example `WHERE id = $1 AND customer_id = $2` using the customer id from the session, and return 404 when nothing matches. Apply the same check to invoices, addresses and any other route that takes an id.

References

CWE-639 Authorization Bypass Through User-Controlled Key · https://cwe.mitre.org/data/definitions/639.html
OWASP Web Security Testing Guide WSTG-ATHZ-04 · https://owasp.org/www-project-web-security-testing-guide/
OWASP Top 10 A01:2021 Broken Access Control · https://owasp.org/Top10/

Verification and retest

Verified, proof level 3. Retest: Not retested.

Finding F-04

Stored script injection in the profile display name

▲ High CVSS v4.0 **7.1** ● Open Proof level 3 · Verified Traffic evidence: 2

CVSS v4.0 vector (severity baseline)

`CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:L/VA:N/SC:N/SI:N/SA:N`

CWE	OWASP Top 10
CWE-79 Improper Neutralization of Input During Web Page Generation	A03:2021 Injection
Test (WSTG)	**MITRE ATT&CK**
WSTG-CLNT-01	T1189 Drive-by Compromise

Affected asset

PUT /api/profile

Description

The profile display name is stored as typed and written into pages as raw HTML. The name appears on product reviews, order confirmations and the staff order view.

Business impact

A customer can save a name that contains script.

The script then runs for every visitor of a page that shows the name, with that visitor's session. Combined with the session cookie that is readable by scripts, this allows taking over other accounts, including staff accounts.

Evidence

Request

```
PUT /api/profile HTTP/1.1
Host: shop.yourcompany.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/131.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
Cookie: sid=s%3A7J... (shortened)
Content-Type: application/json
Content-Length: 74

{"display_name":"Alex<img src=x onerror=console.log('saqr-marker-7f2a')>"}
```

Response

```
HTTP/1.1 200 OK
Date: Thu, 05 Nov 2026 22:04:56 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 84
Connection: keep-alive
X-Powered-By: Express

{"ok":true,"display_name":"Alex<img src=x onerror=console.log('saqr-marker-7f2a')>"}
```

Request

```
GET /products/mug-blue HTTP/1.1
Host: shop.yourcompany.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/131.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
```

Response

```
HTTP/1.1 200 OK
Date: Thu, 05 Nov 2026 22:05:56 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 231
Connection: keep-alive
X-Powered-By: Express

<section class="reviews">
  <article class="review">
    <span class="review-author">Alex<img src=x onerror=console.log('saqr-marker-7f2a')></span>
    <p class="review-body">Great mug, arrived quickly.</p>
  </article>
</section>
```

Reproduction steps

1. Sign in and open the profile page.
2. Save a display name that contains an HTML element with a harmless marker.
3. Open a product page where a review by this account is shown: the marker runs.

Remediation

Escape the name when it is written into a page. In React, render it as text (`{user.displayName}`) and remove the `dangerouslySetInnerHTML` call in the review component. Also validate the name on save (letters, digits, spaces and common punctuation, at most 60 characters) and add a Content-Security-Policy as a second layer.

References

CWE-79 Improper Neutralization of Input During Web Page Generation · `https://cwe.mitre.org/data/definitions/79.html`
OWASP Web Security Testing Guide WSTG-CLNT-01 · `https://owasp.org/www-project-web-security-testing-guide/`
OWASP Top 10 A03:2021 Injection · `https://owasp.org/Top10/`

Verification and retest

Verified, proof level 3. Retest: Not retested.

Finding F-05

Error pages reveal stack traces and file paths

Medium CVSS v4.0 **6.9** Open Proof level 2 · Verified Traffic evidence: 1

CVSS v4.0 vector (weakness class baseline)

`CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N`

CWE	OWASP Top 10
CWE-200 Exposure of Sensitive Information to an Unauthorized Actor	A01:2021 Broken Access Control
Test (WSTG)	**MITRE ATT&CK**
WSTG-INFO-05	No technique applies

Affected assets

`shop.yourcompany.com/api/products/abc` `/srv/app/src/routes/products.js`

Description

The API runs with detailed errors turned on in production. Any request that causes an error returns the stack trace and the database query.

Business impact

The details make other issues easier to find: table and column names, file layout and exact library versions.

They do not expose customer data on their own.

Evidence

Request

```
GET /api/products/abc HTTP/1.1
Host: shop.yourcompany.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/131.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
```

Response

```
HTTP/1.1 500 Internal Server Error
Date: Thu, 05 Nov 2026 22:14:56 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 397
Connection: keep-alive
X-Powered-By: Express

{"error":"invalid input syntax for type integer: \"abc\"","query":"SELECT * FROM products WHERE id
= $1","stack":"error: invalid input syntax for type integer: \"abc\"\n    at
Parser.parseErrorMessage (/srv/app/node_modules/pg-protocol/dist/parser.js:287:98)\n    at
getProduct (/srv/app/src/routes/products.js:41:22)\n    at Layer.handle (/srv/app/node_module...
```

Reproduction steps

1. Request `/api/products/abc`.
2. The `500` response contains `stack`, `query` and server file paths.

Remediation

Set `NODE_ENV=production` and add one error handler that logs the full error on the server and returns only a short message and a request id to the client. Validate ids before they reach the database.

References

CWE-200 Exposure of Sensitive Information to an Unauthorized Actor · `https://cwe.mitre.org/data/definitions/200.html`
OWASP Web Security Testing Guide WSTG-INFO-05 · `https://owasp.org/www-project-web-security-testing-guide/`
OWASP Top 10 A01:2021 Broken Access Control · `https://owasp.org/Top10/`

Verification and retest

Verified, proof level 2. Retest: Not retested.

Finding F-06

No rate limit on the sign-in endpoint

Medium CVSS v4.0 **6.3** Open Proof level 2 · Verified Traffic evidence: 1

CVSS v4.0 vector (weakness class baseline)

`CVSS:4.0/AV:N/AC:L/AT:P/PR:N/UI:N/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N`

CWE	OWASP Top 10
CWE-307 Improper Restriction of Excessive Authentication Attempts	A07:2021 Identification and Authentication Failures
Test (WSTG)	**MITRE ATT&CK**
WSTG-ATHN-03	T1110 Brute Force

Affected asset

`POST /api/auth/login`

Description

`POST /api/auth/login` has no limit on failed attempts per account or per address.

Business impact

Passwords can be guessed at scale, and passwords leaked from other sites can be tried against all customer emails.

Accounts with weak or reused passwords would be taken over.

Evidence

Request

```
POST /api/auth/login HTTP/1.1
Host: shop.yourcompany.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/131.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
Content-Type: application/json
Content-Length: 68

{"email":"scan-test@yourcompany.com","password":"(hidden)"}
```

Response

```
HTTP/1.1 401 Unauthorized
Date: Thu, 05 Nov 2026 22:09:56 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 37
Connection: keep-alive
X-Powered-By: Express

{"error":"Invalid email or password"}
```

Reproduction steps

1. Send 40 sign-in requests for one account with wrong passwords within 20 seconds.
2. Every response is `401` with the same body. There is no `429` and no delay.

Remediation

Limit failed sign-ins per account and per address, for example 5 per minute with a growing delay, using `express-rate-limit` with a shared store. After repeated failures ask for a human check, and notify the account owner by email.

References

CWE-307 Improper Restriction of Excessive Authentication Attempts · `https://cwe.mitre.org/data/definitions/307.html`
OWASP Web Security Testing Guide WSTG-ATHN-03 · `https://owasp.org/www-project-web-security-testing-guide/`
OWASP Top 10 A07:2021 Identification and Authentication Failures · `https://owasp.org/Top10/`

Verification and retest

Verified, proof level 2. Retest: Not retested.

Finding F-07

Security headers are missing on every page

Low CVSS v4.0 **2.3** Open Proof level 2 · Verified Traffic evidence: 1

CVSS v4.0 vector (weakness class baseline)

`CVSS:4.0/AV:N/AC:L/AT:P/PR:N/UI:P/VC:N/VI:L/VA:N/SC:N/SI:N/SA:N`

CWE	OWASP Top 10
CWE-693 Protection Mechanism Failure	A05:2021 Security Misconfiguration
Test (WSTG)	**MITRE ATT&CK**
WSTG-CONF-12	No technique applies

Affected assets

shop.yourcompany.com/ /login /account

Description

The shop does not send the standard browser security headers. They are missing on the home page, the sign-in page and the account page.

Business impact

On their own the missing headers do not expose data.

They remove protections that limit the damage of other issues: without a Content-Security-Policy an injected script runs freely, without X-Frame-Options the shop can be framed by another site, and without Strict-Transport-Security a first visit can be downgraded to plain HTTP.

Evidence

Request

```
GET / HTTP/1.1
Host: shop.yourcompany.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/131.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
```

Response

```
HTTP/1.1 200 OK
Date: Tue, 03 Nov 2026 22:00:56 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 196
Connection: keep-alive
X-Powered-By: Express

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Shop</title>
<script type="module" src="/assets/index-8f3c1a.js"></script>
</head>
<body><div id="root"></div></body>
```

```
</html>
```

Reproduction steps

1. Request `https://shop.yourcompany.com/`.
2. Read the response headers: none of the headers listed above are present.

Remediation

Add the headers in one place, for example with the `helmet` middleware in the Express app: `Strict-Transport-Security: max-age=31536000; includeSubDomains`, `X-Content-Type-Options: nosniff`, `X-Frame-Options: DENY`, `Referrer-Policy: strict-origin-when-cross-origin` and a Content-Security-Policy that allows only your own scripts.

References

CWE-693 Protection Mechanism Failure · `https://cwe.mitre.org/data/definitions/693.html`
OWASP Web Security Testing Guide WSTG-CONF-12 · `https://owasp.org/www-project-web-security-testing-guide/`
OWASP Top 10 A05:2021 Security Misconfiguration · `https://owasp.org/Top10/`

Verification and retest

Verified, proof level 2. Retest: Not retested.

Finding F-08

Session cookie is set without the Secure and HttpOnly flags

Low CVSS v4.0 **2.1** Open Proof level 2 · Verified Traffic evidence: 1

CVSS v4.0 vector (severity baseline)

`CVSS:4.0/AV:N/AC:H/AT:P/PR:L/UI:P/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N`

CWE	OWASP Top 10
CWE-200 Exposure of Sensitive Information to an Unauthorized Actor	A01:2021 Broken Access Control
Test (WSTG)	**MITRE ATT&CK**
WSTG-INFO-05	No technique applies

Affected asset

GET /login

Description

The shop sets its session cookie `sid` without the `Secure`, `HttpOnly` and `SameSite` attributes.

Business impact

A script running in the page can read the session cookie, which turns any script injection issue into account takeover.

Without `Secure` the cookie can travel over an unencrypted connection.

Evidence

Request

```
GET /login HTTP/1.1
Host: shop.yourcompany.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/131.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
```

Response

```
HTTP/1.1 200 OK
Date: Tue, 03 Nov 2026 22:04:56 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 109
Connection: keep-alive
X-Powered-By: Express
Set-Cookie: sid=s%3AQ1... (shortened); Path=/

<!doctype html>
<html lang="en"><head><title>Sign in</title></head><body><div id="root"></div></body></html>
```

Reproduction steps

1. Request the sign-in page.
2. Read the `Set-Cookie` response header: `sid=...; Path=/`.

Remediation

Set the cookie with `Secure; HttpOnly; SameSite=Lax`. In an Express session configuration: `cookie: { secure: true, httpOnly: true, sameSite: "lax" }`.

References

CWE-200 Exposure of Sensitive Information to an Unauthorized Actor · `https://cwe.mitre.org/data/definitions/200.html`
OWASP Web Security Testing Guide WSTG-INFO-05 · `https://owasp.org/www-project-web-security-testing-guide/`
OWASP Top 10 A01:2021 Broken Access Control · `https://owasp.org/Top10/`

Verification and retest

Verified, proof level 2. Retest: Not retested.

Finding F-09

Source map files are publicly downloadable

Low CVSS v4.0 **2.1** Risk accepted Proof level 2 · Verified Traffic evidence: 1

CVSS v4.0 vector (severity baseline)
`CVSS:4.0/AV:N/AC:H/AT:P/PR:L/UI:P/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N`

CWE	OWASP Top 10
CWE-200 Exposure of Sensitive Information to an Unauthorized Actor	A01:2021 Broken Access Control
Test (WSTG)	**MITRE ATT&CK**
WSTG-INFO-05	No technique applies

Affected asset
`shop.yourcompany.com/assets/index-8f3c1a.js.map`

Description

Source maps for the production bundle are served next to the JavaScript files.

Business impact

The original source code, comments and internal route names (including the admin pages) are readable by anyone.

This helps a person understand the app, but does not expose data on its own.

Evidence

Request

```
GET /assets/index-8f3c1a.js.map HTTP/1.1
Host: shop.yourcompany.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/131.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
```

Response

```
HTTP/1.1 200 OK
Date: Thu, 05 Nov 2026 22:19:56 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 273
Connection: keep-alive
X-Powered-By: Express

{"version":3,"file":"index-8f3c1a.js","sources":
["../../src/main.tsx","../../src/api/client.ts","../../src/pages/Checkout.tsx","../../src/pages/ad
min/Orders.tsx"],"sourcesContent":["import React from \"react\";\n// TODO: move payment key to the
server before launch\n..."]}
```

Reproduction steps

1. Request `/assets/index-8f3c1a.js.map`.
2. The response lists the original files and their content.

Remediation

Turn off source maps for production builds (`build.sourcemap: false` in the Vite config), or upload them to your error tracker and do not deploy them.

References

CWE-200 Exposure of Sensitive Information to an Unauthorized Actor · `https://cwe.mitre.org/data/definitions/200.html`
OWASP Web Security Testing Guide WSTG-INFO-05 · `https://owasp.org/www-project-web-security-testing-guide/`
OWASP Top 10 A01:2021 Broken Access Control · `https://owasp.org/Top10/`

Verification and retest

Verified, proof level 2. Retest: Not retested.

Finding F-10

The uploads directory lists its files

Low CVSS v4.0 **2.1** False positive Proof level 2 · Verified Traffic evidence: 1

CVSS v4.0 vector (severity baseline)
`CVSS:4.0/AV:N/AC:H/AT:P/PR:L/UI:P/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N`

CWE	OWASP Top 10
CWE-548 Exposure of Information Through Directory Listing	A01:2021 Broken Access Control
Test (WSTG)	**MITRE ATT&CK**
WSTG-CONF-04	T1083 File and Directory Discovery

Affected asset
`shop.yourcompany.com/uploads/`

Description

The web server shows a file index for `/uploads/` and its subfolders.

Business impact

Anyone can browse all uploaded files, including photos customers attached to return requests.

File names can contain order numbers.

Evidence

Request

```
GET /uploads/ HTTP/1.1
Host: shop.yourcompany.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/131.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
```

Response

```
HTTP/1.1 200 OK
Date: Thu, 05 Nov 2026 22:24:56 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 325
Connection: keep-alive
X-Powered-By: Express

<html><head><title>Index of /uploads/</title></head><body><h1>Index of /uploads/</h1><hr><pre>
<a href="products/">products/</a>          03-Nov-2026 09:12       -
<a href="returns/">returns/</a>           04-Nov-2026 16:40       -
<a href="avatars/">avatars/</a>           05-Nov-2026 08:02       -
</pre><hr></body></html>
```

Reproduction steps

1. Request `/uploads/`.
2. The response is an index page with links to every folder and file.

Remediation

Turn off directory indexes (`autoindex off;` in nginx, or remove `serve-index` in Express). Store private uploads outside the public folder and serve them through a route that checks the session.

References

CWE-548 Exposure of Information Through Directory Listing · `https://cwe.mitre.org/data/definitions/548.html`
OWASP Web Security Testing Guide WSTG-CONF-04 · `https://owasp.org/www-project-web-security-testing-guide/`
OWASP Top 10 A01:2021 Broken Access Control · `https://owasp.org/Top10/`

Verification and retest

Verified, proof level 2. Retest: Not retested.

Appendices

A. Asset inventory

Asset	Findings
/account	F-07
/login	F-07
/srv/app/src/routes/products.js	F-05
GET /login	F-08
POST /api/auth/login	F-06
PUT /api/profile	F-04
shop.yourcompany.com/	F-07
shop.yourcompany.com/api/orders/1042	F-03
shop.yourcompany.com/api/orders/1043	F-03
shop.yourcompany.com/api/products/abc	F-05
shop.yourcompany.com/assets/index-8f3c1a.js	F-01
shop.yourcompany.com/assets/index-8f3c1a.js.map	F-09
shop.yourcompany.com/invoices/INV-2026-1043.pdf	F-02
shop.yourcompany.com/uploads/	F-10

B. Coverage

Coverage tracking was turned off for this engagement.

C. Tools used

Tool	Kind	Runs
curl	Command line	24
browser_navigate	Built-in browser	8
jq	Command line	7
browser_login	Built-in browser	3
sslscan	Command line	3
browser_type	Built-in browser	1
traffic_search	Platform	1

D. Timeline

Time	Event
4 November 2026, 05:40 UTC	Quick task on shop.yourcompany.com started
4 November 2026, 05:44 UTC	First finding recorded
4 November 2026, 05:49 UTC	Quick task on shop.yourcompany.com finished

Time	Event
6 November 2026, 08:45 UTC	Full task on shop.yourcompany.com started
6 November 2026, 09:26 UTC	Full task on shop.yourcompany.com finished
12 November 2026, 14:16 UTC	Quick task on shop.yourcompany.com started
12 November 2026, 14:24 UTC	Quick task on shop.yourcompany.com finished
19 November 2026, 14:20 UTC	Quick task on shop.yourcompany.com started
19 November 2026, 14:28 UTC	Quick task on shop.yourcompany.com finished
23 November 2026, 08:43 UTC	Full task on shop.yourcompany.com started
23 November 2026, 08:52 UTC	Last of 10 findings recorded
23 November 2026, 09:00 UTC	Report version 1 generated

E. Integrity

Use these values to check that this document and its evidence have not changed. The evidence hash covers every finding's recorded evidence and report, every bound request and response, and every cited screenshot. The audit chain head is the last entry of Saqr's hash-chained audit log when the report was generated.

Report ID	`SQR-0001-01`
Generated	23 November 2026, 09:00 UTC
Evidence set	45 items
Evidence SHA-256	`04af19c78e9b9c2784eaf98634e2a059a8530c3cf771b371d093956e1601d21e`
Audit chain head	`#412 9c41d07e5b2a86f3c1e0d4b7a95f28e6d3c0b1a4f7e2d9c8b5a6f3e0d1c2b4a7`

Sign-off and attestation

Security Assessment Team attests that the testing described in this report was performed within the authorised scope and testing window, that every finding is backed by the evidence recorded during the engagement, and that the report presents the results without omission. Findings reflect the state of the systems during testing.

Prepared by
Security Assessment Team

Reviewed by
Name, role and date

Accepted by
The owner of shop.yourcompany.com