Vextrix

FIND THE GAPS / SEE THE RISKS / MOVE FORWARD

AI-ASSISTED SECURITY ASSESSMENT / TECHNICAL REPORT

# WEB APPLICATION SECURITY ASSESSMENT.

Web application assessed against OWASP Web Security Testing Guide. Evidence-backed findings, two independent CVSS calculations, mapped weaknesses.

IN SCOPE / VERIFIED

**ASSESSMENT TARGET**
portal.northwind.example

**REPORT REFERENCE**
VX-SAMPLE-WEB-001

**ISSUED**
19 NOV 2026
VERSION 1

01 / EXECUTIVE SUMMARY

# Security, at a glance.

Eight findings were confirmed across the Northwind Analytics analytics portal: three High, four Medium and one Low on the official severity policy. Three of them are reachable without any account, and the two highest-rated findings give an ordinary analyst account read and write access to the application database and a route into the internal network.

| 0 | 3 | 4 | 1 | 0 |
|---|---|---|---|---|
| CRITICAL | HIGH | MEDIUM | LOW | INFO |

Counts are confirmed findings only, classified by the official report severity (CVSS v3.1 Base). Candidates, inconclusive and rejected results are not counted here.

## WHAT THIS ASSESSMENT ESTABLISHED

The portal's perimeter controls behave as intended for the interactive interface: sign-in works, roles are distinguished and the session cookie is not readable by script. The weaknesses are concentrated one layer below that, in routes and queries that were added for automation and were not brought under the same authorisation and input-handling rules as the pages they sit beside.

That pattern explains why remediation clusters so tightly. Two of the three highest findings are closed by the same change in the data-access layer, and the injection and request-forgery findings share a root cause — a value supplied by the caller being passed to an interpreter, in one case SQL and in the other an HTTP client.

## MOST SIGNIFICANT VALIDATED RISKS

1. **SQL Injection in the Saved Report Filter** HIGH
   VEX-WEB-001 — An analyst account reads and modifies the application database directly, bypassing every control the application layer applies.
2. **Server-Side Request Forgery in the Webhook Validator** HIGH
   VEX-WEB-002 — The application makes attacker-directed requests from inside the network boundary and returns what it receives.
3. **Invoice Export Endpoint Accessible Without Authentication** HIGH
   VEX-WEB-003 — Commercial records are retrievable with no account at all, from a route that was never placed behind authentication.

## RECOMMENDED REMEDIATION FOCUS

- Close the two injection paths first (VEX-WEB-001, VEX-WEB-002): they give the widest access and both stem from unvalidated input reaching an interpreter.
- Make billing authorisation deny-by-default and tenant-scoped, which closes the anonymous export and the cross-tenant read together (VEX-WEB-003, VEX-WEB-004).
- Treat the remaining four as hardening work on a normal release schedule; none of them grants access on its own.

**Assessment limitations.** Testing ran against a staging replica, so findings are attributed to the application as deployed in staging. Production was not tested and nothing here establishes production exposure. No destructive testing, no denial-of-service testing and no social engineering were performed. Bulk extraction was deliberately not attempted.

02 / METHODOLOGY

# A defined method. A clear evidence boundary.

Authenticated and unauthenticated testing of the portal and its API against the OWASP Web Security Testing Guide v4.2, combining automated discovery with manual validation. Every candidate was independently re-tested before being confirmed, and candidates the evidence did not support were rejected rather than reported.

**01 Authorise**
Scope, permissions, exclusions and rules of engagement agreed before any testing.

**02 Discover**
Map the reachable surface: hosts, endpoints, parameters, services and authentication.

**03 Explore**
Bounded, evidence-driven testing within the authorised scope and operating limits.

**04 Evidence**
Record what was observed, with the request, response or output that demonstrates it.

**05 Review**
Validate each candidate, score it, classify the weakness and reject what the evidence does not support.

**06 Retest**
Where agreed, recheck reported weaknesses after remediation using fresh evidence.

These six stages describe the Vextrix product lifecycle. They do not replace the assessment-specific technical methodology below, and they are not a claim that every stage produced complete coverage.

## SCOPE AS AUTHORISED

| IN SCOPE | TYPE | ENVIRONMENT |
|---|---|---|
| portal.northwind.example | **Web application** | Staging replica |
| https://portal.northwind.example/api/v2/* | **Application API** | Staging replica |

## EXPLICIT EXCLUSIONS

- The production deployment and its data.
- Third-party identity, payment and email providers integrated with the portal.
- The underlying cloud account, its control plane and its network infrastructure.
- Denial-of-service testing, physical security and social engineering.

## TECHNICAL METHODOLOGY APPLIED

Primary reference: **OWASP Web Security Testing Guide**; risk taxonomy: **OWASP Top 10**.

| STANDARD | EDITION | APPLIED TO |
|---|---|---|
| WSTG | **4.2** | https://owasp.org/www-project-web-security-testing-guide/v42/ |

| STANDARD | EDITION | APPLIED TO |
|---|---|---|
| ASVS | **5.0.0** | https://github.com/OWASP/ASVS/tree/v5.0.0 |
| NIST-800-115 | **2008** | https://csrc.nist.gov/pubs/sp/800/115/final |

A reference states which guidance a test followed. It is not certification, and it is not a claim that the standard was covered exhaustively.

## PROCEDURES PERFORMED

| | |
|---|---|
| **Discovery** | Endpoint, parameter and technology inventory across the interactive interface and the API, authenticated as each role and unauthenticated. |
| **Authentication** | Sign-in flow, session issue and renewal, cookie attributes, logout, and attempt-limiting behaviour under repeated failures. |
| **Authorisation** | Object and function access compared between two tenants and two roles, including direct requests to routes the interface does not link. |
| **Input handling** | Injection testing across query, path and body parameters, with validation in a browser context where execution was claimed. |
| **Configuration** | Response headers, transport settings and files reachable from the web root. |

## AUTHENTICATION

| MODE | AVAILABLE | ACTUALLY EXERCISED |
|---|---|---|
| Anonymous | **yes** | yes |
| Analyst (standard role) | **yes** | yes |
| Administrator | **yes** | yes |
| Multi-factor | **no** | no |
| API token | **yes** | yes |

## COVERAGE AND WHAT WAS NOT ESTABLISHED

| OUTCOME | TESTS | MEANING |
|---|---|---|
| Confirmed finding | **8** | Validated independently; released and counted in this report. |
| Tested, no issue confirmed | **34** | The test ran and the evidence did not support a finding. |
| Rejected candidate | **6** | Observed and investigated; validation contradicted the hypothesis. |
| Inconclusive | **1** | Validation ran but could not settle the question. A gap, not a pass. |
| Not supported | **1** | Not available in this environment; recorded rather than skipped silently. |
| Not applicable | **4** | Discovery established the condition the check requires does not exist here. |

**Gaps.** The following were not established:

- File upload handling — the upload service was unavailable for the whole window.
- Multi-factor enrolment and recovery — disabled in the staging replica.
- Production configuration — out of scope; staging may differ.

### HOW RESULTS WERE INTERPRETED

- Do not imply every WSTG area was assessed; report only performed or materially limited areas.
- Reflected input is not executable script until observed executing in a browser context.

02B / WEB APPLICATION ANALYSIS

# What the web application actually exposes.

## AUTHENTICATION AND SESSION TESTING

Session behaviour observed across sign-in, renewal and sign-out.

| CONTROL | OBSERVED | RESULT |
|---|---|---|
| Session cookie HttpOnly | **Set** | Pass |
| Session cookie Secure | **Absent** | VEX-WEB-008 |
| Session cookie SameSite | **Lax** | Pass |
| Session rotated on sign-in | **Yes** | Pass |
| Session invalidated on sign-out | **Yes, server-side** | Pass |
| Idle expiry | **30 minutes** | Pass |
| Attempt limiting | **None observed in 500 attempts** | VEX-WEB-005 |
| Strict-Transport-Security | **Absent** | Contributes to VEX-WEB-008 |

Controls marked Pass were exercised and behaved as intended during the testing window. That is not a guarantee of correctness in every state.

## BUSINESS LOGIC TESTING

Workflows exercised against the rules the application states it enforces.

| WORKFLOW | ATTEMPTED ABUSE | OUTCOME |
|---|---|---|
| Report scheduling | **Schedule a report for another tenant** | Rejected correctly |
| Invoice retrieval | **Request another tenant's identifier** | VEX-WEB-004 |
| Seat allocation | **Exceed the licensed seat count** | Rejected correctly |
| Export | **Request an export with no session** | VEX-WEB-003 |

A section that does not appear in this report was not performed or did not apply. Its absence is not evidence that the corresponding area is secure.

03 / FINDINGS INDEX

# Eight findings. One severity policy.

Sorted by official report severity, then remediation priority. Every row resolves to a detailed finding with two independent CVSS calculations and a mapped CWE weakness.

| # | FINDING | AFFECTED | SEVERITY | CVSS 3.1 | CVSS 4.0 | PRIORITY |
|---|---|---|---|---|---|---|
| 01 | **SQL Injection in the Saved Report Filter**<br>VEX-WEB-001 · CWE-89 | 1 endpoint | HIGH | **8.1**<br>High | 8.6<br>High | immediate<br>Confirmed |
| 02 | **Server-Side Request Forgery in the Webhook Validator**<br>VEX-WEB-002 · CWE-918 | 1 endpoint | HIGH | **7.7**<br>High | 8.3<br>High | immediate<br>Confirmed |
| 03 | **Invoice Export Endpoint Accessible Without Authentication**<br>VEX-WEB-003 · CWE-306 | 1 endpoint | HIGH | **7.5**<br>High | 8.7<br>High | immediate<br>Confirmed |
| 04 | **Invoice Records Readable Across Tenants via Sequential Identifier**<br>VEX-WEB-004 · CWE-639 | 1 endpoint, 2 tenants | MEDIUM | **6.5**<br>Medium | 7.1<br>High | immediate<br>Confirmed |
| 05 | **No Rate Limiting on the Authentication Endpoint**<br>VEX-WEB-005 · CWE-307 | 1 endpoint | MEDIUM | **6.5**<br>Medium | 6.9<br>Medium | high<br>Confirmed |
| 06 | **Reflected Cross-Site Scripting in the Search Parameter**<br>VEX-WEB-006 · CWE-79 | 1 endpoint | MEDIUM | **6.1**<br>Medium | 5.1<br>Medium | high<br>Confirmed |
| 07 | **Application Dependency Manifest Publicly Accessible**<br>VEX-WEB-007 · CWE-552 | 2 files | MEDIUM | **5.3**<br>Medium | 6.9<br>Medium | planned<br>Confirmed |
| 08 | **Session Cookie Issued Without the Secure Attribute**<br>VEX-WEB-008 · CWE-614 | 1 cookie | LOW | **3.7**<br>Low | 6.3<br>Medium | planned<br>Confirmed |

### HOW MISMATCHED CVSS RATINGS ARE HANDLED

This report has one official severity: **CVSS v3.1 Base**. Its qualitative band controls the summary counts, the ordering above and each finding's primary badge. **CVSS v4.0 Base** is calculated independently and shown with its own rating and complete vector. The two are never averaged, converted, or silently resolved to the higher value.

Where the ratings differ, both are printed as calculated. Remediation priority may still sit above the severity band when exposure or evidence confidence justifies it — the rationale is recorded with each finding.

04 / TECHNICAL FINDING 1 OF 8

VEX-WEB-001 / INJECTION

HIGH

# SQL Injection in the Saved Report Filter

**Validation** Confirmed **HTTP method** GET **Authentication state** Authenticated (standard analyst) **Required role** analyst

**WSTG reference** WSTG-INPV-05 **OWASP Top 10 category** A03:2021 Injection

## AFFECTED ASSET

APPLICATION
https://portal.northwind.example

AFFECTED ENDPOINT
/api/v2/reports/saved?filter=

PARAMETER / INPUT LOCATION
filter (query string)

CVSS v3.1 · PRIMARY
**8.1** HIGH
CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N

CVSS v4.0 · ADDITIONAL
**8.6** HIGH
CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:H/VA:N/SC:N/SI:N/SA:N

CWE CLASSIFICATION **CWE-89 — Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')**

## DESCRIPTION

The saved-report feature lets an analyst narrow results with a free-text filter. The filter value is concatenated into the SQL statement that backs the report rather than bound as a parameter, so the database parses the submitted text as query syntax.

A single quotation mark in the filter produced a database syntax error rather than an empty result set or a validation message — the first indication that the input reaches the parser. A boolean condition appended to the filter then changed the number of rows returned in a way that tracked the truth of the injected condition, and a UNION-based test returned a value selected from outside the report's own tables.

The injection point is reachable by any account with the analyst role, which is the default role assigned to new users in the fixture tenant.

## IMPACT AND IMPLICATION

**Demonstrated.** Arbitrary SELECT statements executed in the application's database context, returning data from tables the report feature does not expose. The database user also accepted a write statement in a transaction that the test rolled back, establishing that the same path permits modification.

**Potential, not demonstrated.** A party with an analyst account could read the full customer and billing tables and modify stored records. Extraction of the complete dataset, credential material or onward access to the database host was not attempted and is not established by this finding.

**Conditions required.** Any authenticated account holding the analyst role.

## REPRODUCTION AND VALIDATION

1. Sign in as a standard analyst account and open any saved report.
2. Submit a single quotation mark as the filter value and observe that the response carries a database syntax error rather than a validation message.
3. Submit a filter containing an always-true boolean condition and record the row count; repeat with an always-false condition and observe the count change.
4. Submit a UNION SELECT returning a constant and confirm the constant appears in the rendered report, proving the statement structure is attacker-controlled.

### EVIDENCE · EV-NWA-001

**REQUEST**

```
GET /api/v2/reports/saved?filter=%27 HTTP/1.1
Host: portal.northwind.example
Accept: application/json
Cookie: nwa_session=[REDACTED]
```

**OBSERVED RESPONSE**

```
HTTP/1.1 500 Internal Server Error
Content-Type: application/json

{
  "error": "query failed",
  "detail": "unterminated quoted string at or near \"'\"",
  "position": 118
}
```

**EV-NWA-001 · HTTP exchange.** The database error discloses that filter text is parsed as SQL. A validation failure would have produced a 400 with a field message; the 500 with parser output is what distinguishes injection from ordinary input rejection.
Redacted artifact SHA-256 6864c9351cc16165755fb6ab7f758731d546951cc82c3618fc610985e5b44ffd.

### EVIDENCE · EV-NWA-002

**REQUEST**

```
GET /api/v2/reports/saved?
filter=x%27%20UNION%20SELECT%20%27vextrix-probe%27,NULL,NULL--
HTTP/1.1
Host: portal.northwind.example
Cookie: nwa_session=[REDACTED]
```

**OBSERVED RESPONSE**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "rows": [
    {"label": "vextrix-probe", "value": null, "period": null}
  ],
  "row_count": 1
}
```

**EV-NWA-002 · HTTP exchange.** A UNION SELECT returns an attacker-chosen constant inside the report output, demonstrating control of the statement rather than of the result text alone.
Redacted artifact SHA-256 1fa9f91b06d76e3d58a45f0dffb46b859c4fd55b402ad934a205381b2e990d1a.

### COMPARISON CONTROL

A control request submitting the same filter text with the quotation mark removed returned HTTP 200 with an empty result set and no error, confirming the error is produced by the quotation mark reaching the parser and not by the request shape.

### RECOMMENDED REMEDIATION

1. Replace string concatenation in the saved-report query builder with parameterised statements, binding the filter value rather than interpolating it.
2. Where the filter must select a column or sort direction, resolve the supplied value against an explicit allow-list of column names instead of passing it through to SQL.
3. Remove database error detail from API responses and return a generic failure, retaining the detail in server-side logs only.
4. Reduce the application database role to the privileges the application requires; the role currently accepts statements the report feature never needs.

**How to verify the fix.** Re-run steps 2 to 4 above. The fix holds when the quotation mark returns a normal empty result set, the boolean conditions no longer change the row count, and the UNION constant does not appear in the output.

**Dependencies and cautions.** Parameterising the builder changes how saved filters are stored; existing saved filters containing SQL fragments will need migration or invalidation.

### REFERENCES


- CWE-89: Improper Neutralization of Special Elements used in an SQL Command — https://cwe.mitre.org/data/definitions/89.html
- OWASP WSTG v4.2 — Testing for SQL Injection (WSTG-INPV-05) — https://owasp.org/www-project-web-security-testing-guide/v42/
- OWASP SQL Injection Prevention Cheat Sheet — https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html


> **Validation boundary:** No data was extracted beyond the single constant shown, and the write test was rolled back within its transaction. Database host access, credential recovery and the full extent of the reachable schema were not established.

04 / TECHNICAL FINDING 2 OF 8

VEX-WEB-002 / SERVER-SIDE REQUEST HANDLING

HIGH

# Server-Side Request Forgery in the Webhook Validator

**Validation** Confirmed **HTTP method** POST **Authentication state** Authenticated (standard analyst) **Required role** analyst

**WSTG reference** WSTG-INPV-19 **OWASP Top 10 category** A10:2021 Server-Side Request Forgery

## AFFECTED ASSET

APPLICATION
https://portal.northwind.example

AFFECTED ENDPOINT
/api/v2/integrations/webhooks/validate

PARAMETER / INPUT LOCATION
callback_url (JSON body)

CVSS v3.1 · PRIMARY
**7.7** HIGH
`CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N`

CVSS v4.0 · ADDITIONAL
**8.3** HIGH
`CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:N/VA:N/SC:H/SI:N/SA:N`

CWE CLASSIFICATION — **CWE-918 — Server-Side Request Forgery (SSRF)**

## DESCRIPTION

Before a webhook is saved, the application verifies the destination by issuing a request to it and reporting whether it responded. The destination is taken from the submitted `callback_url` without restricting the address it resolves to.

Supplying an internal address caused the server to issue the request from its own network position and return the response body in the validation result. The link-local metadata address was reachable this way, as was an internal service that is not exposed externally.

This is the distinction the finding rests on: not that the URL was accepted, but that the server's response contained content only obtainable from inside the network boundary.

## IMPACT AND IMPLICATION

**Demonstrated.** The server performed attacker-directed requests to internal addresses and returned the response bodies, including the index of an internal metadata endpoint that is not routable from the internet.

**Potential, not demonstrated.** If credential material is exposed at any reachable internal endpoint, this path would disclose it. No credential was retrieved during testing and no internal service was authenticated to; that remains unestablished.

**Conditions required.** Any authenticated account holding the analyst role.

## REPRODUCTION AND VALIDATION

1. Sign in as a standard analyst account and open integration settings.
2. Submit a webhook validation request whose callback_url points at the link-local metadata address.
3. Observe that the validation result carries a response body rather than a connection error, and that its content is not reachable from the client network.
4. Repeat against an external address under your own control to confirm the request originates from the application's egress address, not the browser.

### EVIDENCE · EV-NWA-003

**REQUEST**

```
POST /api/v2/integrations/webhooks/validate HTTP/1.1
Host: portal.northwind.example
Content-Type: application/json
Cookie: nwa_session=[REDACTED]

{"callback_url": "http://169.254.169.254/latest/meta-data/"}
```

**OBSERVED RESPONSE**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "reachable": true,
  "status": 200,
  "body": "ami-id\nhostname\niam/\ninstance-id\nlocal-ipv4\n"
}
```

**EV-NWA-003 · HTTP exchange.** The validation result returns the body of a request the client could not have made. An accepted URL alone would not demonstrate SSRF; the returned internal content is what does.
Redacted artifact SHA-256 c69dc844d5e42ff05f6f4eb593ece74364ecdfb39476bf3e664f44bfa9f881f7.

### COMPARISON CONTROL

The same address requested directly from the testing client timed out, confirming the response was obtained from the server's network position rather than the tester's.

### RECOMMENDED REMEDIATION

1. Resolve the submitted URL and reject any address in private, link-local, loopback or otherwise internal ranges, re-checking after each redirect rather than only on the original URL.
2. Restrict the validator to HTTP and HTTPS schemes and to the default ports.
3. Return only whether validation succeeded, never the response body or headers from the destination.
4. Route outbound webhook traffic through an egress proxy with its own allow-list, so a bypass of the application check does not reach internal services.

**How to verify the fix.** Re-run step 2. The fix holds when the internal address is rejected before any request is made, the rejection survives a redirect to an internal address, and no destination response body appears in the result.

### REFERENCES


- CWE-918: Server-Side Request Forgery (SSRF) — https://cwe.mitre.org/data/definitions/918.html
- OWASP WSTG v4.2 — Testing for Server-Side Request Forgery (WSTG-INPV-19) — https://owasp.org/www-project-web-security-testing-guide/v42/
- OWASP SSRF Prevention Cheat Sheet — https://cheatsheetseries.owasp.org/cheatsheets/Server_Side_Request_Forgery_Prevention_Cheat_Sheet.html


> **Validation boundary:** Reachability of the metadata endpoint was established; retrieval of credentials, authentication to any internal service and onward movement were not attempted.

04 / TECHNICAL FINDING 3 OF 8

VEX-WEB-003 / BROKEN ACCESS CONTROL

HIGH

# Invoice Export Endpoint Accessible Without Authentication

**Validation** Confirmed **HTTP method** GET **Authentication state** Unauthenticated **Required role** none enforced **WSTG reference** WSTG-ATHZ-01

**OWASP Top 10 category** A01:2021 Broken Access Control

## AFFECTED ASSET

APPLICATION
https://portal.northwind.example

AFFECTED ENDPOINT
/api/v2/billing/invoices/export

PARAMETER / INPUT LOCATION
period (query string)

CVSS v3.1 · PRIMARY
**7.5** HIGH
`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N`

CVSS v4.0 · ADDITIONAL
**8.7** HIGH
`CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:N/VA:N/SC:N/SI:N/SA:N`

CWE CLASSIFICATION — **CWE-306 — Missing Authentication for Critical Function**

## DESCRIPTION

The billing area exposes a CSV export of invoices. The interactive invoice list requires a session, but the export route that backs the download button does not: it returns the same records to a request carrying no session cookie and no Authorization header.

The route appears to have been added for a scheduled export job and never placed behind the middleware that protects the rest of the billing area. It is not referenced from the unauthenticated interface, but it is directly reachable and requires no identifier to guess — the period parameter accepts a plain month.

## IMPACT AND IMPLICATION

**Demonstrated.** A request with no credentials returned a CSV containing invoice numbers, customer organisation names, line-item descriptions and amounts for the requested period.

**Potential, not demonstrated.** Any party who learns the route can retrieve billing data for arbitrary periods. Whether the same gap exists on sibling export routes was not established; only the invoice export was confirmed.

**Conditions required.** Network access to the application. No account is required.

## REPRODUCTION AND VALIDATION

1. Open a client with no cookies, no cached credentials and no Authorization header.
2. Request the invoice export route with a period parameter for a recent month.
3. Observe HTTP 200 and a CSV body rather than a redirect to sign-in or a 401.
4. Repeat with an authenticated session and confirm the two responses match, showing the route serves the same data in both states.

### EVIDENCE · EV-NWA-004

**REQUEST**

```
GET /api/v2/billing/invoices/export?period=2026-09 HTTP/1.1
Host: portal.northwind.example
Accept: text/csv
# no Cookie header, no Authorization header
```

**OBSERVED RESPONSE**

```
HTTP/1.1 200 OK
Content-Type: text/csv
Content-Disposition: attachment; filename="invoices-2026-09.csv"

invoice_no,organisation,issued,description,amount_usd
INV-20496,[REDACTED],2026-09-02,Analytics seats (40),12800.00
INV-20497,[REDACTED],2026-09-04,Data egress overage,1430.55
… 212 further rows
```

**EV-NWA-004 · HTTP exchange.** No session cookie and no Authorization header are present in the request, and the response is the data rather than a redirect to authentication.
Redacted artifact SHA-256 8b0a78a38257451ea18e8afdab4ce1527c4c86afa21ccb20d14d73683c934d6d.

### RECOMMENDED REMEDIATION

1. Place the export route behind the same authentication and authorisation middleware as the rest of the billing area, and make that middleware deny by default so a new route is protected unless it is explicitly exempted.
2. If a scheduled job needs the export, give it a service identity with a scoped credential rather than leaving the route anonymous.
3. Audit the remaining routes under /api/v2 for the same omission and add a test that asserts every billing route rejects an anonymous request.

**How to verify the fix.** Repeat step 2. The fix holds when the anonymous request returns 401 or 403 with no CSV body, while the authenticated request continues to succeed.

### REFERENCES

- CWE-306: Missing Authentication for Critical Function — https://cwe.mitre.org/data/definitions/306.html
- OWASP WSTG v4.2 — Testing Directory Traversal and Authorization Schema (WSTG-ATHZ-01) — https://owasp.org/www-project-web-security-testing-guide/v42/

> **Validation boundary:** Retrieval was confirmed for one period. The number of periods retained, and whether other export routes behave the same way, were not established.

04 / TECHNICAL FINDING 4 OF 8

VEX-WEB-004 / BROKEN ACCESS CONTROL

MEDIUM

# Invoice Records Readable Across Tenants via Sequential Identifier

**Validation** Confirmed **HTTP method** GET **Authentication state** Authenticated (standard analyst, Tenant A) **Required role** analyst

**WSTG reference** WSTG-ATHZ-04 **OWASP Top 10 category** A01:2021 Broken Access Control

## AFFECTED ASSET

APPLICATION
https://portal.northwind.example

AFFECTED ENDPOINT
/api/v2/billing/invoices/{invoice_id}

PARAMETER / INPUT LOCATION
invoice_id (path)

CVSS v3.1 · PRIMARY
**6.5** MEDIUM
`CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N`

CVSS v4.0 · ADDITIONAL
**7.1** HIGH
`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 CLASSIFICATION — **CWE-639 — Authorization Bypass Through User-Controlled Key**

**Note:** CVSS v3.1 rates this Medium (6.5) and CVSS v4.0 rates it High (7.1). The official report severity stays Medium, as the declared primary standard. The ratings are reported as calculated and are not averaged or reconciled.

## DESCRIPTION

An invoice is retrieved by a numeric identifier in the URL path. The handler authenticates the session but then loads the invoice by identifier alone, without checking that it belongs to the tenant the session is bound to.

Identifiers are allocated sequentially across all tenants, so a record belonging to another tenant sits at a predictable offset from a record the requester legitimately owns. Signed in to the Tenant A fixture account, a request for an identifier belonging to the Tenant B fixture account returned that record in full.

The interface never offers the other tenant's identifier, but nothing prevents it being requested directly.

## IMPACT AND IMPLICATION

**Demonstrated.** A Tenant A session retrieved a complete invoice belonging to Tenant B, including the organisation name, line items and amount. Two further sequential identifiers returned records from the same other tenant.

**Potential, not demonstrated.** Enumerating the identifier range would expose invoices across the customer base. The tests requested three identifiers only and did not attempt enumeration at scale; the total number of exposed records is not established.

**Conditions required.** Any authenticated account in any tenant, plus one identifier to extrapolate from.

## REPRODUCTION AND VALIDATION

1. Sign in as the Tenant A fixture account and open any invoice, noting its numeric identifier.
2. Request the adjacent identifier directly.
3. Observe HTTP 200 and an invoice whose organisation differs from the signed-in tenant.
4. Repeat as the Tenant B account against a Tenant A identifier to confirm the gap is symmetric and not a property of one tenant's configuration.

### EVIDENCE · EV-NWA-005

**REQUEST**

```
GET /api/v2/billing/invoices/20513 HTTP/1.1
Host: portal.northwind.example
Accept: application/json
Cookie: nwa_session=[REDACTED — Tenant A analyst]
```

**OBSERVED RESPONSE**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "invoice_no": "INV-20513",
  "tenant": "tenant-b-fixture",
  "organisation": "[REDACTED — other tenant]",
  "amount_usd": "7420.00",
  "status": "issued"
}
```

**EV-NWA-005 · HTTP exchange.** The session belongs to Tenant A; the returned record's organisation field shows Tenant B. The tenant mismatch in a 200 response is the finding.
Redacted artifact SHA-256 cf35b64598af6916298d262bc6ddd70006cc173347ca315c6fc9038fe06e7ecc.

### COMPARISON CONTROL

A request for an identifier that does not exist returned 404, confirming the handler distinguishes missing records from inaccessible ones — so the 200 above is an authorisation failure rather than an artefact of generic error handling.

### RECOMMENDED REMEDIATION

1. Scope the invoice lookup to the authenticated principal's tenant in the query itself, so a record outside that tenant cannot be loaded regardless of the identifier supplied.
2. Return 404 rather than 403 for records outside the caller's tenant, so the response does not confirm that an identifier exists elsewhere.
3. Replace sequential public identifiers with opaque per-tenant identifiers; this reduces the value of enumeration but is not a substitute for the ownership check.
4. Add automated cross-tenant tests to the billing suite using two fixture tenants.

**How to verify the fix.** Re-run steps 2 and 4. The fix holds when both directions return 404 and the caller's own invoices remain retrievable.

### REFERENCES


- CWE-639: Authorization Bypass Through User-Controlled Key — https://cwe.mitre.org/data/definitions/639.html
- OWASP WSTG v4.2 — Testing for Insecure Direct Object References (WSTG-ATHZ-04) — https://owasp.org/www-project-web-security-testing-guide/v42/
- OWASP Authorization Cheat Sheet — https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html


> **Validation boundary:** Three identifiers were requested. No bulk enumeration was performed, no record was modified, and the size of the exposed range is not established.

04 / TECHNICAL FINDING 5 OF 8

VEX-WEB-005 / AUTHENTICATION

MEDIUM

# No Rate Limiting on the Authentication Endpoint

**Validation** Confirmed **HTTP method** POST **Authentication state** Unauthenticated **WSTG reference** WSTG-ATHN-03

**OWASP Top 10 category** A07:2021 Identification and Authentication Failures

## AFFECTED ASSET

APPLICATION
https://portal.northwind.example

AFFECTED ENDPOINT
/api/v2/auth/login

PARAMETER / INPUT LOCATION
email, password (JSON body)

CVSS v3.1 · PRIMARY
**6.5** MEDIUM
`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N`

CVSS v4.0 · ADDITIONAL
**6.9** MEDIUM
`CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N`

CWE CLASSIFICATION — **CWE-307 — Improper Restriction of Excessive Authentication Attempts**

## DESCRIPTION

The login endpoint accepted 500 consecutive failed attempts against a single fixture account within approximately four minutes. No attempt was rejected for rate, no delay was introduced, no challenge appeared and the account was not locked.

Response timing did not separate a wrong password for a valid account from a wrong password for an account that does not exist, so this finding does not include user enumeration. The absence of any attempt limit is the issue on its own: it converts a weak or reused password into a reachable one.

## IMPACT AND IMPLICATION

**Demonstrated.** 500 failed attempts against one account were accepted at full speed with no throttling, lockout or challenge. The 500th attempt was served as quickly as the first.

**Potential, not demonstrated.** Credential stuffing and password spraying against this endpoint would be limited only by network capacity. No account was compromised during testing and no password was recovered.

**Conditions required.** Network access to the application.

## REPRODUCTION AND VALIDATION

1. Select a fixture account whose password is known to be wrong for this test.
2. Submit 500 sequential login requests with incorrect passwords.
3. Record the status code and latency of each response.
4. Confirm that no response indicates rate limiting, that latency does not grow, and that the account remains usable afterwards with its correct password.

EVIDENCE · EV-NWA-006

**AUTHENTICATION ATTEMPT HARNESS**

```
attempt    1/500  status 401  latency 94ms
attempt  100/500  status 401  latency 91ms
attempt  250/500  status 401  latency 97ms
attempt  500/500  status 401  latency 92ms

distinct status codes observed : {401}
responses indicating throttling: 0
lockout observed               : no
elapsed                        : 3m 58s
post-test login with correct password: 200 OK
```

**EV-NWA-006 · Test harness output.** The 500th failed attempt returns the same status and comparable latency as the first, and the account is immediately usable afterwards.
Redacted artifact SHA-256 cade5b37c41d6c1bb77e125637994b698de13618d6bc48a304a70af179483118.

## RECOMMENDED REMEDIATION

1. Apply per-account and per-source rate limiting to the login endpoint, with a progressive delay or temporary lockout after a small number of failures.
2. Track failures per account as well as per IP address, so a distributed attempt against one account is still limited.
3. Introduce a challenge after repeated failures rather than an indefinite lockout, which would otherwise allow a denial of service against a known account.
4. Alert on bursts of authentication failures across many accounts, which is the signature of spraying that per-account limits alone will not surface.

**How to verify the fix.** Repeat step 2. The fix holds when attempts are rejected or delayed well before the 500th, and a legitimate sign-in still succeeds afterwards.

**Dependencies and cautions.** Rate limiting must account for shared egress addresses, or customers behind a corporate NAT will be limited collectively.

## REFERENCES


- CWE-307: Improper Restriction of Excessive Authentication Attempts — https://cwe.mitre.org/data/definitions/307.html
- OWASP WSTG v4.2 — Testing for Weak Lock Out Mechanism (WSTG-ATHN-03) — https://owasp.org/www-project-web-security-testing-guide/v42/
- OWASP Credential Stuffing Prevention Cheat Sheet — https://cheatsheetseries.owasp.org/cheatsheets/Credential_Stuffing_Prevention_Cheat_Sheet.html


> **Validation boundary:** No password was guessed and no account was accessed. This finding establishes the absence of a control, not that any specific account is compromisable.

04 / TECHNICAL FINDING 6 OF 8

VEX-WEB-006 / INJECTION

MEDIUM

# Reflected Cross-Site Scripting in the Search Parameter

**Validation** Confirmed **HTTP method** GET **Authentication state** Unauthenticated **WSTG reference** WSTG-INPV-01

**OWASP Top 10 category** A03:2021 Injection

## AFFECTED ASSET

APPLICATION
https://portal.northwind.example

AFFECTED ENDPOINT
/search?q=

PARAMETER / INPUT LOCATION
q (query string)

CVSS v3.1 · PRIMARY
**6.1** MEDIUM
`CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N`

CVSS v4.0 · ADDITIONAL
**5.1** MEDIUM
`CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:A/VC:L/VI:L/VA:N/SC:L/SI:L/SA:N`

CWE CLASSIFICATION — **CWE-79 — Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')**

## DESCRIPTION

The search results page echoes the submitted query back to the user inside the results heading. The value is written into the HTML without encoding, so markup in the query becomes part of the document.

A script payload supplied in the query executed in the browser under the application's origin. This was confirmed in a real browser context rather than inferred from the response body: the reflection alone would not establish execution, and a payload that is escaped on output would still appear in the raw response.

The page is reachable without authentication, so a crafted link can be delivered to a signed-in user without the sender holding any account.

## IMPACT AND IMPLICATION

**Demonstrated.** Script supplied in the query string executed in the browser in the application's origin, confirmed by the script writing an observable marker into the document.

**Potential, not demonstrated.** In a signed-in victim's browser this position could read page content and issue requests as that user. Session cookie theft was not demonstrated — see the limitation below — and no victim session was accessed.

**Conditions required.** The target must open a crafted link. No account is required to craft or send it.

## REPRODUCTION AND VALIDATION

1. Request the search page with a query containing a benign HTML tag.
2. Confirm in the raw response that the tag is reflected without encoding.
3. Load the same URL in a browser with a script payload that writes a marker element.
4. Confirm the marker is present in the rendered DOM, establishing execution rather than reflection alone.

### EVIDENCE · EV-NWA-007

**REQUEST**

```
GET /search?q=%3Csvg%2Fonload%3Dwindow.__vx%3D1%3E HTTP/1.1
Host: portal.northwind.example
Accept: text/html
```

**OBSERVED RESPONSE**

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8

<h1 class="results-title">
  Results for <svg/onload=window.__vx=1>
</h1>
<p>0 matching datasets</p>
```

**EV-NWA-007 · HTTP exchange.** The query is written into the heading without encoding. Reflection is a necessary condition; execution was confirmed separately in a browser.
Redacted artifact SHA-256 0ac6c38100a3ab7aa4a09985a8c8e20369c2dc9902a5b9f9e6c87c32f88a2b37.

### EVIDENCE · EV-NWA-008

**HEADLESS BROWSER TRACE**

```
navigate  https://portal.northwind.example/search?q=<svg/onload=…>
load event fired
evaluate  typeof window.__vx   -> "number"
evaluate  window.__vx          -> 1
origin    https://portal.northwind.example
console   (no errors, no CSP violation reported)
```

**EV-NWA-008 · Browser observation.** Execution confirmed in a real rendering context: the payload set a marker that was read back from the live DOM after load.
Redacted artifact SHA-256 2e6085e5eb40e7e952c4beb65269ae44b30717d8dee285191f2908eed00d4374.

### RECOMMENDED REMEDIATION

1. Encode the reflected query for the HTML context at the point of output; prefer the template engine's contextual escaping over manual replacement.
2. Serve a Content-Security-Policy that disallows inline script execution, so an escaping mistake elsewhere does not become an executable one.
3. Review other reflections of request input across the search and reporting pages, which share this template.

**How to verify the fix.** Repeat steps 3 and 4. The fix holds when the marker is absent from the DOM and the payload appears as visible text in the heading.

### REFERENCES


- CWE-79: Improper Neutralization of Input During Web Page Generation — https://cwe.mitre.org/data/definitions/79.html
- OWASP WSTG v4.2 — Testing for Reflected Cross Site Scripting (WSTG-INPV-01) — https://owasp.org/www-project-web-security-testing-guide/v42/
- OWASP Cross Site Scripting Prevention Cheat Sheet — https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html


> **Validation boundary:** The session cookie carries HttpOnly, so this payload cannot read it. Script execution in the user's origin was demonstrated; session theft was not, and is not claimed.

04 / TECHNICAL FINDING 7 OF 8

VEX-WEB-007 / SECURITY MISCONFIGURATION

MEDIUM

# Application Dependency Manifest Publicly Accessible

**Validation** Confirmed **HTTP method** GET **Authentication state** Unauthenticated **WSTG reference** WSTG-CONF-04

**OWASP Top 10 category** A05:2021 Security Misconfiguration

## AFFECTED ASSET

| APPLICATION | AFFECTED ENDPOINT |
|---|---|
| https://portal.northwind.example | /package-lock.json |

| CVSS v3.1 · PRIMARY | CVSS v4.0 · ADDITIONAL |
|---|---|
| 5.3 MEDIUM | 6.9 MEDIUM |
| CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N | 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 CLASSIFICATION** — **CWE-552 — Files or Directories Accessible to External Parties**

## DESCRIPTION

The dependency lock file and a source map are served from the application root. Both are build artefacts: they were deployed alongside the application rather than being excluded from the published directory.

The lock file enumerates every direct and transitive dependency with its exact version. That does not itself constitute a vulnerability, and this finding makes no claim that any listed dependency is exploitable in this deployment — establishing that would require confirming the affected code path is reachable. What it does provide is an accurate, effort-free inventory for selecting an attack, including internal package names that disclose the organisation's own module naming.

## IMPACT AND IMPLICATION

**Demonstrated.** Anonymous retrieval of the complete dependency manifest with exact versions, and of a source map exposing original module paths and names.

**Potential, not demonstrated.** The inventory shortens reconnaissance for an attacker selecting a known vulnerability in a listed component. No dependency was tested for exploitability in this deployment.

**Conditions required.** Network access to the application.

## REPRODUCTION AND VALIDATION

1. Request the lock file path directly with no credentials.
2. Confirm HTTP 200 with JSON content rather than 404.
3. Repeat for the source map referenced by the application bundle.

### EVIDENCE · EV-NWA-009

**REQUEST**

```
GET /package-lock.json HTTP/1.1
Host: portal.northwind.example
Accept: */*
```

**OBSERVED RESPONSE**

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 418233

{
  "name": "northwind-portal",
  "lockfileVersion": 3,
  "packages": {
    "node_modules/@northwind/billing-ui": { "version": "4.2.1" },
    "node_modules/@northwind/report-engine": { "version": "2.9.0"
},
    … 1,184 further packages with exact versions
  }
}
```

**EV-NWA-009 · HTTP exchange.** The manifest is served to an anonymous request. The excerpt is truncated: the file lists the full dependency tree with exact versions. Redacted artifact SHA-256 a83c15995f00e163edc18cc6256098a335bc68c36598e0887963f5cbac2147bc.

### RECOMMENDED REMEDIATION

1. Exclude build artefacts from the published directory: lock files, source maps, CI configuration and environment templates should not be deployed to the web root.
2. Deny requests for these paths at the web server or CDN as a second control, so a future build that reintroduces them is not immediately exposed.
3. Add a deployment check that fails the release when a known build artefact is present in the published output.

**How to verify the fix.** Repeat steps 1 to 3. The fix holds when each path returns 404 and the application continues to function, confirming nothing at runtime depended on them.

### REFERENCES

- CWE-552: Files or Directories Accessible to External Parties — https://cwe.mitre.org/data/definitions/552.html
- OWASP WSTG v4.2 — Review Old Backup and Unreferenced Files (WSTG-CONF-04) — https://owasp.org/www-project-web-security-testing-guide/v42/

> **Validation boundary:** No listed dependency was tested for a known vulnerability, and no claim is made that any is exploitable in this deployment.

04 / TECHNICAL FINDING 8 OF 8

VEX-WEB-008 / SESSION MANAGEMENT

LOW

# Session Cookie Issued Without the Secure Attribute

**Validation** Confirmed **HTTP method** POST **Authentication state** Authenticated (any role) **WSTG reference** WSTG-SESS-02

**OWASP Top 10 category** A05:2021 Security Misconfiguration

## AFFECTED ASSET

APPLICATION
https://portal.northwind.example

AFFECTED ENDPOINT
/api/v2/auth/login (Set-Cookie)

PARAMETER / INPUT LOCATION
nwa_session cookie

CVSS v3.1 · PRIMARY
**3.7** LOW
CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N

CVSS v4.0 · ADDITIONAL
**6.3** MEDIUM
CVSS:4.0/AV:N/AC:H/AT:P/PR:N/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N

CWE CLASSIFICATION — **CWE-614 — Sensitive Cookie in HTTPS Session Without 'Secure' Attribute**

**Note:** CVSS v3.1 rates this Low (3.7) and CVSS v4.0 rates it Medium (6.3). The official report severity stays Low, as the declared primary standard. The ratings are reported as calculated and are not averaged or reconciled.

## DESCRIPTION

The session cookie is issued with HttpOnly and SameSite=Lax but without Secure. The browser will therefore include it on any plaintext HTTP request to the domain, even though the application itself redirects HTTP to HTTPS.

The redirect does not remove the exposure: the cookie is sent on the initial plaintext request, before the redirect is received. HSTS is also absent, so a browser that has not previously visited the site has nothing instructing it to use HTTPS for that first request.

This is a transport-hardening weakness rather than a directly exploitable one. Exploitation requires a position on the network path at the moment a plaintext request is made, which is why its access complexity is rated high.

## IMPACT AND IMPLICATION

**Demonstrated.** The Set-Cookie header omits Secure, and a plaintext request to the domain included the session cookie before the redirect to HTTPS was issued.

**Potential, not demonstrated.** A party positioned on the network path could capture the session cookie from a plaintext request and reuse it. No such capture was performed and no session was hijacked during this assessment.

**Conditions required.** A network position between the user and the application, and a plaintext request from a browser that has not pinned HTTPS for the domain.

## REPRODUCTION AND VALIDATION

1. Authenticate and capture the Set-Cookie response header.
2. Confirm that Secure is absent while HttpOnly and SameSite are present.
3. Issue a plaintext HTTP request to the domain with the cookie jar from step 1 and observe the cookie in the request before the redirect.
4. Confirm no Strict-Transport-Security header is present on the HTTPS response.

### EVIDENCE · EV-NWA-010

**REQUEST**

```
POST /api/v2/auth/login HTTP/1.1
Host: portal.northwind.example
Content-Type: application/json

{"email": "analyst@fixture.example", "password": "[REDACTED]"}
```

**OBSERVED RESPONSE**

```
HTTP/1.1 200 OK
Set-Cookie: nwa_session=[REDACTED]; Path=/; HttpOnly;
SameSite=Lax
Content-Type: application/json
# no Secure attribute
# no Strict-Transport-Security header on this response
```

**EV-NWA-010 · HTTP exchange.** HttpOnly and SameSite are set; Secure is not. The absence of Strict-Transport-Security on the same response is shown because it removes the mitigation that would otherwise prevent the plaintext request.
Redacted artifact SHA-256 3db517e7c8c3dfecdab46fb2e3574845b0a9a5bb8b69942123e3bf0fa56e303a.

### RECOMMENDED REMEDIATION

1. Add the Secure attribute to the session cookie so the browser never transmits it over plaintext.
2. Serve Strict-Transport-Security with a meaningful max-age on all HTTPS responses, so a browser that has seen the site once will not issue a plaintext request again.
3. Consider the __Host- cookie name prefix, which browsers accept only when Secure and a root path are set, making a regression visible immediately.

**How to verify the fix.** Repeat steps 1 to 4. The fix holds when Secure is present on the Set-Cookie header, the plaintext request carries no session cookie, and HSTS is returned.

**Dependencies and cautions.** Enable HSTS with a short max-age first and raise it once confirmed, since the policy is cached by browsers and a premature long max-age is difficult to undo.

### REFERENCES

- CWE-614: Sensitive Cookie in HTTPS Session Without 'Secure' Attribute — https://cwe.mitre.org/data/definitions/614.html
- OWASP WSTG v4.2 — Testing for Cookies Attributes (WSTG-SESS-02) — https://owasp.org/www-project-web-security-testing-guide/v42/
- OWASP Session Management Cheat Sheet — https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html

> **Validation boundary:** No session cookie was intercepted and no session was reused. The finding establishes that the attribute is absent, not that interception has occurred.

05 / REMEDIATION

# From detection to resolution.

Grouped by remediation priority. Priority weighs verified technical impact, exposure, prerequisites and evidence confidence — it is a separate judgement from the CVSS severity band, and it is recorded with its rationale.

## IMMEDIATE · SUGGESTED WITHIN 7 DAYS

severity high; confirmed read and write access to the application database; reachable by the default analyst role; severity high; no authentication required; confirmed disclosure of commercial records to an anonymous caller; severity high; reaches internal network positions from an externally reachable application; low privilege requirement; severity medium under v3.1; cross-tenant disclosure in a multi-tenant product and a CVSS v4.0 rating of High raise the action order above the severity band

### 01 Parameterise the saved-report query and reduce the database role

Bind the filter value instead of concatenating it, resolve column and sort inputs against an allow-list, and remove database error detail from API responses.

Addresses VEX-WEB-001 · owner: Application team · verify by: injected quote returns an empty result set, not a parser error

### 02 Restrict outbound destinations in the webhook validator

Reject internal address ranges after every redirect, limit schemes and ports, and stop returning the destination's response body.

Addresses VEX-WEB-002 · owner: Platform team · verify by: internal address rejected before any request is issued

### 03 Make billing authorisation deny-by-default and tenant-scoped

Place the export route behind the billing middleware, and scope every invoice lookup to the authenticated principal's tenant inside the query rather than after it. One change in the data-access layer closes both the anonymous export and the cross-tenant read.

Addresses VEX-WEB-003, VEX-WEB-004 · owner: Application team · verify by: anonymous export returns 401; cross-tenant identifier returns 404

## HIGH · SUGGESTED WITHIN 30 DAYS

severity medium; unauthenticated and internet-facing; severity medium; unauthenticated and internet-facing; no content security policy limits the consequence

### 01 Throttle authentication attempts per account and per source

Apply progressive delay or temporary lockout, count failures per account as well as per address, and alert on failure bursts across many accounts.

Addresses VEX-WEB-005 · owner: Application team · verify by: attempts rejected or delayed well before the 500th

### 02 Encode reflected output and add a content security policy

Apply contextual escaping to the search heading and serve a policy that disallows inline script, so a future escaping mistake is not executable.

Addresses VEX-WEB-006 · owner: Application team · verify by: payload renders as visible text; no marker in the DOM

## PLANNED · SUGGESTED WITHIN 90 DAYS

severity low; requires a network position, but the missing HSTS header removes the control that would otherwise prevent the plaintext request; severity medium; information exposure without direct access impact

## 01 Remove build artefacts from the published directory

Exclude lock files and source maps from deployment, deny those paths at the edge, and fail the release when they reappear in the output.

Addresses VEX-WEB-007 · owner: Platform team · verify by: each path returns 404 and the application still functions

## 02 Harden session cookie transport

Set Secure on the session cookie and serve Strict-Transport-Security, raising max-age once the short value is confirmed.

Addresses VEX-WEB-008 · owner: Platform team · verify by: plaintext request carries no session cookie; HSTS returned

**A change is not proof.** Suggested timeframes are Vextrix guidance, not a contractual or regulatory deadline. A remediation is only established as effective once it has been retested against fresh evidence.

06 / CONCLUSION

# What this establishes, and what it does not.

The assessment established eight confirmed weaknesses in the staging replica of the Northwind Analytics portal. Three are reachable with no account; the two most serious give an ordinary analyst account access well beyond the role's intended boundary — direct database access in one case and a route into the internal network in the other.

The interactive interface is not the weak point. Every confirmed finding sits on an API route or a query path added for automation and export, which did not inherit the authorisation and input-handling rules applied to the pages. That is a process observation as much as a technical one, and it is the reason a small number of changes closes most of the report.

## CONSEQUENCES SUPPORTED BY THE EVIDENCE

- An analyst account can read and modify application data directly, which removes any assurance the application layer's own permission checks provide.
- Commercial records are retrievable without authentication, and records belonging to one tenant are retrievable by another.
- The application can be directed to make requests from inside the network boundary and to return what it receives.

## HIGHEST-PRIORITY REMEDIATION

Close VEX-WEB-001 and VEX-WEB-002 first — they grant the widest access and share a root cause. Then apply the deny-by-default, tenant-scoped change in the data-access layer, which closes VEX-WEB-003 and VEX-WEB-004 together.

## RESIDUAL UNCERTAINTY

- File upload handling was never exercised: the service was unavailable throughout the window. It is untested, not clean.
- Multi-factor enrolment and recovery are disabled in staging and were not assessed.
- Testing was performed against a staging replica. Where production configuration differs, these results may not transfer, in either direction.
- The extent of the data reachable through VEX-WEB-001 and VEX-WEB-004 was deliberately not explored; both were demonstrated with the minimum evidence needed to confirm them.

### RECOMMENDED NEXT VERIFICATION STEP

Remediate the three immediate-priority items and request the included retest. The retest replays the tests behind each confirmed finding and records Fixed, Still Present or Could Not Verify against fresh evidence — a code change alone does not establish that a finding is closed. Re-assessing file upload and multi-factor once available in a test environment would close the two known coverage gaps.

This assessment does not establish that the web application is secure. It records what was tested, what was observed and what could not be established within the authorised scope, window and coverage.

07 / SCORING POLICY

# Severity is technical. Priority is contextual.

One classification standard governs this report. These definitions describe exactly the labels used in the preceding sections — nothing here is generic filler.

## OFFICIAL REPORT SEVERITY — CVSS V3.1 BASE

| BAND | CVSS V3.1 BASE | USED FOR |
|---|---|---|
| CRITICAL | 9.0 – 10.0 | Severity badges, summary counts, the register ordering and the severity distribution. Calculated by the cvss 3.6 (FIRST CVSS v3.1 and v4.0 specifications), never entered by hand. |
| HIGH | 7.0 – 8.9 | |
| MEDIUM | 4.0 – 6.9 | |
| LOW | 0.1 – 3.9 | |
| INFO | None / not scored | |

An informational observation is not automatically CVSS 0.0. Hardening observations carry no CVSS vector at all rather than an invented zero.

### CVSS V4.0

**Calculated independently**

Its own vector, score and qualitative rating. Never converted from v3.1, never averaged with it, never used to silently raise or lower the official severity.

### REMEDIATION PRIORITY

**A separate decision**

Derived from severity plus exposure, prerequisites and evidence confidence. It may sit above or below the severity band; each finding records why.

## CWE CLASSIFICATION

Each confirmed finding carries the most precise applicable weakness from the MITRE CWE List (4.17), with its official name. Where the evidence does not support a precise mapping the finding is held for mapping review rather than assigned a broad or unrelated weakness. CWE classifies the weakness; it is not a severity score.

## VALIDATION STATES

| STATE | MEANING |
|---|---|
| Confirmed | **Independent validation demonstrated the condition the finding claims. Only confirmed findings are released and counted.** |
| Candidate | **Observed but not yet validated. Visible internally; never published as a vulnerability.** |
| Inconclusive | **Validation ran but could not settle the question. Recorded as a coverage gap, not as a negative result.** |
| Rejected | **Validation contradicted the hypothesis. Retained internally for traceability.** |

08 / REFERENCES

# Standards-led. Evidence-controlled.

TESTING GUIDANCE
**OWASP Web Security Testing Guide v4.2**

The primary technical reference for this assessment type.
https://owasp.org/www-project-web-security-testing-guide/v42/

RISK TAXONOMY
**OWASP Top 10:2021**

Risk categories cited on each finding.
https://owasp.org/Top10/

VERIFICATION STANDARD
**OWASP ASVS 5.0.0**

Control expectations referenced in remediation guidance.
https://github.com/OWASP/ASVS/tree/v5.0.0

SCORING
**FIRST CVSS v3.1 and v4.0 specifications**

Both scores are calculated from their vectors, never converted between versions.
https://www.first.org/cvss/

WEAKNESS CLASSIFICATION
**MITRE CWE List**

Each finding carries the most precise applicable weakness with its official name.
https://cwe.mitre.org/

PROCESS
**NIST SP 800-115**

Technical guide to information security testing and assessment.
https://csrc.nist.gov/pubs/sp/800/115/final

## APPENDIX A · EVIDENCE INDEX

| ID | KIND | COLLECTED | METHOD | REDACTED SHA-256 |
|---|---|---|---|---|
| EV-NWA-001 | **HTTP exchange** | 2026-11-12T11:20:00+08:00 | mediated tool call | 6864c9351cc16165755fb6ab7f758731d546951cc82c3618fc610985e5b44ffd |
| EV-NWA-002 | **HTTP exchange** | 2026-11-12T11:20:00+08:00 | mediated tool call | 1fa9f91b06d76e3d58a45f0dffb46b859c4fd55b402ad934a205381b2e990d1a |
| EV-NWA-003 | **HTTP exchange** | 2026-11-12T11:20:00+08:00 | mediated tool call | c69dc844d5e42ff05f6f4eb593ece74364ecdfb39476bf3e664f44bfa9f881f7 |
| EV-NWA-004 | **HTTP exchange** | 2026-11-12T11:20:00+08:00 | mediated tool call | 8b0a78a38257451ea18e8afdab4ce1527c4c86afa21ccb20d14d73683c934d6d |
| EV-NWA-005 | **HTTP exchange** | 2026-11-12T11:20:00+08:00 | mediated tool call | cf35b64598af6916298d262bc6ddd70006cc173347ca315c6fc9038fe06e7ecc |
| EV-NWA-006 | **Test harness output** | 2026-11-12T11:20:00+08:00 | mediated tool call | cade5b37c41d6c1bb77e125637994b698de13618d6bc48a304a70af179483118 |
| EV-NWA-007 | **HTTP exchange** | 2026-11-12T11:20:00+08:00 | mediated tool call | 0ac6c38100a3ab7aa4a09985a8c8e20369c2dc9902a5b9f9e6c87c32f88a2b37 |

| ID | KIND | COLLECTED | METHOD | REDACTED SHA-256 |
|---|---|---|---|---|
| EV-NWA-008 | **Browser observation** | 2026-11-12T11:20:00+08:00 | mediated tool call | 2e6085e5eb40e7e952c4beb65269ae44b30717d8dee285191f2908eed00d4374 |
| EV-NWA-009 | **HTTP exchange** | 2026-11-12T11:20:00+08:00 | mediated tool call | a83c15995f00e163edc18cc6256098a335bc68c36598e0887963f5cbac2147bc |
| EV-NWA-010 | **HTTP exchange** | 2026-11-12T11:20:00+08:00 | mediated tool call | 3db517e7c8c3dfecdab46fb2e3574845b0a9a5bb8b69942123e3bf0fa56e303a |

Hashes are of the redacted artifacts distributed with this report. Original evidence is retained in controlled internal storage and is not distributed.

## REPORT INTEGRITY

| | |
|---|---|
| Report reference | **VX-SAMPLE-WEB-001** |
| Version | **1** |
| Snapshot SHA-256 | **d384a5c07e91b57e48d99924716f64eb7f6b30ab7d7b460b3617ba6950c43dd0** |
| Report schema | **3.0** |
| Template | **v3.1** |
| Scoring policy | **1.0** |

This document was generated from a frozen report snapshot. The PDF and HTML deliverables are two presentations of that one snapshot; a change to the underlying assessment data invalidates the candidate and requires a new version.

This report is confidential and prepared for the named recipient. It describes the state of the assessed environment during the recorded testing window only. Security assessment is point-in-time and bounded by the authorised scope: the absence of a finding is not proof that a weakness does not exist. Redistribute only to parties with a legitimate need to act on it.