



# Penetration Test Report

Northwind Cloud Billing — staging environment — 6-10 October 2026

## This is a sample.

The client, application, hostnames and findings below are fictional and illustrate the format of a real report. No real system was tested.

|                |                                                     |
|----------------|-----------------------------------------------------|
| Client         | Northwind Billing Ltd. (fictional)                  |
| System tested  | Northwind Cloud Billing, staging environment        |
| Test window    | 6-10 October 2026 (5 working days)                  |
| Test type      | Grey-box web application and API penetration test   |
| Tester         | Vallum Security — Ismail Regragui                   |
| Classification | Confidential — for the client and its auditors only |

## Executive summary

The platform's overall risk is **High**, because any logged-in customer can read and change another customer's invoices, and a low-privileged support user can take over user management. We found 2 high, 4 medium, 3 low and 2 informational issues. The core platform is well built, and most findings come from missing authorization checks on specific endpoints, which can be fixed within days.

### Top three actions

1. Enforce tenant ownership checks on every invoice endpoint (F-01).
2. Restrict user-management endpoints to administrator roles on the server (F-02).
3. Rate-limit login and password reset, and end all sessions when a password changes (F-04, F-05).

### Results at a glance



### Business impact

- **Customer data exposure.** F-01 lets any customer read other customers' invoices, which contain names, addresses and amounts. This is a confidentiality failure between tenants and may qualify as a personal data incident under GDPR.
- **Loss of administrative control.** F-02 and F-03 together let an attacker holding an ordinary customer account become an administrator.

- **Audit and customer assurance.** Until F-01 and F-02 are fixed, the platform should not be described to auditors or customers as having effective tenant isolation or access control.

## Scope and methodology

### In scope

- Customer web app (Backoffice, Video Call Widget, Video Player)
- Public REST API (OpenAPI documented) and internal APIs
- Authorization across 3 roles, 2 test tenants

### Rules of engagement

- Testing from a fixed source IP, 08:00-20:00 UTC
- Request rate below 20 req/s per host
- Critical findings reported within 24 hours
- Authorized in writing (ref. NW-LOA-2026-009)

### Out of scope

- Production environment and any real customer data
- Third-party services, denial-of-service testing, social engineering

### Methodology

Manual testing following the OWASP Web Security Testing Guide, the OWASP API Security Top 10 (2023) and NIST SP 800-115, in five phases: reconnaissance, authentication and session, authorization, input handling and logic, configuration and reporting. Severity follows the CVSS v3.1 base score.

## Findings at a glance

| ID   | FINDING                                        | SEVERITY | CVSS | CWE |
|------|------------------------------------------------|----------|------|-----|
| F-02 | Support Agent can call administrator endpoints | High     | 8.8  | 285 |
| F-01 | Cross-tenant read and write access to invoices | High     | 8.1  | 639 |
| F-04 | No rate limiting on login and password reset   | Medium   | 6.5  | 307 |
| F-03 | Stored cross-site scripting in ticket comments | Medium   | 5.4  | 79  |
| F-06 | Blind SSRF in webhook settings                 | Medium   | 5.0  | 918 |
| F-05 | Sessions stay valid after a password change    | Medium   | 4.2  | 613 |
| F-07 | Error responses disclose stack traces          | Low      | 3.7  | 209 |
| F-08 | No Content Security Policy                     | Low      | 3.1  | 693 |
|      |                                                |          |      |     |

| ID   | FINDING                                  | SEVERITY | CVSS | CWE |
|------|------------------------------------------|----------|------|-----|
| F-09 | 30-day session lifetime, no idle timeout | Low      | 3.1  | 613 |

---

## Attack chain: from a support ticket to cross-tenant data theft

Several findings are more serious in combination. This scenario was demonstrated on staging using only an ordinary Customer User account in Tenant A.

1. **Plant the script (F-03).** The attacker opens a support ticket with a comment containing a script payload.
2. **Agent opens the ticket.** The script runs in the Support Agent's browser, inside the support dashboard.
3. **Escalate (F-02).** The script calls the administrator user-management API with the agent's session, which succeeds because the API does not check roles. It creates an administrator account for the attacker.
4. **Stay in (F-05).** Resetting the passwords of the affected accounts does not end sessions the attacker already holds, so access continues until sessions are revoked by hand.
5. **Reach other tenants (F-01).** Using any valid session, the attacker iterates invoice IDs and reads, or voids, invoices belonging to every other customer.

Fixing F-01 and F-02 breaks the chain at its most damaging steps. Fixing F-03 removes the entry point.

---

## Findings

## Cross-tenant access to invoices

CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N · CWE-639 · OWASP API1 – Broken Object Level Authorization

### DESCRIPTION

The invoice endpoints check that the user is logged in but not that the invoice belongs to the user's organization. Invoice IDs are sequential integers, so a customer can read or edit any other customer's invoices.

### EVIDENCE

```
GET /api/v2/invoices/1043
Authorization: Bearer [Tenant A, Customer User]

200 OK {"org_id":"org_b81","customer":"Contoso
Retail","total":"4820.00","status":"open"}
```

### IMPACT

Any customer can view other customers' billing data and alter their invoices. This breaches data confidentiality between tenants and may be a reportable personal data incident.

### REMEDIATION

Check the invoice's organization against the caller's organization in the data layer for every read and write. Replace sequential IDs with random identifiers as defense in depth, and add automated tests that call each endpoint with another tenant's IDs.

## Support Agent can use administrator functions

CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H · CWE-285 · OWASP API5 – Broken Function Level Authorization

### DESCRIPTION

The web interface hides user management from the Support Agent role, but the underlying API endpoints only check that the caller is logged in. The restriction exists in the interface alone.

### EVIDENCE

```
POST /api/v2/admin/users
Authorization: Bearer [Support Agent]
{"email":"tester@example.test","role":"admin"}

201 Created {"id":"usr_9f3","role":"admin","org_id":"org_a42"}
```

### IMPACT

A support account, which many staff share and which is a common phishing target, can create administrators, reset other users' access, and take over the organization.

### REMEDIATION

Enforce role checks on the server for every administrative endpoint, using a central permission layer instead of per-endpoint code. Review all endpoints under /admin/ and add tests that call each one with every role.

## No rate limiting on login and password reset

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N · CWE-307 / CWE-204 · OWASP API2 – Broken Authentication

### DESCRIPTION

The login and password-reset endpoints accept unlimited attempts, with no lockout, delay, or CAPTCHA. Error messages also differ for existing and unknown email addresses.

### EVIDENCE

```
500 login attempts for one account, ~100 seconds:  
500 x HTTP 401 (no 429 responses, no lockout, no added delay)
```

### IMPACT

Attackers can run credential-stuffing attacks with passwords leaked elsewhere and build a list of valid customer accounts.

### REMEDIATION

Apply per-account and per-IP rate limits with progressive delays, return identical responses for known and unknown emails, and offer multi-factor authentication.

## Stored cross-site scripting in ticket comments

CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:L/A:N · CWE-79 · OWASP A03 – Injection

### DESCRIPTION

Ticket comments are stored and later rendered in the support dashboard without output encoding. A customer can place a script in a comment that runs in a support agent's browser when the ticket is opened.

### EVIDENCE

```
POST /api/v2/tickets/771/comments  
{ "body": "<img src=x onerror=alert(document.domain)>" }  
→ dialog shows staging.northwind-billing.test when agent opens ticket 771
```

### IMPACT

An attacker can act as the support agent inside the dashboard. Combined with F-02, this allows account takeover through a single ticket.

### REMEDIATION

Encode output by default, add a Content Security Policy that blocks inline scripts, and sanitize rich-text input with an allow-list library.

## Blind server-side request forgery in webhook settings

CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:L/I:N/A:N · CWE-918 · OWASP API7 – Server-Side Request Forgery

### DESCRIPTION

Customer Admins can register a webhook URL, and the server sends a test request to it. The URL is only checked for the https scheme, so the server can be made to connect to internal addresses. The response body is not shown, so the issue is blind.

### EVIDENCE

```
POST /api/v2/webhooks/test
{"url":"https://10.0.4.12:6379/"}
200 OK {"result":"connection_ok","elapsed_ms":14}
```

### IMPACT

An attacker with a customer administrator account can map the internal network and discover services that were never meant to be reachable.

### REMEDIATION

Resolve the hostname on the server and reject private, loopback and link-local ranges after DNS resolution. Prefer an allow-list of destinations or a dedicated outbound proxy, and return one generic error for all failures.

## Sessions stay valid after a password change

CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:L/I:L/A:N · CWE-613 · OWASP A07 – Identification and Authentication Failures

### DESCRIPTION

Session tokens are not revoked when a user changes their password. Anyone who already holds a stolen token keeps access after the legitimate user has reset the password.

### EVIDENCE

```
GET /api/v2/account
Authorization: Bearer [token issued before password change]
200 OK {"id":"usr_204","email":"user@example.test"}
```

### IMPACT

Password resets do not remove an attacker who has already stolen a session, which weakens incident response.

### REMEDIATION

Store a session version per user and increment it on password change, logout-everywhere, and role change. Reject tokens issued before the current version.

## Lower-severity findings

**F-07 — Error responses disclose stack traces**  
(Low, 3.7)

**F-09 — 30-day sessions, no idle timeout** (Low, 3.1)

|                                                                                                                                                       |                                                                                                                               |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| Malformed input returns internal file paths and a database table name. <i>Fix:</i> generic error with a correlation ID.                               | Including for administrators. <i>Fix:</i> 30-minute idle timeout, 12-hour absolute lifetime for admins.                       |
| <b>F-08 — No Content Security Policy</b> (Low, 3.1)<br>Would have limited the impact of F-03. <i>Fix:</i> deploy a policy disallowing inline scripts. | <b>I-01 / I-02 — Informational</b><br>Outdated front-end library (jQuery 2.2.4); administrative actions are not audit-logged. |

## Positive findings

- Database queries use parameterized statements — no SQL injection was found in any tested endpoint
- Passwords use a modern slow hashing algorithm, and TLS is configured correctly on all tested hosts
- File uploads are restricted by type and size, and stored outside the web root
- Security headers (HSTS, X-Content-Type-Options) are present, and cookies use Secure and HttpOnly flags

## Remediation roadmap

| PRIORITY | FINDING       | ACTION                                                             | TARGET       |
|----------|---------------|--------------------------------------------------------------------|--------------|
| 1        | F -01         | Enforce tenant ownership in the data layer; add cross-tenant tests | 7 days       |
| 2        | F -02         | Central server-side role checks on all /admin/ endpoints           | 7 days       |
| 3        | F -04         | Rate limits, uniform responses, MFA option                         | 30 days      |
| 4        | F -03 / F -08 | Output encoding and Content Security Policy                        | 30 days      |
| 5        | F -06         | Resolve-then-validate outbound requests; egress allow-list         | 30 days      |
| 6        | F -05 / F -09 | Session versioning; revoke on password change                      | 30 days      |
| 7        | F -07 / I -01 | Generic errors; dependency upgrade                                 | Next release |
| 8        | I -02         | Audit log for administrative actions                               | Next quarter |

Fixes for high-severity findings will be retested within 30 days of this report at no additional charge, with an updated report and attestation letter for your auditors.

## Appendices

### A. Tools used

Burp Suite Professional, custom Python scripts to replay requests across roles and tenants, ffuf for endpoint discovery (rate-limited), a self-hosted callback server for out-of-band confirmation in F-06.

### B. Limitations

This is a point-in-time assessment of the staging environment, performed over five working days. It does not prove the absence of other vulnerabilities, and production may differ from staging.

---

**Vallum Security — Ismail Regragui**

Rabat, Morocco · [YOUR EMAIL]

NW-PT-2026-014 · v1.0