CETONIX

SECURITY ASSURANCE · OFFENSIVE SECURITY PRACTICE

CONFIDENTIAL · TLP:AMBER+STRICT

API PENETRATION TEST REPORT

REST and GraphQL · OWASP API Security Top 10

Authenticated, Multi-Role and Multi-Tenant Testing

PREPARED FOR

ACME Corporation, Inc.

ACME Platform API — api.acme-corp.example

ENGAGEMENT ID	REPORT VERSION	TEST WINDOW	ISSUED
CTX-PT-2026-0444	1.2 — Final (Post-Retest)	31 Aug - 11 Sep 2026	18 September 2026

REDACTED DISTRIBUTION COPY

Request/response bodies, credentials, tokens, host identifiers and customer data are masked. The client copy contains all evidence in full. See §2 Redaction Key.

Cetonix · EON Freezone, Kharadi, Pune 411014, India · info@cetonix.com

Contents

This report is issued as a redacted distribution copy. Section 2 explains exactly what has been masked and what the client copy contains in its place.

1. Document Control

1.1 Engagement details

Engagement reference	CTX-PT-2026-0444
Client	ACME Corporation, Inc.
Engagement type	API penetration test — authenticated, multi-role and multi-tenant testing of REST and GraphQL interfaces
Target	ACME Platform API — api.acme-corp.example (REST v2 and GraphQL)
Testing window	31 August 2026 – 11 September 2026 (10 working days)
Retest window	15 September 2026 – 16 September 2026
Report issued	18 September 2026
Report version	1.2 — Final (post-retest)
Classification	CONFIDENTIAL · TLP:AMBER+STRICT
Testing standard	OWASP API Security Top 10 (2023) · OWASP WSTG v4.2 · OWASP ASVS v4.0.3 · PTES · NIST SP 800-115

1.2 Version history

VER.	DATE	AUTHOR	CHANGE
0.1	9 Sep 2026	Lead Consultant	Initial draft issued for internal peer review
0.9	12 Sep 2026	Technical QA Reviewer	Peer review complete; CVSS vectors validated; two findings re-scored
1.0	14 Sep 2026	Practice Manager	Final draft issued to client for remediation
1.1	16 Sep 2026	Lead Consultant	Retest results incorporated; nine findings verified as resolved
1.2	**18 September 2026**	**Practice Manager**	**Final issue — attestation letter appended; redacted distribution copy produced**

1.3 Distribution and authorised recipients

This document is released to the named recipients below. Onward distribution requires written approval from the client's engagement owner. Cetonix retains no copy of client credentials or captured data beyond the retention period stated in section 2.4.

NAME / ROLE	ORGANISATION	COPY ISSUED
[Client Engagement Owner] Head of Security	ACME Corporation, Inc.	**Client copy — unredacted**
[Client Technical Lead] Director of Platform Engineering	ACME Corporation, Inc.	**Client copy — unredacted**

NAME / ROLE	ORGANISATION	COPY ISSUED
[Client Compliance Lead] Director, GRC	ACME Corporation, Inc.	**Client copy — unredacted**
External auditor / assessor	[Audit firm]	**Redacted distribution copy**
Customer vendor-risk reviewers	Various	**Redacted distribution copy**
Lead Consultant, Test Team	Cetonix	Master copy — encrypted at rest

1.4 Assessment team

ROLE	RESPONSIBILITY	QUALIFICATION PROFILE
Lead Consultant	Test execution, findings, report authorship	Offensive security specialist; API and multi-tenant authorisation testing; 8 years' practice
Senior Analyst	GraphQL, partner integration and bulk-interface testing	Backend engineering background; API authorisation specialisation
Technical QA Reviewer	Independent peer review, CVSS validation	Did not participate in testing; reviews all findings prior to issue
Practice Manager	Engagement governance, client communication	Accountable for scope, impartiality screening and final sign-off

> **Named team, not a rotating pool**
> The same lead consultant runs the engagement from scoping through retest and is available to your engineering team throughout. Analyst identities are provided to the client in the unredacted copy and withheld here.

2. Confidentiality, Handling and Redaction Key

This copy has been prepared for onward distribution to parties outside the client's security team — auditors, assessors, prospective customers and vendor-risk reviewers. It carries the full narrative, methodology, findings, severity ratings and remediation guidance of the original, with sensitive technical detail masked so that the document cannot itself be used to attack the platform it describes.

> **What redaction does and does not remove**
> Nothing about the existence, severity, impact or remediation of a finding is withheld. What is removed is the material that would let a reader reproduce the attack: live identifiers, credentials, tokens, payload strings, internal hostnames and customer data.

2.1 Redaction key

Every masked value is replaced by a solid bar and a bracketed tag naming what was removed. The tags used in this report are listed below.

TAG	WHAT IT MASKS	IN THE CLIENT COPY
[REDACTED — SESSION TOKEN]	Bearer tokens, JWTs, refresh tokens and cookies captured during testing	Full value, truncated to 32 chars
[REDACTED — CREDENTIAL]	Account passwords, API keys, HMAC secrets and MFA codes	Full value
[REDACTED — TENANT UUID]	Workspace and tenant identifiers that map to real customers	Full identifier
[REDACTED — OBJECT ID]	Record identifiers used to demonstrate direct object reference flaws	Full identifier plus the enumeration range tested
[REDACTED — CUSTOMER PII]	Names, email addresses, billing contacts and any personal data observed	Recorded as a data-class count only; raw PII is never retained
[REDACTED — PAYLOAD]	Working exploit strings and injection payloads	Full payload with encoding notes
[REDACTED — PARTNER KEY]	Partner and integration API keys issued for the engagement	Full value and the scope attached to it
[REDACTED — SCHEMA DETAIL]	GraphQL type, field and resolver names recovered through introspection	Full schema extract and the affected field paths
[REDACTED — PRE-SIGNED URL]	Signed download URLs issued by the export and storage interfaces	Full URL, its validity window and its scope
[REDACTED — CREDENTIAL MATERIAL]	Any secret material returned by a vulnerable interface	**Masked in every copy — see 2.3**

2.2 Shared copy versus client copy

Figure 2.1 — The same authentication capture as it appears in this copy (left) and in the client copy (right).

2.3 Material redacted in every copy

Three classes of data are masked even in the client's own copy, because retaining them would create risk that outlasts the engagement:

- Live credential material returned by a vulnerable interface — partner keys, internal service tokens and signed URLs recovered during testing. Cetonix records the type and scope and instructs the client to rotate; the value itself is destroyed at capture.
- Customer personal data returned during cross-tenant testing. Records are counted and classified, never copied. Sampling stops at the minimum needed to prove impact, and export artefacts are cancelled and destroyed.
- Cardholder data. Testing is designed so that primary account numbers are never retrieved; where a payment reference is returned it is recorded as a token reference only.

2.4 Handling, retention and destruction

Transfer	Encrypted transfer to named recipients only. Reports are not sent as unprotected email attachments.
Storage	Evidence and drafts held encrypted at rest, accessible only to the assigned engagement team and the QA reviewer.
Retention	Evidence retained for 90 days after the retest closes, to support re-issue and auditor queries; the report itself is retained for the period agreed in the master services agreement.
Destruction	All captures, credentials and test accounts are destroyed on client request or at the end of the retention period, with written confirmation.
Third parties	No client data, capture or credential is shared with, or processed by, any third-party service. See section 10 for the position on AI tooling.

3. Executive Summary

PURPOSE

ACME Corporation engaged Cetonix to perform an independent penetration test of the ACME Platform API. The engagement was commissioned because the API now serves three first-party clients and six partner integrations, and because enterprise prospects had begun asking for API-specific testing evidence rather than accepting a web application test as coverage.

OVERALL ASSESSMENT

Overall risk rating: CRITICAL — reduced to LOW following remediation and retest

A read-only partner integration key reached an undocumented administrative namespace, enumerated the platform's customer base, and produced a downloadable export containing records belonging to multiple tenants. Nine of thirteen findings, including both rated Critical and all three rated High, were remediated and verified during the retest window.

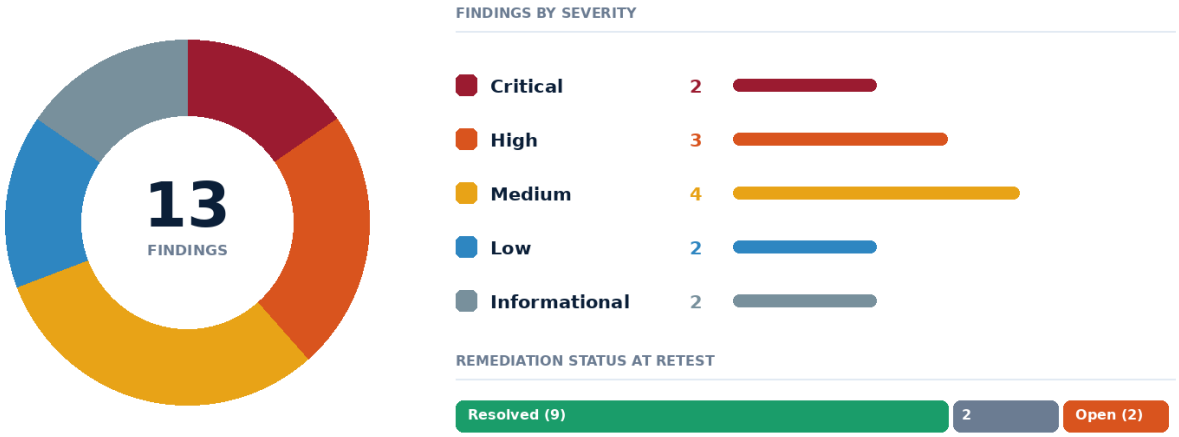


Figure 3.1 — Findings by severity and remediation status at retest.

3.1 What we found

Authentication is not the problem. Every interface tested required a valid credential, tokens behaved as documented, and no authentication bypass was achievable anywhere in the assessed surface.

Authorisation is the problem, and it fails at three different levels: the ***route*** (can this caller invoke this function at all), the ***object*** (does this caller own this record), and the ***property***. Each level is checked in some places and not others, which is the characteristic signature of authorisation implemented per endpoint rather than centrally.

Principal business risks

RISK	HOW IT ARISES	EXPOSURE
Customer data exposure across tenants	Objects are returned by identifier without checking the caller's tenant (API-01).	**Records of 4 tenants reached**
Customer base	An undocumented admin namespace carries no	**100 tenant records listed**

RISK	HOW IT ARISES	EXPOSURE
enumerated by a partner	role check (API-02).	
Bulk extraction at scale	The async export interface applies a caller-supplied tenant filter verbatim (API-04).	**Cross-tenant export produced**
Export artefacts reachable without auth	Pre-signed download URLs require no further authentication and are long-lived (API-03).	**URL alone is sufficient**
Over-disclosure in ordinary responses	Responses return internal properties that clients filter out but callers can read (API-05).	**Internal fields in normal traffic**

3.2 What was done well

- Authentication is correctly implemented across every interface. Token validation, expiry and revocation all behaved as documented, and no bypass was achievable.
- Transport security is correct: TLS 1.2 and 1.3 only, HSTS present, no weak suites, and no cleartext interface reachable.
- Input handling against classic injection classes was sound. No SQL, NoSQL, command or template injection was achievable across 148 endpoints and 84 GraphQL types.
- The payment path is properly tokenised — no primary account number was retrievable through any interface, and the cardholder data environment remained segmented.
- The documented REST specification is accurate for the endpoints it covers, which made systematic coverage possible and is better than most API engagements encounter.
- The platform team disabled the administrative namespace for partner credentials within three hours of notification, ahead of the formal report.

3.3 Recommended priorities

1. Introduce a single authorisation layer that every request passes through, resolving caller identity, tenant and role from the credential and applying them at the data-access boundary rather than in individual handlers.
2. Apply a default-deny route policy so that a new or undocumented route is unreachable until a role requirement is declared for it.
3. Treat every caller-supplied filter, identifier and property name as untrusted, particularly in bulk and asynchronous interfaces where one request can move a great deal of data.
4. Narrow pre-signed URL scope and validity, and bind artefacts to the identity that requested them.
5. Reconcile the deployed route table against the published specification continuously, so that shadow endpoints are found internally rather than by an assessor.

4. Compliance and Regulatory Alignment

The API is the control point for most of the data the platform holds, so API testing evidence carries weight across the whole compliance programme. The position is summarised below.

4.1 Framework coverage

FRAMEWORK	OBLIGATION ADDRESSED	FINDINGS AFFECTING	POSITION AT ISSUE
PCI DSS v4.0.1	Req. 11.4.2 / 11.4.3 internal and external penetration testing at least every 12 months to a documented methodology (11.4.1), with exploitable findings corrected and retested (11.4.4); Req. 6.2.4 secure development	API-01, API-02, API-05, API-07	**Satisfied — retest complete**
SOC 2 (TSC 2017)	CC6.1 logical access, CC6.3 access to data based on roles, CC6.6 protection against external threats, CC7.1 vulnerability identification	API-01, API-02, API-03, API-04, API-05	**Evidence provided**
ISO/IEC 27001:2022	Annex A 5.15 access control, A 8.3 information access restriction, A 8.8 technical vulnerability management, A 8.29 security testing	All findings	**Evidence provided**
HIPAA Security Rule	§164.308(a)(1)(ii)(A) risk analysis, §164.308(a)(8) periodic technical evaluation, §164.312(a)(1) access control, §164.312(b) audit controls	API-01, API-02, API-04, API-09	**Findings affecting ePHI paths remediated**
GDPR	Art. 25 data protection by design, Art. 32 security of processing, Art. 33 breach notification readiness	API-01, API-02, API-04	**Art. 32(1)(d) testing evidence**
Partner and customer contracts	Contractual commitments on tenant isolation and on the scope of partner integration access	API-01, API-02, API-04	**Review of partner scoping recommended**

Why the partner credential matters most

API-01 and API-02 were reached with a credential ACME issues to third parties under a read-only integration scope. That makes the exposure contractual as well as technical: six partners held a credential that could, on the evidence of this test, reach the customer base of the platform.

4.2 Control mapping by finding

ID	OWASP API TOP 10	PCI DSS V4.0.1	ISO 27001:2022	SOC 2
API-01	API1:2023 BOLA	7.2.1, 7.2.2	A 5.15, A 8.3	CC6.1, CC6.3
API-02	API5:2023 BFLA	7.2.2	A 5.18, A 8.2	CC6.1, CC6.3
API-03	API8:2023	3.4.1, 7.2.1	A 5.14, A 8.3	CC6.1, CC6.7
API-04	API6:2023	7.2.1	A 8.3, A 8.12	CC6.3
API-05	API3:2023 BOPLA	6.2.4	A 8.3	CC6.1
API-06	API4:2023	6.2.4	A 8.6	CC7.2
API-07	API4:2023	8.3.4	A 8.5, A 8.6	CC6.1, CC7.2
API-08	API9:2023	11.4.3, 6.5.3	A 5.9, A 8.9	CC8.1
API-09	API10:2023	6.2.4	A 5.19, A 8.28	CC6.6
API-10	API8:2023	6.4.2	A 8.9	CC6.6
API-11	API8:2023	6.2.4	A 8.28	CC7.1
API-12	API8:2023	6.3.3	A 8.8	CC7.1
API-13	API9:2023	—	A 5.9	CC8.1

5. Scope and Rules of Engagement

Scope, authorisation and constraints were agreed in writing before any traffic was sent to client infrastructure. The signed rules of engagement are held with the engagement record and are available to assessors on request.

5.1 In-scope assets

ASSET	TYPE	ENVIRONMENT	TEST DEPTH
api.acme-corp.example/v2	REST API	Production	148 endpoints, every role, both tenants
api.acme-corp.example/graphql	GraphQL	Production	84 types, 312 fields — authorisation, depth, complexity, batching
Export and job interface	Async API	Production	Filter handling, artefact scoping, URL issuance
Webhook dispatcher	Outbound	Production	Destination validation and delivery-log handling
Partner integration credentials	API keys	Production	Scope enforcement from a third-party credential
[redacted] legacy v1	REST API	Production	Discovered during recon — see API-08

5.2 Explicitly out of scope

EXCLUDED	REASON
Denial-of-service and volumetric testing	Excluded by agreement; production platform serving live customers
Third-party identity provider infrastructure	Operated by a third party; no authorisation to test
Payment processor environment	Out of the client's control; tested only at the tokenised boundary
Underlying cloud infrastructure	Covered by the provider's shared-responsibility model
First-party web and mobile clients	Assessed separately under CTX-PT-2026-0418, 0431 and 0437
Source code review	Not commissioned here; strongly recommended as a follow-on for the authorisation layer

5.3 Caller identities

Five distinct caller identities were provisioned, including two accounts in separate tenants. The second tenant is what makes isolation testable; the partner key is what made the two Critical findings visible, because it is the identity most likely to be held by someone outside ACME's control.

IDENTITY	CREDENTIALS	TENANT	PURPOSE
Unauthenticated	—	—	Confirming that every interface requires a credential
Member	2	Tenant A and Tenant B	Baseline permissions; object-level isolation testing
Workspace admin	2	Tenant A and Tenant B	Function-level authorisation; integration configuration
Partner integration	1	Partner tenant	Read-only scope by design — the identity that reached API-01 and API-02
Staff / internal	1	Cross-tenant	Establishing the intended boundary of the admin namespace

5.4 Rules of engagement

Testing window	09:00–21:00 client local time, Monday to Friday. Out-of-hours testing by prior agreement only.
Rate limiting	Automated requests capped at 5 per second; resource-consumption testing capped by agreement rather than by server-side controls.
Data handling	Where a flaw exposed another tenant's data, sampling stopped at the minimum required to evidence impact. Export artefacts were cancelled and destroyed at proof.
Destructive actions	No deletion, modification or corruption of production records. Write operations limited to objects owned by the test accounts.
Critical escalation	Findings of Critical severity reported to the client's named contact within two hours of confirmation. Both Criticals were reported within the hour.
Identification	All test traffic carried the header X-Cetonix-Assessment: CTX-PT-2026-0444 and a dedicated user-agent.
Stop conditions	Testing halts immediately on evidence of prior compromise, on production instability, or on client instruction.

Scope limitations affecting coverage

Resource-consumption testing was capped by the rules of engagement well below the point at which the platform showed distress, so API-06 and API-07 describe the absence of controls rather than a measured breaking point. No other constraint materially limited coverage.

6. Methodology

The engagement applied the OWASP API Security Top 10 (2023) systematically across every interface, using the test procedures of the OWASP Web Security Testing Guide v4.2 and the verification requirements of ASVS v4.0.3, within the phase structure of PTES and NIST SP 800-115. Tooling provided coverage; every finding was manually reproduced before it was written up.

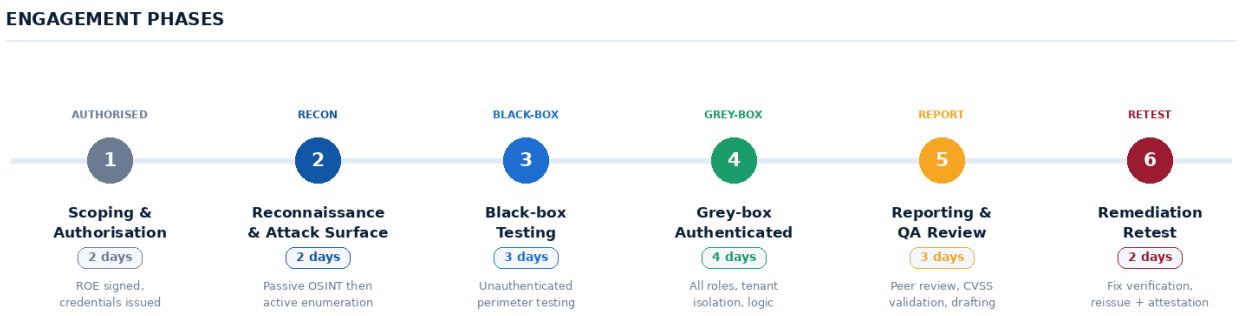


Figure 6.1 — Engagement phases and duration.

6.1 The authorisation matrix

The core of an API engagement is mechanical and unglamorous: every request, replayed as every identity, against objects belonging to every tenant. 148 endpoints across 5 identities and 2 tenants is a large matrix, and the findings in this report are the cells where the answer differed from the intended one.

Route coverage	Every documented endpoint invoked by every identity, including identities that should receive a refusal, to confirm the refusal actually occurs.
Object coverage	Every object-returning endpoint invoked with identifiers belonging to the other tenant, to test isolation rather than existence.
Property coverage	Responses compared across identities to find properties returned to callers who should not see them, and request bodies extended with properties that should not be settable.
Undocumented surface	First-party client traffic proxied and compared against the specification, to find routes the documentation does not mention.
Asynchronous surface	Export, job and webhook interfaces tested separately, because they accept filters and destinations that ordinary endpoints do not.

6.2 Coverage against the OWASP API Security Top 10

CATEGORY	TESTED	OUTCOME
API1:2023 Broken object level authorization	**Yes**	API-01 — principal weakness
API2:2023 Broken authentication	**Yes**	No issues identified; authentication is sound
API3:2023 Broken object property	**Yes**	API-05

CATEGORY	TESTED	OUTCOME
level authorization		
API4:2023 Unrestricted resource consumption	Yes	API-06, API-07
API5:2023 Broken function level authorization	Yes	API-02
API6:2023 Unrestricted access to sensitive business flows	Yes	API-04
API7:2023 Server side request forgery	Yes	Webhook destination validation found sound at this engagement
API8:2023 Security misconfiguration	Yes	API-03, API-10, API-11
API9:2023 Improper inventory management	Yes	API-08, API-13
API10:2023 Unsafe consumption of APIs	Yes	API-09

7. Reconnaissance and Attack Surface

API reconnaissance answers a question the documentation cannot: what is actually deployed. The published specification describes what the team believes it runs; proxying the first-party clients and probing systematically describes what is reachable.

7.1 Discovered attack surface

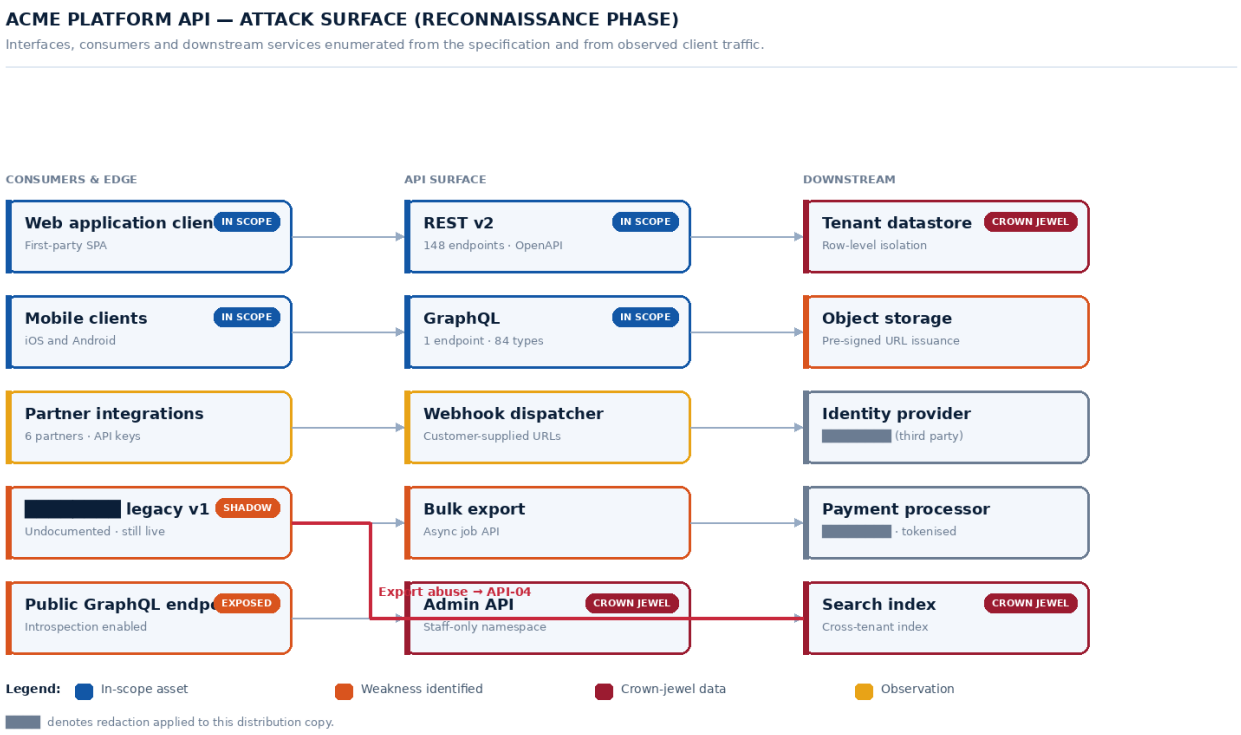


Figure 7.1 — API attack surface with the demonstrated attack path overlaid.

7.2 Specification versus deployment

SURFACE	STATUS	ASSESSMENT
REST v2 — documented	**148 endpoints**	The published specification is accurate for what it covers, which made systematic coverage possible.
Admin namespace	**Undocumented**	Discovered by path inference from the staff console client. Not in the specification, and carrying no role check — API-02.
Legacy v1 interface	**Live**	Superseded interface still answering, under an older authorisation model that does not carry the v2 fixes — API-08.
GraphQL endpoint	**Introspection on**	84 types and 312 fields recoverable on request, handing an attacker the inventory — API-06.
Export and job interface	**Partially documented**	Filter semantics are not described, and the filter is applied verbatim — API-04.

SURFACE	STATUS	ASSESSMENT
Webhook dispatcher	**Sound**	Destination validation is applied after resolution and redirects are refused. Tested specifically; no issue identified.

7.3 GraphQL assessment

GraphQL assessment — API-06 · introspection, depth and batching

```
$ cetonix-api graphql --endpoint https://api.acme-corp.example/graphql

[*] Introspection   ENABLED in production · 84 types · 312 fields recovered
[*] Schema written to local analysis only — not included in this report

[!] DEPTH       no depth limit · nested query to depth 14 accepted
[!] COMPLEXITY  no cost analysis · single query fanned out to 4,0[redacted] resolvers
[!] ALIASING    120 aliased mutations accepted in one document
[!] BATCHING    array batching enabled · rate limit applied per request, not per op

[!] FIELD-AUTH  field [redacted] [REDACTED]      resolved for a role that cannot read the parent
[!] ERRORS      resolver errors leak internal type and table names

[*] Response time under complexity test: 14.2 s (baseline 0.2 s)
[*] Testing capped by rules of engagement, not by any server-side control

[=] 4 resource-consumption weaknesses · 1 field-level authorisation failure
```

Evidence API-06-A · Captured 04 Sep 2026 13:20 UTC · Field names and schema detail withheld from this copy

Evidence API-06-A — GraphQL introspection, depth and batching results. Field names withheld from this copy.

8. Authorisation Testing

This is the substance of the engagement. Each level of authorisation was tested independently, because a platform commonly gets one right and another wrong, and the difference determines both severity and remedy.

8.1 Function-level authorisation

Every endpoint was invoked by every identity, including identities that should have been refused. The refusals were correct throughout the documented surface. They were absent entirely in the undocumented administrative namespace, which no role check covers because no role requirement was ever declared for routes the specification does not mention.

Evidence API-02-A — A read-only partner key invoking the administrative tenant directory. Tenant names masked.

8.2 Object-level authorisation

Every object-returning endpoint was invoked with identifiers belonging to a different tenant. Eleven of fourteen object types are correctly scoped. Three are not, and the pattern is consistent: those three resolve the object first and check the caller afterwards, or not at all.

Evidence API-01-A — A partner credential retrieving another tenant's contact record. Identifiers and customer data masked.

8.3 Property-level authorisation

Responses were compared across identities, and request bodies were extended with properties that ought not to be settable.

TEST	RESULT	OBSERVATION
Excessive data in responses	**Fail**	Internal scoring, ownership and cost fields returned to all callers; first-party clients filter them out — API-05
Mass assignment on update	**Fail**	Ownership and workspace properties accepted from the request body on two endpoints — API-05
GraphQL field-level authorisation	**Fail**	One field resolved for a role that cannot read its parent object — API-06
Sensitive fields in list responses	**Pass**	List endpoints correctly return a reduced projection

8.4 Sensitive business flows

Bulk and asynchronous interfaces deserve separate attention because one request can move far more data than an ordinary endpoint, and because they frequently accept parameters — filters, ranges, formats — that ordinary endpoints do not.

Evidence API-04-A — An export job accepting a cross-tenant filter. Row counts and URLs masked.

9. Resource Consumption, Misconfiguration and Logging

9.1 Resource consumption

CONTROL	PRESENT	ASSESSMENT
REST rate limiting	Partial	Applied per credential on most endpoints; absent on the authentication and export endpoints — API-07
GraphQL depth limiting	No	A query nested to depth 14 was accepted and executed — API-06
GraphQL complexity analysis	No	A single query fanned out to approximately 4,0[redacted] resolvers, taking 14.2 seconds against a 0.2 second baseline
Batching and aliasing limits	No	120 aliased mutations accepted in one document; rate limits apply per request, not per operation
Pagination limits	Yes	Maximum page size enforced server-side and cannot be raised by the caller
Export job concurrency	No	No limit on concurrent export jobs per credential

9.2 Configuration

- CORS is configured with a permissive origin reflection on the GraphQL endpoint, which undermines the browser's origin protection for any credentialed request — API-10.
- Resolver errors return internal type and table names, disclosing schema detail that introspection is meant to gate — API-11.
- Security headers are present on REST responses but absent on the GraphQL endpoint — API-10.
- Transport configuration is correct throughout: TLS 1.2 and 1.3 only, HSTS present, no weak suites, no cleartext interface reachable.

9.3 Logging and detectability

As in the platform's other engagements, the question is not only whether a weakness exists but whether its exploitation would be visible afterwards.

ACTIVITY	LOGGED	COMMENT
Authentication and key usage	Yes	Recorded with credential identifier and source address
Cross-tenant object access	No	Object reads are not logged with the caller's tenant, so API-01 is not reconstructable after the fact
Admin namespace invocation	No	The undocumented namespace has no audit coverage at all — API-02 would leave no trace
Export job creation and filters	Partial	Job creation is recorded; the filter applied is not, so the scope of an export is not

ACTIVITY	LOGGED	COMMENT
		reconstructable
Pre-signed URL issuance and use	**No**	Neither issuance nor download is logged — API-03
GraphQL query cost	**No**	No per-query cost or depth recorded, so resource abuse is invisible until it causes an outage

The question you cannot currently answer

Three of the five most serious findings would have produced no audit record. After the fixes land, ACME still cannot establish whether a partner credential reached another tenant's data before this engagement found that it could. That gap is worth closing on its own terms.

10. Use of Artificial Intelligence in this Engagement

Cetonix discloses the role of AI tooling in every engagement, because a client buying assurance is entitled to know what was assessed by a person and what was not. The position is simple: AI accelerates the analyst, and never substitutes for them.

Enterprise models only — your data is never used for training

All AI assistance runs exclusively on enterprise-tier model services procured under commercial agreements that contractually prohibit the use of customer or engagement data for model training, with zero data retention enabled where the provider offers it. Consumer and free-tier AI services are prohibited by Cetonix policy and are blocked on engagement systems. Nothing you disclose to us — code, credentials, captures, findings or this report — enters a training corpus.

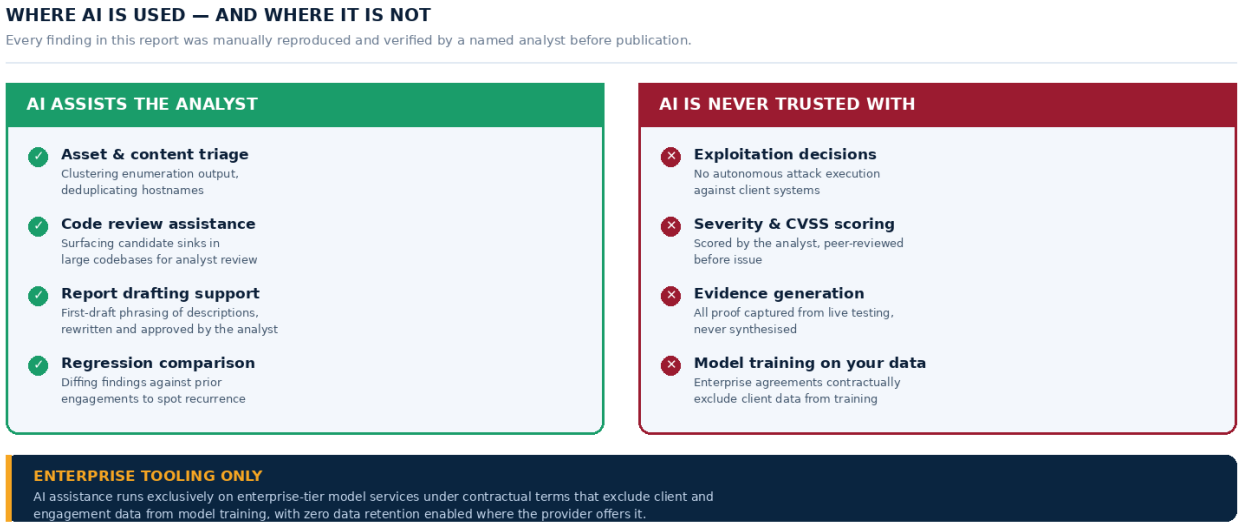


Figure 10.1 — Division between AI-assisted and analyst-only activity.

10.1 Where AI was used in this engagement

ACTIVITY	HOW AI WAS USED	HUMAN CONTROL
Endpoint inventory	Grouping 148 REST endpoints and 84 GraphQL types by authorisation pattern to plan coverage	Every endpoint and type was tested; grouping affected sequencing only
Response differential review	Highlighting statistically unusual responses across large request sets for analyst attention	Every highlighted response was manually inspected; API-01 was confirmed by hand
Specification comparison	Diffing the published OpenAPI specification against endpoints observed in client traffic	Analyst confirmed each undocumented endpoint manually; three were escalated as API-08
Report drafting	First-draft phrasing of finding descriptions and remediation text from analyst notes	Rewritten and approved by the lead consultant; technical content originates from the analyst

ACTIVITY	HOW AI WAS USED	HUMAN CONTROL
Regression comparison	Comparing findings against the web application and mobile engagements on the same backend	Analyst validated each match; two shared weaknesses confirmed and noted in section 15

10.2 Where AI was not used

Exploitation	No autonomous agent was permitted to interact with client systems. Every request that reached the client's systems was issued by, or under the direct control of, a named analyst.
Severity and CVSS scoring	All ratings were assigned by the analyst and independently validated by the QA reviewer. No score in this report was generated by a model.
Evidence	All evidence is captured from live testing. Nothing in section 13 is reconstructed, illustrative or synthesised.
Finding determination	No finding was published on the basis of a model's assertion. Reproduction by a human analyst is a precondition of inclusion.

10.3 Model provenance and data handling

The controls below are contractual and technical, not aspirational. They are available for review during vendor due diligence, and they apply to every engagement Cetonix performs.

Service tier	Enterprise or business-tier model services only, procured under a signed commercial agreement. Consumer and free-tier AI services are prohibited by policy and blocked on engagement systems.
Training exclusion	Every agreement contractually excludes customer content — prompts, files, outputs and engagement material — from model training, fine-tuning and human review for product improvement.
Retention	Zero data retention is enabled where the provider offers it. Where it is not, retention is limited to the provider's minimum abuse-monitoring window and no client identifiers are submitted.
Data minimisation	Client credentials, captured traffic, customer records and personal data are never submitted to an AI service. Assistance operates on analyst notes and Cetonix-side metadata.
Residency and access	Requests are issued from Cetonix-controlled accounts with named-user access, audit logging and no third-party plugin or connector access to engagement material.
Evidence on request	The relevant provider terms and the internal AI usage policy can be supplied to the client's procurement or security function on request.

Our commitment on AI

If a finding appears in a Cetonix report, a named analyst reproduced it, scored it and stands behind it. AI changes how quickly we get to the interesting requests; it does not change who decides what is true, and it never learns from your data.

11. Findings Summary

ID	FINDING	SEVERITY	CVSS	CWE	STATUS
API-01	Broken object level authorisation on three object types	Critical	9.1	CWE-639	Resolved
API-02	Undocumented admin namespace reachable by a partner key	Critical	9.1	CWE-285	Resolved
API-03	Pre-signed export URLs unauthenticated and long-lived	High	8.2	CWE-522	Resolved
API-04	Export interface accepts a cross-tenant filter	High	8.1	CWE-863	Resolved
API-05	Excessive data exposure and mass assignment	High	7.3	CWE-915	Resolved
API-06	GraphQL introspection with no depth or cost limits	Medium	6.5	CWE-770	Resolved
API-07	No rate limiting on authentication and export endpoints	Medium	5.9	CWE-307	Resolved
API-08	Legacy v1 interface live under an older authorisation model	Medium	5.8	CWE-1059	Resolved
API-09	Upstream API responses consumed without validation	Medium	5.3	CWE-20	Resolved
API-10	Permissive CORS and missing headers on GraphQL	Low	3.7	CWE-942	Open
API-11	Resolver errors disclose internal schema detail	Low	3.5	CWE-209	Accepted
API-12	Outdated dependency in the gateway with no reachable sink	Info	—	CWE-1104	Open
API-13	No published deprecation or versioning policy	Info	—	CWE-1059	Accepted

11.1 Risk positioning

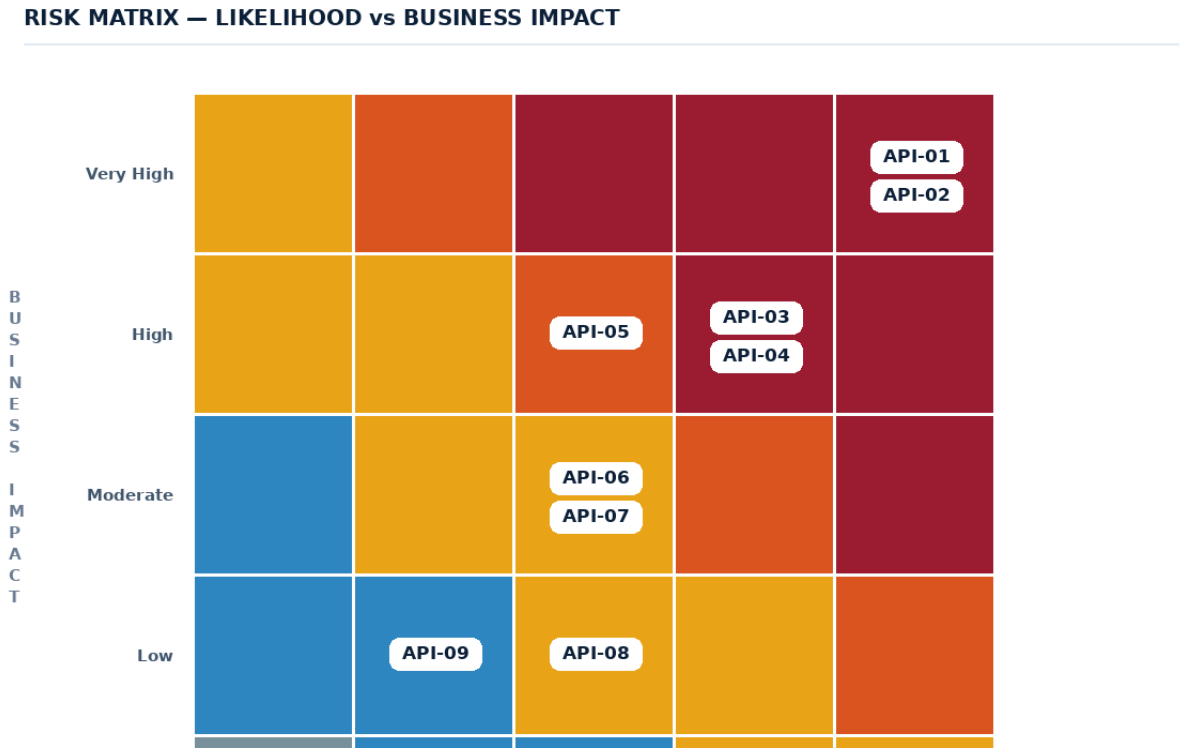


Figure 11.1 — Findings plotted by likelihood of exploitation against business impact.

12. Attack Narrative

The sequence below was executed with a single credential: a read-only partner integration key of the kind ACME issues to third parties. No account password, no employee credential and no insider knowledge was used.

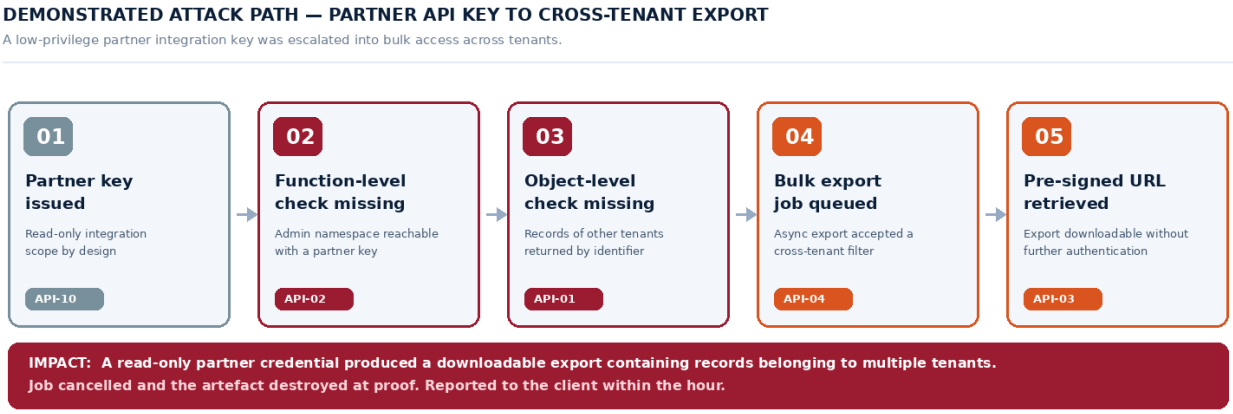


Figure 12.1 — Demonstrated attack path from a partner integration key to a cross-tenant export.

12.1 Narrative

1. **Entry.** A partner integration key was issued under a read-only scope limited to the partner's own tenant — the intended, documented behaviour (API-13).
2. **Function-level gap.** An administrative namespace, absent from the published specification and discovered by path inference from the staff console client, accepted the partner key. No role requirement is declared for routes the specification does not mention, and the default is permit (API-02).
3. **Object-level gap.** Contact, invoice and document objects were returned by identifier without any check that the caller's tenant matched the object's owner. Records belonging to four tenants were confirmed readable (API-01).
4. **Scale.** The asynchronous export interface accepted a workspace filter naming a tenant other than the caller's, and applied it verbatim to the query (API-04).
5. **Retrieval.** The completed job returned a pre-signed download URL requiring no further authentication and valid for seven days. The job was cancelled and the artefact destroyed at proof (API-03).

Elapsed time from first request to a downloadable cross-tenant export: 2 hours 18 minutes

The credential used is one ACME issues to third parties. Six partners held an equivalent key at the time of testing, and nothing in the platform's logs would show whether any of them had done this.

12.2 What would have broken the chain

- A default-deny route policy would have stopped step 2, because an undeclared route would have been unreachable rather than open.

- Tenant scoping applied at the data-access layer would have stopped steps 3 and 4 together, independently of every other fix.
- Binding the export artefact to the requesting identity would have stopped step 5 even after a cross-tenant job was queued.
- Logging the caller's tenant on object reads would not have prevented the chain, but would let ACME answer whether anyone else had walked it.

13. Detailed Findings

Each finding records what was found, how it was proved, what it means for the business, and what to do about it. Evidence is reproduced as captured, with sensitive values masked according to the key in section 2.

API-01 Broken object level authorisation on three object types

CRITICAL CVSS 9.1

CVSS v3.1	AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N	**CWE**	CWE-639 · CWE-284
Affected asset	REST v2 — contacts, invoices and documents	**OWASP**	API1:2023 BOLA
Status	**Resolved — verified 16 Sep 2026**	**Compliance**	PCI 7.2.1 · ISO A 8.3 · SOC 2 CC6.1 · HIPAA §164.312(a)(1)
References	OWASP API Security Top 10 — API1:2023 Broken Object Level Authorization · OWASP WSTG-ATHZ-04 · OWASP ASVS v4.0.3 § 4.2.1 · CWE-639 Authorization Bypass Through User-Controlled Key		

Description

Three object-returning endpoints resolve the requested object by its identifier and return it without verifying that it belongs to the caller's tenant. The caller's tenant is read from a request header rather than derived from the credential, and it is never compared with the object's owner.

Eleven of the fourteen object types tested are correctly scoped, which is what makes this a per-endpoint implementation problem rather than a platform-wide design decision — and what makes it likely to recur on the next endpoint somebody writes.

It was reached with a read-only partner integration key, so the caller need not even be an ACME customer.

Business impact

Contact, invoice and document records belonging to other tenants are readable by any authenticated caller who knows or can guess an identifier. Identifiers are sequential within a predictable range, so guessing is not a barrier.

These records contain personal data under GDPR and, for ACME's healthcare customers, ePHI. Invoice records additionally disclose commercial terms.

Because object reads are not logged with the caller's tenant, prior exploitation could not be ruled in or out from existing logs — which matters for breach-notification decisions as much as for remediation.

Evidence

Evidence API-01-A — A partner credential retrieving another tenant's contact record. Identifiers and customer data masked.

Reproduction

1. Authenticate with any valid credential, including a read-only partner integration key.
2. Retrieve an object the caller owns and note the identifier format: GET /api/v2/contacts/ **[REDACTED — OBJECT ID]**
3. Obtain an identifier belonging to another tenant. Identifiers are sequential within a narrow range, so this requires no prior knowledge.
4. Reissue the request with the original credential and the foreign identifier, leaving the workspace header set to the caller's own tenant.
5. Observe HTTP 200 with the full object body of the other tenant's record. Repeat for invoices and documents. Confirmed across four tenants, then halted.

Remediation

- Derive the caller's tenant from the credential server-side. Never accept a tenant or workspace identifier from a header, parameter or body field for authorisation purposes.
- Apply the tenant predicate in the data-access layer rather than in each handler, so that a missing check in a new endpoint cannot reintroduce the flaw. Row-level security in the database is the strongest form of this.
- Return HTTP 404 rather than 403 for objects outside the caller's tenant, so the response does not confirm that another tenant's record exists.
- Replace sequential identifiers with unguessable ones. This is defence in depth, not a fix — it raises the cost of enumeration without restoring authorisation.
- Add an automated test asserting that every object-returning endpoint refuses a cross-tenant identifier, and run it in the pipeline on every release.

> **One authorisation layer, not 148 of them**
> API-01, API-02, API-04 and API-05 are four expressions of the same root cause: authorisation decided per handler, from data the caller supplied. A single layer that resolves identity, tenant and role from the credential and applies them at the data boundary resolves all four and prevents the next one.

API-02 Undocumented admin namespace reachable by a partner key

CRITICAL CVSS 9.1

CVSS v3.1	AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:L/A:N	**CWE**	CWE-285 · CWE-1220
Affected asset	REST v2 — /api/v2/admin/*	**OWASP**	API5:2023 BFLA
Status	**Resolved — verified 16 Sep 2026**	**Compliance**	PCI 7.2.2 · ISO A 5.18 · SOC 2 CC6.3 · HIPAA §164.312(a)(1)
References	OWASP API Security Top 10 — API5:2023 Broken Function Level Authorization · OWASP WSTG-ATHZ-02 · OWASP ASVS v4.0.3 § 4.1.3 · CWE-285 Improper Authorization		

Description

An administrative namespace serves the internal staff console. It does not appear in the published OpenAPI specification and is not referenced by any customer-facing client, but it is served from the same host and the same gateway as the public API.

Routes in this namespace carry no role requirement. The platform's authorisation middleware applies role checks where a route declares one, and permits the request where none is declared — so an undocumented route is not merely undocumented, it is unprotected.

A read-only partner integration key invoked the tenant directory search endpoint and enumerated one hundred customer organisations with their plan and seat counts.

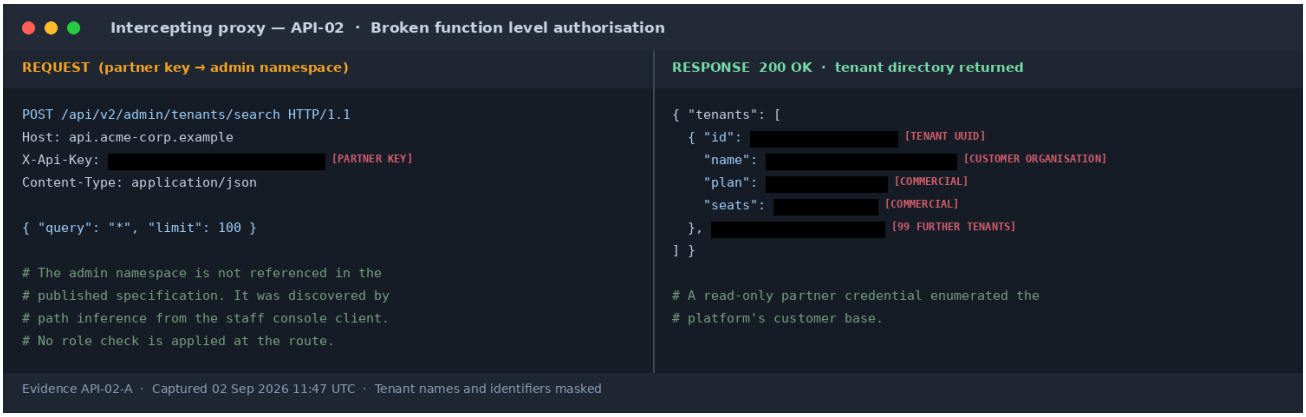
Business impact

Any authenticated caller — customer, trial account or third-party partner — can reach staff-only functionality. The tenant directory alone discloses ACME's entire customer base, including organisation names and commercial terms, which is competitively sensitive independent of the security impact.

Because the namespace has no audit coverage, an organisation could not establish afterwards whether it had been reached, by whom, or how often.

The contractual dimension is as significant as the technical one: six partners held a credential capable of this at the time of testing.

Evidence



Evidence API-02-A — A read-only partner key invoking the administrative tenant directory. Tenant names and identifiers masked.

Reproduction

1. Authenticate with a read-only partner integration key.
2. Infer the administrative path from the staff console client's traffic; it is not present in the published specification.
3. Invoke the directory search endpoint with a permissive query: POST /api/v2/admin/tenants/search
4. Observe HTTP 200 with a list of tenant records including organisation names, plans and seat counts.
5. Confirm the same credential reaches further routes in the namespace. Enumeration was halted at one hundred records and reported within the hour.

Remediation

- Change the middleware default from permit to deny: a route with no declared role requirement must be unreachable, and the application should refuse to start if any registered route lacks one.
- Serve administrative functionality from a separate host and gateway, restricted at the network layer to internal callers, so that a missing application check is not the only control.
- Declare an explicit staff-role requirement on every route in the namespace, and verify it with an automated test that calls each one as a customer identity.
- Bring the namespace under audit logging, recording actor, action and subject for every call.
- Include undocumented routes in the specification reconciliation described in API-08, so the gap between deployed and documented surface is visible continuously.

API-03 Pre-signed export URLs unauthenticated and long-lived

HIGH CVSS 8.2

CVSS v3.1	AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:N/A:N	**CWE**	CWE-522 · CWE-863
Affected asset	Export interface and object storage	**OWASP**	API8:2023
Status	**Resolved — verified 16 Sep 2026**	**Compliance**	PCI 3.4.1, 7.2.1 · ISO A 5.14 · SOC 2 CC6.7 · HIPAA §164.312(e)(1)
References	OWASP API Security Top 10 — API8:2023 Security Misconfiguration · OWASP ASVS v4.0.3 § 4.3.1 · CWE-522 Insufficiently Protected Credentials		

Description

Completed export jobs are delivered as pre-signed URLs pointing directly at object storage. The URL requires no further authentication, is valid for seven days, is not bound to the identity that requested the export, and its issuance and use are not logged.

Anyone who obtains the URL — from a browser history, a proxy log, a shared link, a forwarded email or a support ticket — retrieves the export.

Business impact

The export is typically the largest single concentration of customer data the platform produces, and the control protecting it is a URL. Combined with API-04, the artefact may also contain records belonging to tenants other than the requester.

A seven-day window is long enough for a URL to travel well beyond its intended recipient through entirely ordinary means.

Because downloads are not logged, ACME cannot determine whether an export was retrieved once or a hundred times, or by whom.

Evidence

Reproduction

1. Queue an export job as any authenticated caller.
2. Retrieve the completed job and note the download URL: [redacted] **[REDACTED — PRE-SIGNED URL]**
3. Present the URL with no credential, from a different network and a clean client. The export downloads.
4. Re-present the URL after several days. It remains valid for the full seven-day window.
5. Review the platform's logs. Neither issuance nor download appears.

Remediation

- Serve exports through an authenticated endpoint that validates the caller and their entitlement to the artefact, streaming from storage rather than redirecting to it.

- If pre-signed URLs must be used, reduce validity to minutes, bind them to the requesting identity, and make them single-use.
- Log issuance and every retrieval, with caller and source address, and alert on retrieval from an unexpected network.
- Encrypt export artefacts at rest with a key the storage layer alone cannot release, and delete them on a short schedule.

API-04 Export interface accepts a cross-tenant filter — HIGH CVSS 8.1

CVSS v3.1	AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N	**CWE**	CWE-863
Affected asset	Export interface — POST /api/v2/exports	**Status**	**Resolved — verified 16 Sep 2026**
Compliance	PCI 7.2.1 · ISO A 8.3, A 8.12 · SOC 2 CC6.3 · HIPAA §164.312(a)(1)		
References	OWASP API Security Top 10 — API6:2023 Unrestricted Access to Sensitive Business Flows · OWASP ASVS v4.0.3 § 4.2.2 · CWE-863 Incorrect Authorization		

Description

The asynchronous export interface accepts a filter object and applies it verbatim to the underlying query. A filter naming a workspace other than the caller's was accepted, and the resulting artefact contained that tenant's records.

Bulk interfaces amplify every authorisation weakness: one accepted request moves what would otherwise take thousands of individual calls, and does so through a code path that ordinary endpoint tests do not cover.

Business impact

This converts the object-level weakness in API-01 from a per-record issue into a bulk extraction capability, and does so through a documented, supported feature. The job was cancelled and the artefact destroyed at proof; a real attacker would have downloaded it within the hour.

Remediation

- Intersect every caller-supplied filter with the caller's own tenant scope server-side; never apply a tenant filter as given.
- Validate filters against an allow-list of permitted fields and operators rather than passing them to the query layer.
- Record the effective filter with the job so the scope of an export is reconstructable afterwards.
- Rate-limit and cap concurrent export jobs per credential, and alert on exports that exceed a normal row count for the tenant.

API-05 Excessive data exposure and mass assignment — HIGH CVSS 7.3

CVSS v3.1	AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:N	**CWE**	CWE-915 · CWE-213
Affected asset	REST v2 — object read and update endpoints	**Status**	**Resolved — verified 16 Sep 2026**
Compliance	PCI 6.2.4 · ISO A 8.3 · SOC 2 CC6.1		
References	OWASP API Security Top 10 — API3:2023 Broken Object Property Level Authorization · OWASP ASVS v4.0.3 § 5.1.2 · CWE-915 Improperly Controlled Modification of Dynamically-Determined Object Attributes		

Description

Object responses return the full internal representation, including risk scores, internal ownership references and cost fields. The first-party clients filter these out before display, so they are invisible in the product and plainly visible to anyone reading the response.

In the opposite direction, two update endpoints bind request bodies onto the persistence model inclusively, accepting ownership and workspace properties that no client is intended to set.

Business impact

Callers see commercially sensitive internal data in the course of ordinary use, with no attack required — only a proxy. The mass assignment half is more serious: setting an ownership or workspace property is a direct route to moving an object between tenants, which is the write-side equivalent of API-01.

Evidence

```
PATH /api/v2/documents/[redacted] [REDACTED — OBJECT ID] HTTP/1.1
X-Api-Key: [redacted] [REDACTED — PARTNER KEY]

{ "title": "assessment",
  "workspace_id": [redacted] [REDACTED — FOREIGN TENANT]
  "owner_id": [redacted] [REDACTED — FOREIGN USER]
# neither property appears in the documented schema

HTTP/1.1 200 OK  — both properties applied
```

Remediation

- Define explicit response projections per endpoint and per role. The client should not be the thing deciding what the caller may see.
- Bind request bodies through an allow-list of client-settable fields. Properties governing ownership, tenancy or entitlement must never appear on that list.
- Review every endpoint that serialises a persistence model directly; the pattern is rarely present in only two places.
- Add a test that asserts responses contain no property outside the documented schema, and run it in the pipeline.

API-06 GraphQL introspection with no depth or cost limits

MEDIUM
CVSS 6.5

CVSS v3.1	AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:H	**CWE**	CWE-770 · CWE-200
Affected asset	GraphQL endpoint	**Status**	**Resolved — verified 16 Sep 2026**
Compliance	PCI 6.2.4 · ISO A 8.6 · SOC 2 CC7.2		
References	OWASP API Security Top 10 — API4:2023 Unrestricted Resource Consumption · OWASP GraphQL Cheat Sheet · CWE-770 Allocation of Resources Without Limits or Throttling		

Description

Introspection is enabled in production, returning 84 types and 312 fields on request. No depth limit, complexity analysis, aliasing limit or per-operation rate limit is applied.

A query nested to depth 14 was accepted; a single document fanned out to approximately 4,0[redacted] resolvers and took 14.2 seconds against a 0.2 second baseline; 120 aliased mutations were accepted in one request. One field also resolved for a role that cannot read its parent object.

Business impact

Introspection hands an attacker the complete inventory, which is how the field-level authorisation gap was found in the first place. The absence of cost controls means a single well-formed request can consume disproportionate backend resources, and the per-request rate limit does not constrain it because one request can carry a hundred operations.

Remediation

- Disable introspection in production; serve the schema to developers through the documentation portal instead.
- Apply a query depth limit and a complexity or cost budget, rejecting queries that exceed it before execution.
- Limit aliasing and batching, and apply rate limits per operation rather than per request.
- Apply authorisation at the field resolver, not only at the root query, and test each field against a role that cannot read its parent.
- Record per-query cost so that resource abuse is visible before it becomes an outage.

API-07 No rate limiting on authentication and export endpoints — MEDIUM CVSS 5.9

CVSS v3.1	AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:L	**CWE**	CWE-307 · CWE-770
Affected asset	REST v2 — /auth/* and /exports	**Status**	**Resolved — verified 16 Sep 2026**
Compliance	PCI 8.3.4 · ISO A 8.5, A 8.6 · SOC 2 CC6.1, CC7.2		
References	OWASP API Security Top 10 — API4:2023 Unrestricted Resource Consumption · OWASP WSTG-ATHN-03 · CWE-307 Improper Restriction of Excessive Authentication Attempts		

Description

Rate limiting is applied per credential across most of the REST surface, but is absent on the authentication endpoints and on export job creation. Sustained automated attempts were accepted at the tested ceiling without throttling, lockout or alert.

The token-exchange endpoint is unthrottled alongside the login endpoint, so a captured refresh token can be exercised without constraint.

Business impact

Credential stuffing against the API is unconstrained, and export job creation can be used to consume backend capacity. Testing was capped by the rules of engagement, not by any control in the platform — so the practical ceiling is unknown to ACME as well as to Cetonix.

Remediation

- Apply rate limiting per credential, per source address and globally on the authentication endpoints, with progressive delay on repeated failure.
- Cap concurrent and daily export jobs per credential, and queue rather than execute beyond the cap.
- Alert on authentication failure volume and on credential-stuffing patterns such as many identities from one source.
- Apply limits at the gateway so that a new endpoint inherits them by default rather than by remembering to add them.

API-08 Legacy v1 interface live under an older authorisation model

MEDIUM
CVSS 5.8

CVSS v3.1	AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:N	**CWE**	CWE-1059 · CWE-1188
Affected asset	Legacy v1 REST interface — host redacted	**Status**	**Resolved — decommissioned 9 Sep 2026**
Compliance	PCI 6.5.3, 11.4.3 · ISO A 5.9, A 8.9 · SOC 2 CC8.1		
References	OWASP API Security Top 10 — API9:2023 Improper Inventory Management · OWASP WSTG-CONF-05 · CWE-1059 Insufficient Technical Documentation		

Description

A superseded v1 interface remains reachable from the internet, serving a subset of the v2 functionality under an older authorisation model. It does not appear in the asset inventory or the published specification.

Two endpoints on v1 returned objects that their v2 equivalents correctly refuse, indicating that the v1 interface has not received the authorisation fixes applied to v2 over time.

Business impact

An attacker who finds the older interface bypasses fixes applied to the current one. This is the characteristic risk of a surviving legacy version: it is not merely unmaintained, it actively undoes remediation performed elsewhere. It also undermines the scope definition on which a PCI assessment depends.

Remediation

- Decommission the interface. Where a partner still depends on it, place it behind authentication and network restriction with a published sunset date.
- Reconcile the deployed route table against the published specification on a recurring schedule so that shadow surface is found internally.
- Include every reachable version in the scope of future assessments, not only the current one.
- Publish a deprecation and versioning policy so that retirement is a process rather than an omission (API-13).

API-09 Upstream API responses consumed without validation

MEDIUM
CVSS 5.3

CVSS v3.1	AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:L	**CWE**	CWE-20
Affected asset	Third-party enrichment integration	**Status**	**Resolved — verified 16 Sep 2026**
Compliance	PCI 6.2.4 · ISO A 5.19, A 8.28 · SOC 2 CC6.6		
References	OWASP API Security Top 10 — API10:2023 Unsafe Consumption of APIs · OWASP ASVS v4.0.3 § 5.2.4 · CWE-20 Improper Input Validation		

Description

Responses from a third-party enrichment service are parsed and persisted without schema validation, size limits or type checking, and fields from them are rendered in first-party clients without encoding.

The integration also follows redirects returned by the upstream service without re-validating the destination.

Business impact

The platform trusts a third party as much as it trusts itself. A compromise at the upstream provider, or a provider bug, propagates directly into ACME's data store and into content rendered to customers. This is the failure mode behind a meaningful share of recent supply-chain incidents.

Remediation

- Validate upstream responses against an explicit schema, and enforce size and type limits before persistence.
- Encode third-party content on output in every client, exactly as user-supplied content is treated.
- Do not follow redirects from upstream services; treat a redirect as an error and alert on it.
- Apply timeouts and circuit breakers so an upstream failure degrades the feature rather than the platform.

API-10 Permissive CORS and missing headers on GraphQL

LOW
CVSS 3.7

CVSS v3.1	AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:L/A:N	**CWE**	CWE-942 · CWE-693
Affected asset	GraphQL endpoint	**Status**	**Open — scheduled for Q4 2026**
Compliance	PCI 6.4.2 · ISO A 8.9 · SOC 2 CC6.6		
References	OWASP API Security Top 10 — API8:2023 Security Misconfiguration · OWASP WSTG-CLNT-07 · CWE-942 Permissive Cross-domain Policy with Untrusted Domains		

Description

The GraphQL endpoint reflects the requesting origin in its Access-Control-Allow-Origin header and permits credentials, which effectively disables the browser's cross-origin protection for credentialed requests.

The security headers present on REST responses are absent on this endpoint.

Business impact

A page under an attacker's control can issue credentialed requests to the GraphQL endpoint from a victim's browser and read the responses. The impact is bounded by what that victim's session can reach, which after the other fixes is their own tenant — but it is a boundary that should not depend on other fixes.

Remediation

- Replace origin reflection with an explicit allow-list of first-party origins.
- Do not combine a permissive origin policy with credentialed requests under any circumstances.
- Apply the same header set to the GraphQL endpoint as to REST responses, at the gateway so it cannot be omitted.

API-11 Resolver errors disclose internal schema detail

LOW
CVSS 3.5

CVSS v3.1	AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N	**CWE**	CWE-209
Affected asset	GraphQL endpoint — error handling	**Status**	**Risk accepted by client**
Compliance	PCI 6.2.4 · ISO A 8.28 · SOC 2 CC7.1		
References	OWASP API Security Top 10 — API8:2023 Security Misconfiguration · OWASP WSTG-ERRH-01 · CWE-209 Generation of Error Message Containing Sensitive Information		

Description

Resolver errors return messages containing internal type names, table names and occasionally column names. The client accepts the risk, citing developer experience for partner integrations.

The disclosure is bounded but persistent: it re-creates, error by error, much of what disabling introspection is intended to withhold.

Business impact

Reconnaissance value. An attacker who cannot introspect the schema can reconstruct a useful portion of it by provoking errors, which reduces the benefit of the API-06 remediation.

Remediation

- Return a generic error with a correlation identifier and log the detail server-side.
- If partner developer experience is the concern, return detailed errors only to credentials flagged as sandbox or development.
- Re-confirm this acceptance at the next assessment; it is reasonable today and becomes less so as the partner programme grows.

API-12 Outdated dependency in the gateway with no reachable sink

INFORMATIONAL
CVSS —

A dependency in the API gateway is several minor versions behind current and carries a published advisory. The vulnerable code path is not reachable in the deployed configuration, so no exploitation was achievable and no severity is assigned.

It is recorded because dependency currency is a control in its own right — a configuration change could make the sink reachable without anyone noticing that a known-vulnerable version is deployed, and the gateway is the component through which all API traffic passes.

Remediation

- Update the dependency as part of routine maintenance.
- Ensure software composition analysis covers infrastructure components and not only application code.
- Maintain an SBOM for the gateway alongside the application one.

References: OWASP A06:2021 Vulnerable and Outdated Components · CWE-1104 Use of Unmaintained Third Party Components

API-13 No published deprecation or versioning policy

INFORMATIONAL
CVSS —

The platform has no published policy describing how API versions are introduced, supported and retired, and no mechanism for signalling deprecation to consumers. The surviving v1 interface in API-08 is the practical consequence: nothing said it should have been switched off, so nobody did.

It is recorded because inventory management is where most API risk begins. An interface that nobody owns is an interface that nobody patches, and partner integrations in particular will use whatever continues to answer.

Remediation

- Publish a versioning and deprecation policy with a defined support window for each version.
- Signal deprecation in responses using the standard Deprecation and Sunset headers so integrations learn about it programmatically.
- Track consumers per version so retirement is a conversation with named partners rather than a leap.

References: OWASP API Security Top 10 — API9:2023 Improper Inventory Management · RFC 8594 · RFC 9745

14. Remediation Roadmap

Findings are sequenced by risk reduction per unit of effort rather than by severity alone. The client completed the immediate and short-term actions during the retest window.

14.1 Prioritised actions

ID	ACTION	PRIORITY	EFFORT	OWNER
API-02	Default-deny route policy; isolate the admin namespace	Immediate	Medium	Platform engineering
API-01	Tenant scoping at the data-access layer	Immediate	Medium	Platform engineering
API-04	Intersect caller filters with the caller's tenant scope	Immediate	Low	Platform engineering
API-03	Authenticated export delivery; short-lived bound URLs	Short term	Medium	Platform engineering
API-05	Response projections; explicit binding allow-lists	Short term	Medium	Platform engineering
—	Audit coverage for object reads and admin calls	Short term	Medium	Platform engineering
API-08	Decommission the legacy v1 interface	Short term	Low	Infrastructure
API-06	Disable introspection; depth and cost limits	Short term	Medium	Platform engineering
API-07	Rate limits on authentication and export	Medium term	Low	Platform engineering
API-09	Validate and encode upstream responses	Medium term	Medium	Integrations
API-10	Explicit CORS allow-list; headers at the gateway	Medium term	Low	Platform engineering
API-11	Generic errors with correlation identifiers	Routine	Low	Platform engineering
API-12	Update the gateway dependency; SCA and SBOM	Routine	Low	Infrastructure
API-13	Publish a versioning and deprecation policy	Routine	Low	Product + platform

14.2 Structural recommendations

Four of the five most serious findings share one root cause. Patching five endpoints resolves this report; it does not resolve the next one.

- **Centralise authorisation.** One layer that resolves identity, tenant and role from the credential and applies them at the data boundary would have prevented API-01, API-02, API-04 and half of API-05.

- **Default to deny.** A route with no declared role requirement should be unreachable, and the application should refuse to start if one exists. This is the single change that makes an undocumented endpoint harmless.
- **Treat bulk interfaces as a separate risk class.** Export, search and batch endpoints accept parameters ordinary endpoints do not, and move far more data when they are wrong. They deserve their own review and their own tests.
- **Reconcile deployed surface against documented surface continuously.** API-02 and API-08 were both found because the deployed route table differs from the specification. A scheduled diff finds those internally.
- **Make authorisation testable.** Pipeline tests asserting cross-tenant and cross-role refusal for every object-returning endpoint turn an invisible class of defect into a failing build.

15. Retest Results

The client remediated during and after the engagement and a formal retest was performed on 15–16 September 2026. Each finding was re-tested using the original reproduction steps, and each fix was probed for bypass rather than merely confirmed present.

ID	FINDING	RESULT	VERIFICATION
API-01	Broken object level authorisation	**Resolved**	Tenant now derived from the credential and applied in the data-access layer. Retested across all fourteen object types, both tenants and all five identities, including header manipulation and parameter pollution.
API-02	Admin namespace reachable	**Resolved**	Middleware default changed to deny and the application now fails to start with an undeclared route. Namespace moved behind an internal gateway. Retested with all customer and partner identities.
API-03	Pre-signed export URLs	**Resolved**	Exports now stream through an authenticated endpoint; residual signed URLs are single-use and valid for five minutes. Issuance and retrieval logged.
API-04	Cross-tenant export filter	**Resolved**	Filters intersected with the caller's scope and validated against a field allow-list. The effective filter is recorded with the job.
API-05	Data exposure and mass assignment	**Resolved**	Response projections defined per endpoint and role; binding allow-lists applied. Nineteen undocumented properties retested; all rejected.
API-06	GraphQL introspection and limits	**Resolved**	Introspection disabled in production; depth limit of 8 and a cost budget applied; aliasing and batching capped; field-level authorisation corrected.
API-07	Rate limiting	**Resolved**	Limits applied at the gateway on authentication and export, with progressive delay. New endpoints now inherit limits by default.
API-08	Legacy v1 interface	**Resolved**	Decommissioned 9 September 2026 and confirmed unreachable at retest.
API-09	Unsafe upstream consumption	**Resolved**	Schema validation and size limits applied; redirects refused; upstream content encoded on output.
API-10	CORS and headers on GraphQL	**Open**	Scheduled for Q4 2026. Cetonix recommends bringing the credentialed-origin change forward independently of the header work.
API-11	Error detail disclosure	**Accepted**	Client accepts the risk, citing partner developer experience. Re-confirmation

ID	FINDING	RESULT	VERIFICATION
			recommended at the next assessment.
API-12	Outdated gateway dependency	**Open**	Scheduled in the routine maintenance cycle.
API-13	No deprecation policy	**Accepted**	Client accepts for now; a policy is committed for the next planning cycle.

Post-retest position

Both Critical findings and all three High findings are resolved, and the two structural changes that mattered most — data-layer tenant scoping and a default-deny route policy — were implemented rather than worked around. Residual risk is limited to two Low and two Informational items, of which two are accepted with compensating controls.

15.1 Findings shared with other engagements on this platform

This API serves the web application assessed under CTX-PT-2026-0418 and the two mobile clients assessed under CTX-PT-2026-0431 and 0437. Two weaknesses recur across those engagements: tenant scoping derived from client-supplied data, and server-side acceptance of client-asserted entitlements. Both were fixed once here and verified from each client in turn.

Where an organisation runs several clients against one API, testing the API directly is the only way to see the surface none of the clients exercises — and, on the evidence of this engagement, that is where the Critical findings were.

Appendix A — Severity and Scoring Methodology

Severity is assigned from CVSS v3.1 base scores, adjusted for the client's environment and reviewed independently before issue. Where the environmental context materially changes the risk, the adjustment and its reasoning are stated in the finding.

SEVERITY	CVSS	DEFINITION AND EXPECTED RESPONSE
Critical	9.0 – 10.0	Direct compromise of sensitive data or platform integrity, exploitable with little effort and no special access. Notified within two hours of confirmation; remediation expected immediately.
High	7.0 – 8.9	Significant compromise, or a reliable step toward one. Notified within one business day; remediation expected within 30 days.
Medium	4.0 – 6.9	Meaningful weakening of the security posture, typically requiring a precondition or chaining. Remediation expected within 90 days.
Low	0.1 – 3.9	Limited direct impact; contributes to attacker capability or defence in depth. Address in routine maintenance.
Informational	—	No direct security impact at present; recorded for completeness, hygiene or future relevance.

Business impact is assessed separately from technical severity. A technically modest flaw affecting regulated data may be escalated, and a technically severe flaw in an isolated component may be moderated. Every adjustment is recorded in the finding and validated by the QA reviewer.

Appendix B — Tooling

Tools support coverage and discovery. No finding in this report rests on tool output alone; each was reproduced manually before inclusion.

CATEGORY	PURPOSE	NOTES
Intercepting proxy	Request interception and manipulation	Primary manual testing platform for all phases
Specification-driven harness	Exercising every documented endpoint across every role	Driven from the client's OpenAPI specification and GraphQL schema
GraphQL assessment tooling	Introspection, depth, complexity, aliasing and batching tests	Schema retained for analysis only; not reproduced in this report
Authorisation matrix harness	Replaying every request as every role and both tenants	Written for this engagement; the core of the multi-tenant testing
Traffic capture from first-party clients	Discovering undocumented endpoints	Web and mobile clients proxied to map real usage against the specification
Rate and resource-consumption tooling	Measuring throttling and cost controls	Capped by the rules of engagement, not by server-side controls

Appendix C — Interface Inventory

The interfaces assessed, reconciled against the client's published specification. Host identifiers for undocumented surface are redacted in this copy.

INTERFACE	TYPE	DOCUMENTED	DISPOSITION
/api/v2/* (148 endpoints)	REST	**Yes**	Tested in full across five identities
/graphql	GraphQL	**Partial**	84 types tested; introspection disabled at retest
/api/v2/exports	Async API	**Partial**	API-03, API-04 — both resolved
/api/v2/webhooks	Outbound	**Yes**	Destination validation sound — no issue
/api/v2/admin/*	REST	**No**	API-02 — moved behind an internal gateway
[redacted] (legacy v1)	REST	**No**	API-08 — decommissioned 9 Sep 2026
Partner integration keys (6)	Credential	**Yes**	Scope enforcement corrected; re-issue recommended

Appendix D — Glossary

BOLA	Broken Object Level Authorization — returning an object to a caller who is not entitled to it, because the server checks that the object exists rather than that the caller owns it. The flaw in API-01.
BFLA	Broken Function Level Authorization — allowing a caller to invoke a function reserved for a higher privilege level, typically because the route is not covered by a role check. The flaw in API-02.
BOPLA	Broken Object Property Level Authorization — returning or accepting individual object properties the caller should not see or set. The flaw in API-05.
Introspection	A GraphQL feature that returns the full schema on request. Useful in development; an inventory for an attacker in production.
Pre-signed URL	A time-limited URL granting direct access to stored content without further authentication. Convenient, and dangerous when the validity window or scope is too broad.
Black-box testing	Testing with no credentials and no prior knowledge, simulating an external attacker.
Grey-box testing	Testing with credentials at each role level, simulating an attacker who has obtained an account.
CVSS	Common Vulnerability Scoring System — an industry-standard method of expressing technical severity as a score and a vector string.
CWE	Common Weakness Enumeration — a catalogue of software weakness types, used to classify the underlying defect.

Rules of engagement	The written agreement defining what may be tested, when, how aggressively, and what is prohibited.

Appendix E — Attestation Letter

The letter below may be shared with auditors, assessors, customers and prospects. It confirms that testing was performed and that findings were remediated, without disclosing technical detail about the platform.

CETONIX

SECURITY ASSURANCE · OFFENSIVE SECURITY PRACTICE

18 September 2026

LETTER OF ATTESTATION — PENETRATION TEST

To whom it may concern,

Cetonix was engaged by ACME Corporation, Inc. to perform an independent penetration test of the ACME Platform application programming interface, comprising the REST v2 interface and the GraphQL endpoint served at api.acme-corp.example. Testing was conducted between 31 August and 11 September 2026 under engagement reference CTX-PT-2026-0444.

The assessment comprised authenticated testing across five caller identities, including two customer tenants and a partner integration credential, applied to 148 documented REST endpoints, 84 GraphQL types and the asynchronous export and webhook interfaces. The methodology followed the OWASP API Security Top 10 (2023), the OWASP Web Security Testing Guide v4.2 and the Application Security Verification Standard v4.0.3, within the phase structure of PTES and NIST Special Publication 800-115. All findings were manually verified by a named analyst and independently peer-reviewed prior to issue.

Thirteen findings were identified, comprising two of Critical severity, three of High severity, four of Medium severity, two of Low severity and two of an informational nature. A remediation retest was performed on 15 and 16 September 2026. All findings rated Critical and High have been remediated and independently verified as resolved. Remaining items are of Low or informational severity and are either scheduled for remediation or formally accepted by the client with compensating controls in place.

This engagement satisfies the annual penetration testing obligation under PCI DSS v4.0.1 Requirements 11.4.2 and 11.4.3 in respect of the assessed interfaces, and provides technical testing evidence supporting ISO/IEC 27001:2022 Annex A 8.8 and A 8.29, SOC 2 Trust Services Criteria CC6.1, CC6.3 and CC7.1, and the evaluation requirement at 45 CFR §164.308(a)(8) of the HIPAA Security Rule.

This letter reflects the security posture of the assessed environment at the conclusion of the retest window. It does not constitute a guarantee against future compromise, and it does not extend to changes made to the environment after that date.

Yours faithfully,

[Name], Practice Manager — Offensive Security
Cetonix · info@cetonix.com · Verification of this letter may be requested at the address above, quoting CTX-PT-2026-0444.

END OF REPORT