Web application penetration test

Prepared for Paul Allen | Example target: leadraptor.io | 18 November 2026 | Version 1.0

> **Read this first.** This is a fictional worked example of a paid assessment report. Both findings, all requests and all results are invented. LeadRaptor has not been penetration tested for this document. Its actual security posture is **not assessed**.

Executive summary - fictional engagement

In this example, a lead-management application has two weaknesses: a client can retrieve another client's lead record, and untrusted lead text is interpreted as a spreadsheet formula when exported. The first issue would put customer contact information at risk. The second would affect staff who open exported files in a spreadsheet.

The priority would be to enforce account ownership on every lead-data request, then make exported text safe for the supported spreadsheet applications. The example demonstrates how a report connects technical evidence to a concrete development task.

Illustrative finding	Severity	Business consequence
EX-01 Cross-client lead disclosure	**HIGH**	A signed-in client obtains a lead belonging to another client.
EX-02 Formula injection in lead exports	**MEDIUM**	Attacker-supplied lead text becomes a formula in a staff spreadsheet.

Management decision in this scenario

EX-01: prioritize containment and an ownership-check fix before expanding portal access. **EX-02:** make the export safe before staff resume the affected spreadsheet workflow. Confirm both fixes through an independent retest.

These are two example findings, not a complete finding inventory. No numerical security score, compliance verdict or claim that other vulnerability classes are absent is made.

Report navigation

02 Scope and limitations
03 Detailed finding: cross-client lead disclosure
04 Detailed finding: formula injection
05 Coverage and assessment method
06 Remediation, retesting and references

Scope and engagement boundaries

An example of the scope information that should accompany every client report.

Actual work used to prepare this example

Read LeadRaptor's local project overview and retrieved its public homepage through a web research tool. The retrieved homepage appeared older than the current project documentation, so it was not used as evidence of current functionality. No authenticated sessions, scanners, exploitation, form submissions or customer-record access were performed.

Proposed scope for a real assessment

Surface	Example boundary and status
Public website	leadraptor.io public pages and approved intake routes. Passive/active checks would be specified in the signed scope. Not tested here.
Client portal	Documented entry: /dashboard/login and /dashboard. Two isolated client accounts with synthetic leads. Not tested here.
Staff and administration	Documented entry: /login, /staff and /admin. Agreed test accounts and permission matrix. Not tested here.
Application APIs	Only explicitly listed LeadRaptor-owned endpoints and approved database access paths. Paths on pages 3-4 are invented.
Third-party services	Provider infrastructure, other tenants and external integrations excluded unless separately authorized.

Example access and rules of engagement

Use client A and client B, one restricted staff user and one administrator, all dedicated test identities. Seed synthetic records with known owners. Confirm the host list, environment, build, testing window, source IPs, request limits and emergency contact before testing.

Restrict changes to designated test records. Stop on unexpected real-data exposure, service degradation or scope uncertainty. Exclude denial of service, phishing and destructive tests unless separately agreed. Obtain explicit approval before triggering emails, bookings or other downstream actions.

> **Context is not a finding.** The project overview describes client, staff and admin portals and server-side access controls. This sample does not evaluate those controls. The fictional flaws below do not assert that the documented controls are missing.

Cross-client lead disclosure

EX-01 | HIGH - illustrative | Broken object-level authorization | WSTG-ATHZ-04

> **FICTIONAL FINDING.** The endpoint, identities, record and response below are invented. No request shown on this page was sent to LeadRaptor.

Example affected component: lead-detail API.
Prerequisites: a valid client A session and a valid identifier for a lead owned by client B. Identifier discovery and bulk enumeration are not demonstrated.

Description and business impact

In the fictional result, a client A session retrieves a client B lead without an ownership check blocking access. A real occurrence could disclose contact information and sales context across customers. The example demonstrates one unauthorized read; it does not establish write access, bulk extraction or administrator access.

Reproduction using controlled test accounts

1. Create synthetic lead **demo-lead-b-001** under test client B.
2. Confirm that client A has no assignment or shared access to client B.
3. Use client A's session to request that known client B record.
4. Compare the response with the seeded record and repeat with an authorized client B session as the positive control.

```
# SYNTHETIC EVIDENCE - NOT A LIVE REQUEST
GET /api/example/leads/demo-lead-b-001 HTTP/1.1
Host: leadraptor.io
Authorization: Bearer <fictional-client-A-session>

HTTP/1.1 200 OK
{"id":"demo-lead-b-001", "client_id":"demo-client-B",
 "email":"synthetic-lead@example.invalid"}
```

Expected: deny access without disclosing the record (for example, 403 or a consistent 404). **Illustrative result:** 200 with the other client's synthetic data. A 200 status alone is not proof: ownership and returned content must also be verified.

Remediation and acceptance criteria

Authenticate the requester and authorize access to the requested object on the server. Apply the same ownership rules to detail, search, export and mutation paths. Use database policies as defense in depth where appropriate; hiding controls or using unguessable IDs is insufficient.

Retest: client A is denied client B data; client B retains legitimate access; staff access follows the assignment matrix. Exercise the fixed route and related access paths with synthetic records. **Actual status:** not tested.

Formula injection in exports

EX-02 | MEDIUM - illustrative | Untrusted data interpreted as spreadsheet formulas

> **FICTIONAL FINDING.** The export feature and behavior are hypothetical. No file was exported from LeadRaptor and no spreadsheet exploit was executed.

Example affected component: CSV lead export.
Prerequisites: an external user can supply a lead field; staff export that field and open the file in a spreadsheet that evaluates it as a formula.

Description and business impact

In this scenario, a lead name passes into a CSV cell without being made safe for spreadsheet interpretation. Opening the file treats the value as a formula rather than literal text. This can corrupt staff analysis. Further effects depend on the spreadsheet application and its security settings; data theft and code execution are not demonstrated.

Safe reproduction and evidence example

1. Add the benign value **=1+1** to a synthetic lead-name field.
2. Export only that test lead using the proposed CSV workflow.
3. Open it in an isolated spreadsheet environment with external connections disabled; record the application version and import settings.
4. Check the cell type and formula bar, not just its displayed value.

```
# INVENTED CSV CONTENT
lead_id,name,email
demo-lead-001,=1+1,synthetic-lead@example.invalid

# INVENTED SPREADSHEET OBSERVATION
Cell B2 displays: 2
Cell B2 formula:  =1+1
Expected: literal text "=1+1", not an active formula
```

Evidence limit: this benign test would demonstrate formula interpretation only. A real report must name the tested application/version and attach the sanitized export and cell evidence. Neither exists for this sample.

Remediation and acceptance criteria

Prefer spreadsheet output with untrusted fields explicitly stored as text and formula auto-detection disabled. If CSV is required, choose escaping for the supported consumers and validate it through open, save and reopen. Quoting CSV fields alone is not a sufficient guarantee.

Retest: formula-like text remains inert after the supported export/import lifecycle, including separator, quote and control-character cases. Legitimate names and phone numbers remain usable. Medium is a scenario-specific rating; reassess it against actual impact and user interaction. **Actual status:** not tested.

Coverage, method and limitations

A findings list should be accompanied by evidence of what was actually examined.

Coverage register for this document

Area	Example real-engagement checks	Actual status
Public surface	Approved host inventory; TLS/HTTP configuration; exposed public content.	Not tested
Authentication	Sign-in and recovery flows; session lifecycle; expiry and logout.	Not tested
Authorization	Client A/B isolation; staff assignments; role and object access checks.	Not tested
Input and output	Injection contexts; stored content; data import and export behavior.	Not tested
Business logic	Permitted lead updates; sales/report integrity; workflow bypass attempts.	Not tested
Integration boundary	Approved webhook authentication, replay handling and tenant routing.	Not tested
Source and data layer	Only if included: sensitive paths, database policies and privileged access.	Not tested

This register does not claim OWASP coverage was completed. A real report should add test-case IDs, dates, environment/build, role pairs, evidence references and one of: tested / blocked / not applicable / not tested. Record a limited negative result as "no issue observed in the tested case," never as a guarantee.

Example delivery method

Agree scope and business risks; inventory the approved application; use automation for repeatable checks; manually validate candidates and exercise authorization and business logic; prepare reproducible evidence; prioritize remediation; retest the agreed fixes. OWASP WSTG v4.2 can organize selected web test cases [1].

Severity and evidence standards

Rate impact, exploit prerequisites, required privileges and user interaction together. High indicates serious exposure under the stated assumptions; Medium indicates meaningful but conditional impact. The sample uses qualitative ratings, not CVSS scores. Each real finding should distinguish reproduced impact from plausible follow-on effects.

Automation or AI output is a lead for investigation. A tester must confirm identity, ownership, exploit conditions and evidence before reporting a vulnerability. A production report should record the actual tools and versions used; this sample has no scanner results.

> **Limits of assurance:** a real assessment is bounded by its time, scope and access. This example provides no security assurance or regulatory attestation. It cannot be submitted as evidence that LeadRaptor passed a penetration test.

Remediation and retesting

Example priorities and completion criteria - no remediation work has been requested or performed.

ID / priority	Suggested action in the fictional scenario	Owner / actual state
EX-01 Urgent	Investigate promptly; contain the affected read path if confirmed. Target an initial response within 48 hours, subject to agreed severity handling.	Application owner Not tested
EX-02 Planned	Fix the export path and validate supported spreadsheet behavior. Illustrative target: within 14 days.	Application owner Not tested

Retest record to include in a real delivery

Record the finding ID, fixed build, environment, date, tester, original reproduction, regression cases and evidence references. Use **fixed** only when the original issue no longer reproduces and the relevant controls still work. Otherwise record **partially fixed**, **not fixed** or **unable to verify**, with the reason.

A retest covers the agreed fixes and adjacent paths; it does not automatically reassess the entire product. Keep the original finding and append the retest outcome so the client can follow the history.

Evidence and client handover

For a real engagement, provide a business summary and a developer walkthrough, plus restricted evidence files identified by finding ID. Include request/response pairs, role and ownership checks, timestamps, affected versions and screenshots where useful. Redact tokens and personal data; preserve originals only in an agreed secure location with an agreed retention period.

References and provenance

[1] OWASP Web Security Testing Guide v4.2
wstg.owasp.org/v4.2/
Methodology reference; no claim that its test suite was executed.

[2] OWASP WSTG-ATHZ-04: Insecure Direct Object References
OWASP authorization test guidance
Background reference for EX-01; the scenario and evidence are synthetic.

[3] OWASP: CSV Injection
owasp.org/www-community/attacks/CSV_Injection
Background reference for EX-02; consumer behavior must be tested.

LeadRaptor context: local project guide, Portals and Security controls sections, updated 18 November 2026; leadraptor.io public homepage retrieved on the same date. These establish context only. No customer data or actual vulnerability evidence is included.

> **How to use this sample:** show the structure and level of detail. Before issuing a genuine report, replace every fictional item with validated results and complete the actual scope, test record and evidence references. Never relabel this document as a completed assessment.