



Penetration Testing

Presentato da:

- Tommaso Arlati
- Diego Margini
- Alessandro Pullega
- Aldo Ferhati
- Elvis Ede

Introduzione

Il presente report fornisce un'analisi approfondita su diverse vulnerabilità rilevate all'interno di un'applicazione web, basandosi sui test di sicurezza effettuati. Sono stati individuati diversi punti deboli che potrebbero compromettere la riservatezza, l'integrità e la disponibilità del sistema e dei dati trattati. Verranno descritti gli attacchi, i rischi associati, prove pratiche, valutazioni di rischio e le relative contromisure.

Membri e Ruoli

- Tommaso Arlati : PM, stilatore documentazione;
- Aldo Ferhati: Programmatore, Penetration tester;
- Alessandro Pullega: Vulnerability Assester;
- Diego Margini: Costilatore, Reportista;
- Elvis Ede: Coprogrammatore.

MATERIALI

01

Docker
(containerizzazione)

02

Debian (MV per la
webapp mirata al
realismo dell'attacco)

03

Kali Linux (MV pentest),
Windows 10, Burp

04

VMware

I Passi Per Mettere In Sicurezza



1. Information Gathering (WSTG-INFO)

Identificare l'ambiente tecnologico dell'applicazione
Eseguire una mappatura delle funzionalità esposte
Scoprire sottodomini e API



2. Configuration and Deployment Management Testing (WSTG-CONF)

Verifica di errori di configurazione
Directory listing attivo
Configurazioni di debug o ambienti di test lasciati esposti



3. Identity Management Testing (WSTG-IDNT)

Controlli su autenticazione (es. enumerazione utenti, brute force)
Verifica del cambio password, login, reset password



4. Authentication Testing (WSTG-ATHN)

Test sull'autenticazione a più fattori
Token di sessione deboli o prevedibili
Bypass dell'autenticazione

I Passi Per Mettere In Sicurezza



5. Authorization Testing (WSTG-ATHZ)

Verifica degli accessi orizzontali e verticali

Accesso non autorizzato a risorse



6. Session Management Testing (WSTG-SESS)

Analisi dei cookie di sessione

Timeout di sessione

Token di sessione non rigenerati dopo il login/logout



7. Input Validation Testing (WSTG-INPV)

SQL Injection, XSS, Command Injection

File Inclusion, Buffer Overflow

Validazione lato client vs lato server



8. Testing for Error Handling (WSTG-ERRH)

Comportamento in caso di eccezioni

Informazioni sensibili rivelate in messaggi di errore

I Passi Per Mettere In Sicurezza



9. Business Logic Testing (WSTG-BUSL)

Comportamenti anomali sfruttabili (es. prezzi negativi)

Manipolazione di flussi logici dell'applicazione



10. Client-Side Testing (WSTG-CLNT)

Sicurezza dei componenti JavaScript

Vulnerabilità nei framework front-end Cross-Origin

Resource Sharing (CORS), DOM XSS

L'esecuzione



PROGRAMMAZIONE

A CURA DI: A. FERHATI (LEAD PROGRAMMER), E. EDE (CO-PROGRAMMER)



A. Ferhati

- Setup VM
- Analisi Html
- Scoperta scoreboard (pagina segreta)
- Analisi di codice
- “Guadagno” accesso FTP
- Injection Html e Javascript
- Accesso metriche del sito (tramite Prometheus)

E. Ede

- Stilazione branch di Git

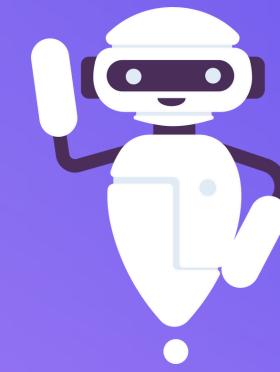
VULNERABILITY ASSESSMENT

A CURA DI: A. PULLEGA



PENETRATION TESTING

A CURA DI: A. FERHATI



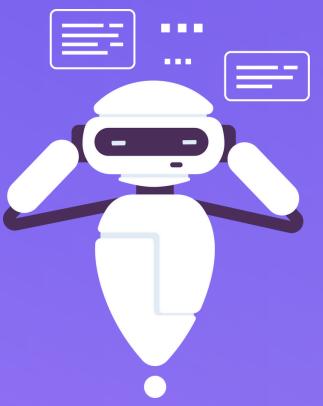
RECONNAISSANCE & ENUMERATION

- Scansione delle porte e servizi.
- Identificazione del sistema operativo.
- Ricerca di configurazioni note.
- Enumeration di utenti, condivisioni, directory, ecc.



EXPLOITATION

- Esecuzione di exploit noti su vulnerabilità scoperte.
- Accesso a sistemi, servizi o dati sensibili.
- Privilege escalation (da utente a root/admin).



POST-EXPLOITATION & REPORTING

- Raccolta di prove (accessi, file letti, comandi eseguiti).
- Mantenimento dell'accesso (opzionale, solo in ambito autorizzato).
- Creazione di un report con le tecniche usate e le vulnerabilità sfruttate.
- Eventuale proposta di contromisure.

REPORTAGE

A CURA DI: D. MARGINI

01

- Condivisione preliminare dei risultati trovati durante il Pen Test e la VA.
- Confronto tecnico tra chi ha condotto i test e gli sviluppatori/sistemisti.
- Verifica delle vulnerabilità realmente sfruttabili vs. false positive.
- Discussione sull'origine delle vulnerabilità: codice, configurazioni, patch mancanti.
- Raccolta di feedback tecnico su priorità, contesto aziendale e fattibilità delle mitigazioni.

02

- Executive Summary: sintesi comprensibile anche per i manager.
- Dettagli tecnici delle vulnerabilità: descrizione, CVE, impatto, prove (screenshot, log).
- Matrice del rischio: classifica le vulnerabilità secondo:
- Impatto (alto/medio/basso)
- Probabilità di sfruttamento
- Priorità di intervento
- Raccomandazioni tecniche e organizzative per ogni vulnerabilità trovata.
- Cronologia delle attività, strumenti usati, metodologie.



MANAGEMENT E DOCUMENTAZIONE

A CURA DI: T. ARLATI.

- Pianificazione del progetto
 - Definisce gli obiettivi del test (es. compliance, sicurezza applicativa, audit).
 - Stila il project plan con timeline, milestone, e risorse.
 - Coordina il kickoff con tutti gli attori: penetration tester, vulnerability analyst, sviluppatori, IT, management.
 -
- Coordinamento operativo
 - Assegna attività ai membri del team (es. chi fa la scansione, chi l'exploit).
 - Mantiene aggiornato il Gantt o altra timeline.
 - Supervisiona la raccolta dei dati, report parziali, progressi tecnici.
 - Facilita il dialogo tra figure tecniche e non tecniche.
- Gestione della comunicazione.
 - Conduce i meeting di aggiornamento (es. stand-up o weekly sync).
 - Valida e distribuisce la documentazione tecnica e il report finale.
- Chiusura del progetto



**Grazie per
L'attenzione!
Contattatemi**

Tommaso Arlati

Project manager

 +39 351 540 5633

 tommaso.arlati@fitstic-edu.com

 www.sitofinto.com