Tugas Lab 1 — Keamanan Aplikasi Web (VulnLab)

Menemukan dan Memitigasi Kerentanan pada Aplikasi Web Latihan (VulnLab)

Tujuan Pembelajaran

Setelah menyelesaikan tugas ini mahasiswa diharapkan dapat:

1. Mengidentifikasi dan mengeksploitasi kerentanan umum pada aplikasi web (SQL Injection, Stored XSS, CSRF).
2. Menjelaskan mengapa kerentanan muncul (root cause).
3. Mengimplementasikan mitigasi praktis untuk setiap kerentanan.
4. Menyusun laporan teknis singkat yang berisi bukti eksploitasi, langkah perbaikan, dan rekomendasi deployment yang aman.

Lingkup Aplikasi

Anda akan menggunakan paket `vuln-flask-app.zip` yang diberikan oleh dosen. Aplikasi ini:

- Menyediakan halaman publik dan halaman private (butuh login).
- Memiliki fitur login, profile (kolom komentar), transfer antar pengguna, dan halaman admin/debug (tergantung versi).
- Aplikasi sengaja rentan untuk latihan — jalankan **hanya** di lingkungan lokal/terisolasi.

Instruksi Umum untuk Mahasiswa

1. Ekstrak paket dan jalankan aplikasi sesuai petunjuk (Docker atau `flask run`).

2. Temukan kerentanan yang tersembunyi di aplikasi (lihat checklist di bawah). Lakukan eksploitasi untuk membuktikan kerentanan — dokumentasikan langkah dan bukti.

3. Implementasikan perbaikan pada kode (branch baru git atau salinan proyek).

4. Buat laporan singkat (maks 4 halaman) berisi: ringkasan temuan, bukti eksploitasi (screenshot/HTTP requests), perubahan kode (diff atau snippet), dan rekomendasi tambahan.

5. Kumpulkan:
    a. Repo atau zip kode perbaikan
    b. laporan PDF/Markdown,
    c. instruksi singkat cara menjalankan versi yang sudah diperbaiki.

Deliverables (yang harus dikumpulkan mahasiswa)

1. `repo` atau ZIP berisi:
    - Source code perbaikan (branch atau folder `fixed`), termasuk file template/perubahan relevant.
    - Instruksi singkat menjalankan aplikasi (README).
2. Laporan (PDF/Markdown) berisi:
    - Ringkasan temuan (apa yang ditemukan, di mana).
    - Langkah eksploitasi dan bukti (screenshot atau curl/httpie request).
    - Perubahan kode (diff + penjelasan).
3. (Opsional) Video pendek 2–3 menit demo exploit & fix — poin ekstra.

Panduan Teknis (perintah berguna)

- Jalankan lokal:
    - Tanpa Docker: buat virtualenv → `pip install -r requirements.txt` → `python db_init.py` → `flask run`
    - Dengan Docker Compose: `docker compose up --build`
- Cek request dengan browser DevTools (Network → Fetch/XHR), atau gunakan `curl`/`httpie` untuk merekam requests.
- Untuk melihat DB SQLite: `sqlite3 app.db` → `SELECT * FROM users;`

Rubrik Penilaian (total 100 poin)

1. **Identifikasi Kerentanan (30 poin)**
    - Menemukan dan menyebutkan lokasi setiap kerentanan target (SQLi, XSS, CSRF): 15 poin (5 poin tiap kerentanan).
    - Memberi bukti minimal (screenshot/request) tiap temuan: 15 poin.
2. **Eksploitasi (25 poin)**
    - Eksploit yang berhasil untuk SQLi: 8 poin.
    - Eksploit XSS (payload men-trigger JS alert atau DOM mod): 8 poin.
    - Eksploit CSRF (terjadi via request dari attacker page): 9 poin.
3. **Perbaikan & Kode (30 poin)**
    - Perbaikan SQLi (10 poin).
    - Perbaikan XSS (10 poin).
    - Perbaikan CSRF (10 poin).
    - Nilai penuh jika perbaikan benar-benar diterapkan dan diuji.
4. **Dokumentasi & Presentasi (10 poin)**
    - Laporan singkat jelas, langkah reproducible, diffs/commit logs, menjalankan instruksi: 7 poin.
    - Kualitas penulisan, kesimpulan dan rekomendasi tambahan: 3 poin.
5. **Ekstra (maks +5 poin)**
    - Bonus: video demo (maks 5 poin).

Total maksimal: 100 + 5 bonus.

Template Laporan (format yang direkomendasikan untuk mahasiswa)

1. Judul, Kelompok, Nama anggota dan NIM, Tanggal
2. Ringkasan singkat (1 paragraf)
3. Lingkup & asumsi lab
4. Temuan (untuk tiap kerentanan: lokasi, bukti, request/response)
5. Perbaikan yang diterapkan (kode/diff dan penjelasan singkat)
6. Testing ulang (screenshots/requests setelah fix)
7. Referensi/tools yang digunakan jika ada

Catatan Etika & Keamanan

- **Tegaskan kepada mahasiswa**: eksperimen hanya di lab terisolasi. Dilarang menguji aplikasi nyata tanpa izin.
- Jangan menyimpan atau membagikan kredensial asli user di luar lab.
- Untuk demonstrasi XSS, gunakan payload yang benign (mis. `alert('x')`) bukan pencurian cookie.
