Modul 4: Finding Common Vulnerabilities

Pelajari cara mengenali kerentanan yang paling sering menghasilkan valid report dalam program bug bounty dan pengujian keamanan aplikasi web.

EDY SUSANTO - FOUNDER C-SIX SECURITY

Tujuan Pembelajaran

Setelah menyelesaikan modul ini, peserta akan memiliki kemampuan praktis untuk mengenali, menganalisis, dan memvalidasi kerentanan umum yang sering ditemukan dalam aplikasi web nyata.

Kenali Pola

Memahami pola kerentanan umum yang sering muncul di aplikasi web modern.

Praktik Langsung

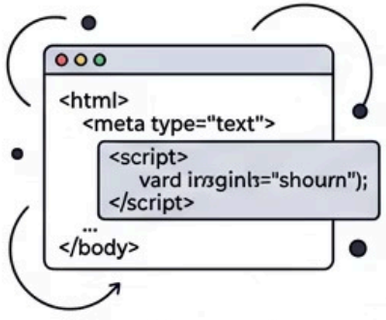
Latihan melalui lab simulasi dan analisis bug sederhana secara hands-on.

Validasi Temuan

Mampu memvalidasi temuan agar layak dijadikan laporan kerentanan yang valid.

Peta Materi Modul 4

Delapan topik kerentanan utama yang akan dipelajari, mulai dari yang paling umum hingga konsep yang lebih teknis.



XSS (Lintas Situs Scripting)

Skrip berbahaya disuntikkan ke dalam halaman web

IDOR (Referensi Objek Langsung Tidak Aman)

Mengakses data pengguna lain dengan mengubah ID

Kontrol Akses Rusak (Broken Access Control)

Pengguna mengakses fitur atau data di luar hak istimewa mereka

Pengungkapan Informasi (Information Disclosure)

Aplikasi secara tidak sengaja membocorkan data sensitif

(Security Misconfiguration)

Salah Konfigurasi Keamanan (Security Misconfiguration)

Pengaturan keamanan yang tidak tepat pada server atau aplikasi

Pengalihan Terbuka (Open Redirect)

Pengalihan Terbuka

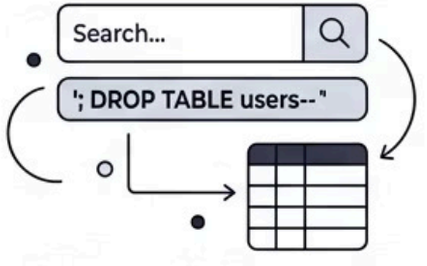
Mengallihkan pengguna ke situs berbahaya menggunakan parameter URL yang tidak divalidasi

Eksposur Data Sensitif (Sensitive Data Exposure)

Eksposur Data Sensitif

Data penting seperti kata sandi atau data kartu kredit tidak terlindungi dengan baik

Konsep Dasar Injeksi SQL



Konsep Dasar Injeksi SQL

Masukan yang tidak divalidasi digunakan untuk memanipulasi kueri database

KERENTANAN #1

Cross-Site Scripting (XSS)

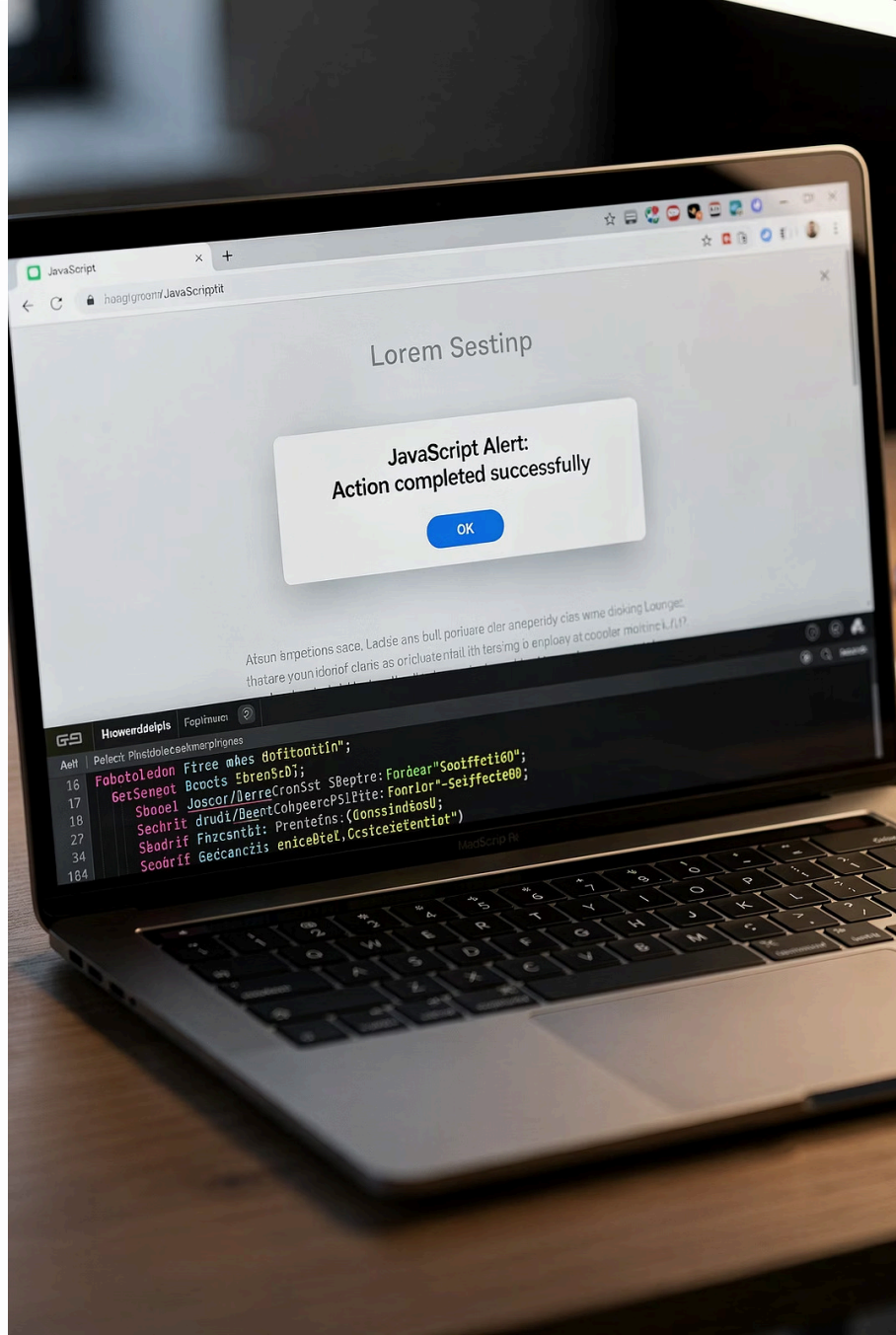
XSS terjadi ketika penyerang berhasil menyisipkan skrip berbahaya ke dalam halaman web yang kemudian dieksekusi oleh browser pengguna lain. Ini adalah salah satu kerentanan paling umum dan sering ditemukan di program bug bounty.

Jenis XSS

- **Reflected XSS** – skrip langsung dari request
- **Stored XSS** – skrip tersimpan di database
- **DOM-based XSS** – manipulasi DOM di sisi klien

Titik Injeksi Umum

- Form pencarian dan input teks
- Parameter URL yang ditampilkan ulang
- Komentar dan profil pengguna
- Header dan cookie yang di-render



KERENTANAN #2

IDOR — Insecure Direct Object Reference

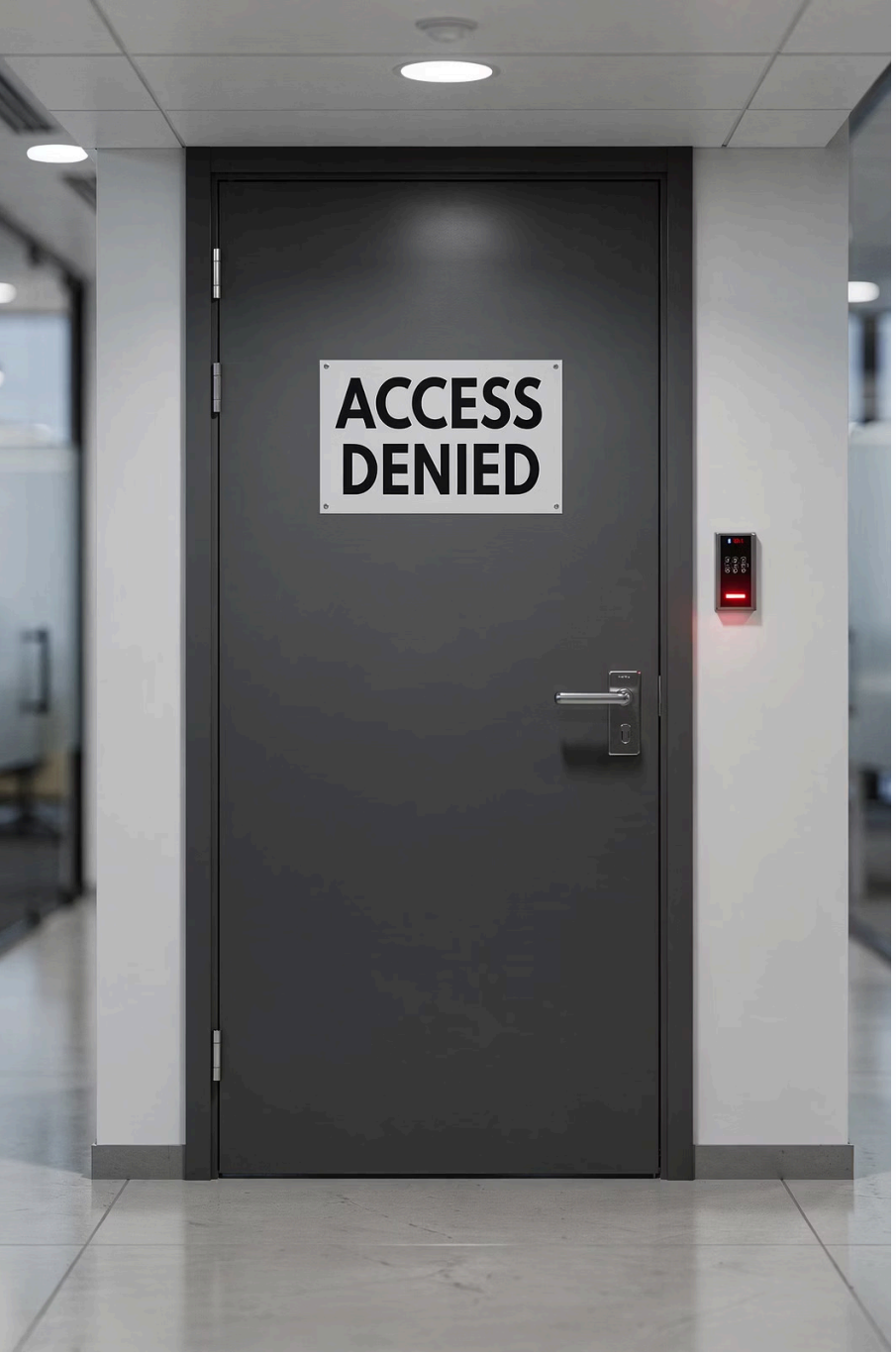
IDOR terjadi ketika aplikasi menggunakan referensi langsung ke objek internal (seperti ID pengguna atau nomor file) tanpa memverifikasi apakah pengguna berwenang mengaksesnya. Temuan IDOR sering bernilai tinggi dalam program bug bounty.

Contoh IDOR Klasik

URL seperti `/profile?user_id=1234` — coba ubah angka tersebut. Jika data pengguna lain muncul tanpa error, aplikasi rentan terhadap IDOR.

IDOR pada API

Endpoint API seperti `/api/orders/5678` yang mengembalikan data order milik pengguna lain tanpa validasi kepemilikan adalah contoh IDOR di level API.



KERENTANAN #3

Broken Access Control

Broken Access Control terjadi ketika batasan akses tidak diterapkan dengan benar, memungkinkan pengguna biasa mengakses fitur atau data yang seharusnya hanya tersedia untuk admin atau peran tertentu.

Akses Fungsi Admin

Pengguna biasa dapat mengakses panel admin hanya dengan mengetikkan URL langsung seperti `/admin/dashboard`.

Manipulasi Parameter Role

Mengubah parameter seperti `role=user` menjadi `role=admin` dalam request untuk mendapatkan hak akses lebih tinggi.

Forced Browsing

Mengakses halaman atau sumber daya tersembunyi yang tidak ditautkan di antarmuka namun dapat diakses secara langsung.

KERENTANAN #4

Information Disclosure

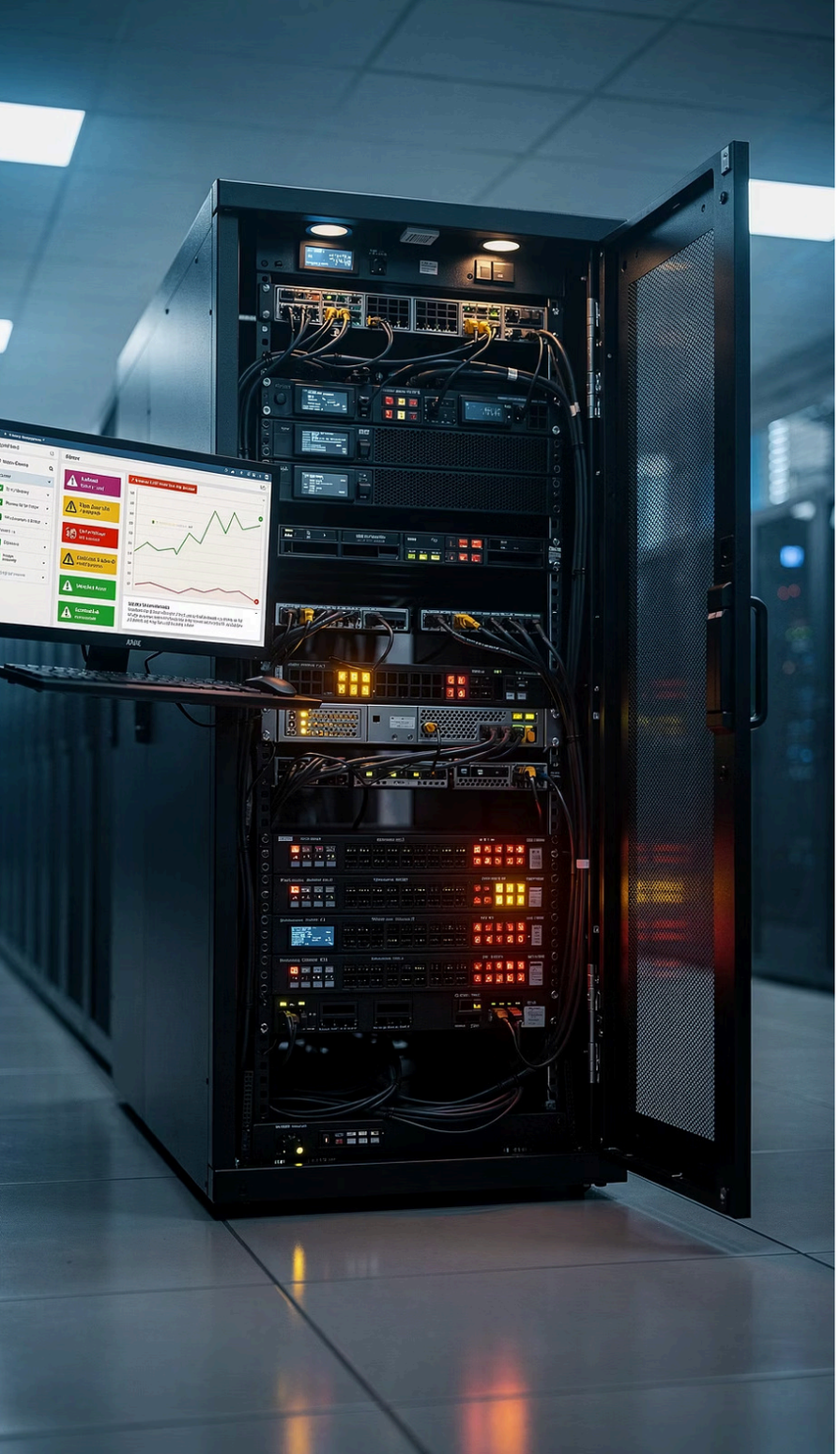
Information Disclosure terjadi ketika aplikasi secara tidak sengaja mengekspos informasi sensitif kepada pengguna yang tidak berwenang, baik melalui pesan error, komentar kode, atau endpoint yang tidak terlindungi.

Sumber Umum Kebocoran

- Pesan error yang mengandung stack trace
- Komentar HTML yang berisi kredensial atau jalur file
- File `robots.txt` dan `.git` yang terbuka
- Response API yang mengandung data internal
- Direktori listing yang aktif

Tips Pencarian

- Cek source code HTML pada setiap halaman
- Periksa response header untuk versi server
- Uji endpoint /debug, /test, /backup
- Gunakan Google Dorks untuk temuan eksternal



KERENTANAN #5

Security Misconfiguration

Security Misconfiguration adalah kategori kerentanan yang terjadi akibat pengaturan keamanan yang tidak tepat pada server, aplikasi, database, atau infrastruktur cloud. Ini adalah salah satu temuan paling umum namun sering diabaikan.

Cloud Storage Terbuka

Bucket S3 atau storage publik yang dapat diakses tanpa autentikasi, mengekspos file sensitif perusahaan.

Default Credentials

Panel admin menggunakan kredensial bawaan seperti `admin/admin` yang tidak pernah diubah setelah instalasi.

Fitur Tidak Diperlukan Aktif

Layanan debug, fitur verbose error, atau port yang tidak diperlukan tetap aktif di lingkungan produksi.

KERENTANAN #6

Open Redirect

Open Redirect terjadi ketika aplikasi menerima URL yang dikontrol pengguna dan mengarahkan pengguna ke URL tersebut tanpa validasi. Meski terlihat sederhana, kerentanan ini dapat digunakan dalam serangan phishing yang canggih.

Contoh Serangan

URL seperti:
https://trusted.com/redirect?url=https://evil.com
Pengguna mempercayai domain awal, lalu diarahkan ke situs berbahaya.

Dampak & Penggunaan

- Digunakan dalam kampanye phishing bertarget
- Bypass whitelist URL pada OAuth flows
- Kombinasi dengan kerentanan lain untuk eskalasi
- Pencurian token autentikasi

⚠ Selalu periksa parameter seperti `redirect_url`, `next`, `return_to`, atau `goto` di setiap aplikasi yang diuji.

KERENTANAN #7

Sensitive Data Exposure

Sensitive Data Exposure terjadi ketika data sensitif seperti password, nomor kartu kredit, atau informasi pribadi tidak dilindungi dengan tepat — baik saat disimpan maupun saat dikirim melalui jaringan.

Data di Transit

- HTTP tanpa enkripsi TLS/HTTPS
- Token dikirim melalui URL (bukan header)
- Data sensitif dalam parameter GET

Data di Penyimpanan

- Password disimpan plaintext atau MD5
- Enkripsi lemah atau kunci yang hardcoded
- Database backup tanpa enkripsi

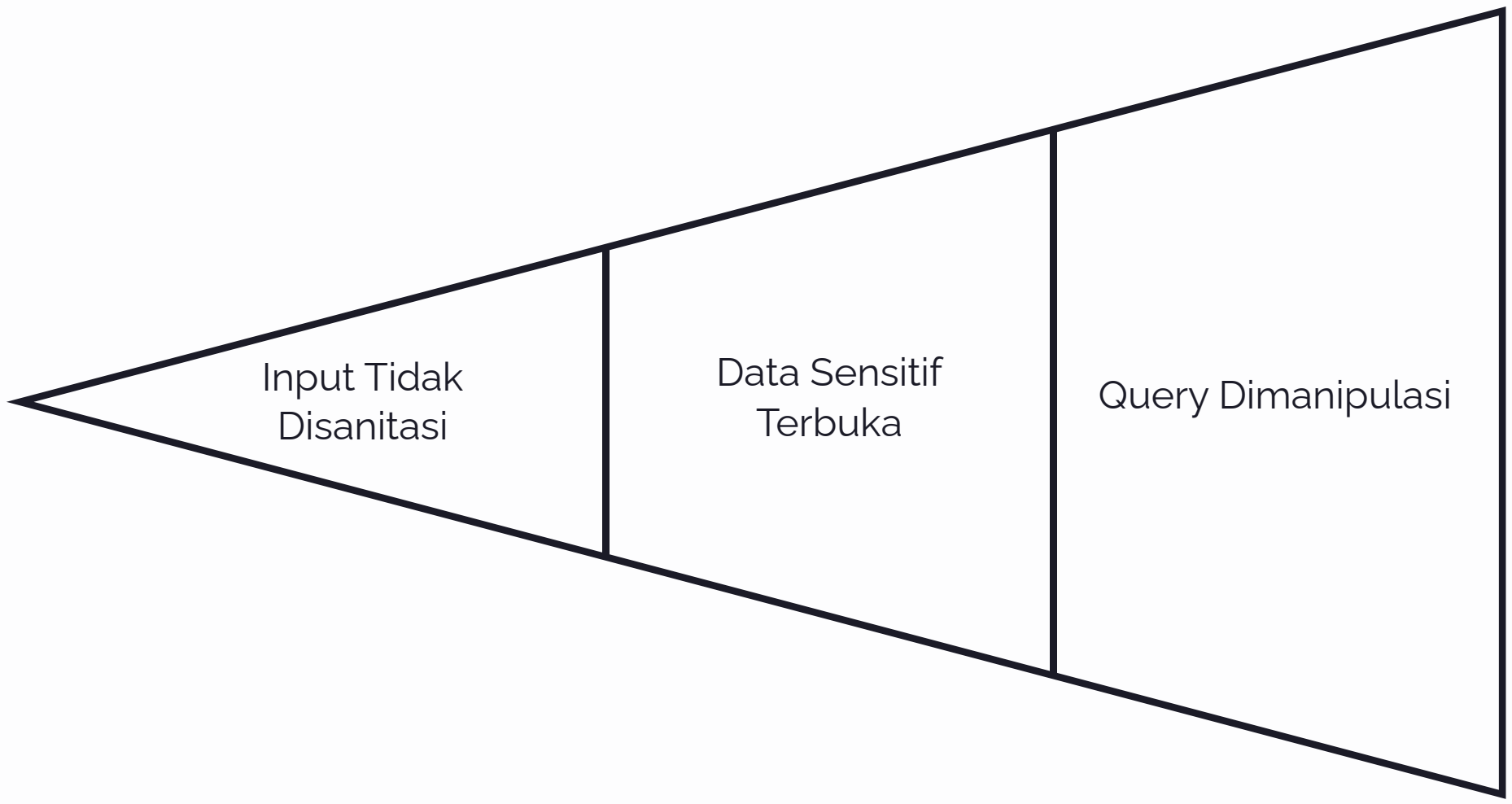
Data dalam Response

- API mengembalikan field yang tidak diperlukan
- Token API bocor dalam response body
- PII pengguna lain dalam response JSON

KERENTANAN #8

Basic SQL Injection Concepts

SQL Injection terjadi ketika input pengguna dimasukkan langsung ke dalam query SQL tanpa sanitasi yang tepat, memungkinkan penyerang memanipulasi query database untuk mengekstrak, memodifikasi, atau menghapus data.



Payload Dasar untuk Testing

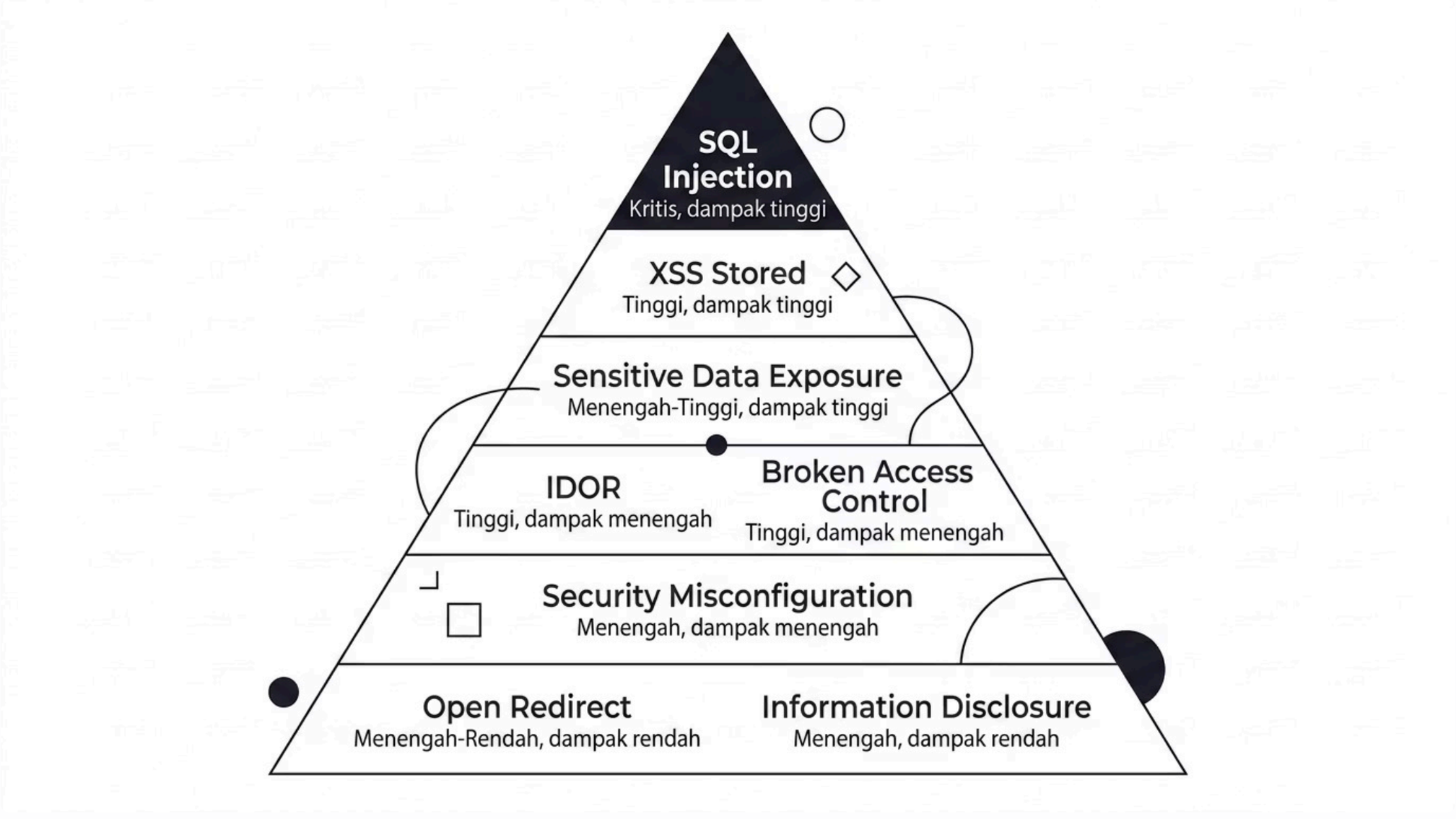
- `'` – trigger syntax error
- `' OR '1'='1` – bypass autentikasi
- `' AND 1=2--` – blind SQL test

Titik Injeksi Umum

- Form login dan pencarian
- Parameter ID di URL
- Filter dan sorting di aplikasi

Perbandingan Tingkat Risiko Kerentanan

Setiap kerentanan memiliki tingkat dampak dan kemudahan eksploitasi yang berbeda. Pemahaman ini membantu memprioritaskan temuan dalam laporan bug bounty.



Semakin tinggi posisi dalam piramida, semakin kritis dampak kerentanan tersebut dan semakin tinggi nilai bounty yang biasanya diberikan.

Praktik: Lab Simulasi Kerentanan

Pemahaman teoritis harus diperkuat dengan latihan langsung. Lab simulasi dirancang untuk mereplikasi kondisi aplikasi web nyata dalam lingkungan yang aman dan legal.

01

Setup Lingkungan Lab

Akses platform lab yang telah disiapkan. Pastikan proxy (Burp Suite/OWASP ZAP) sudah terkonfigurasi dengan benar.

02

Identifikasi Target & Permukaan Serangan

Pelajari aplikasi target, identifikasi semua input, endpoint, dan parameter yang bisa diuji.

03

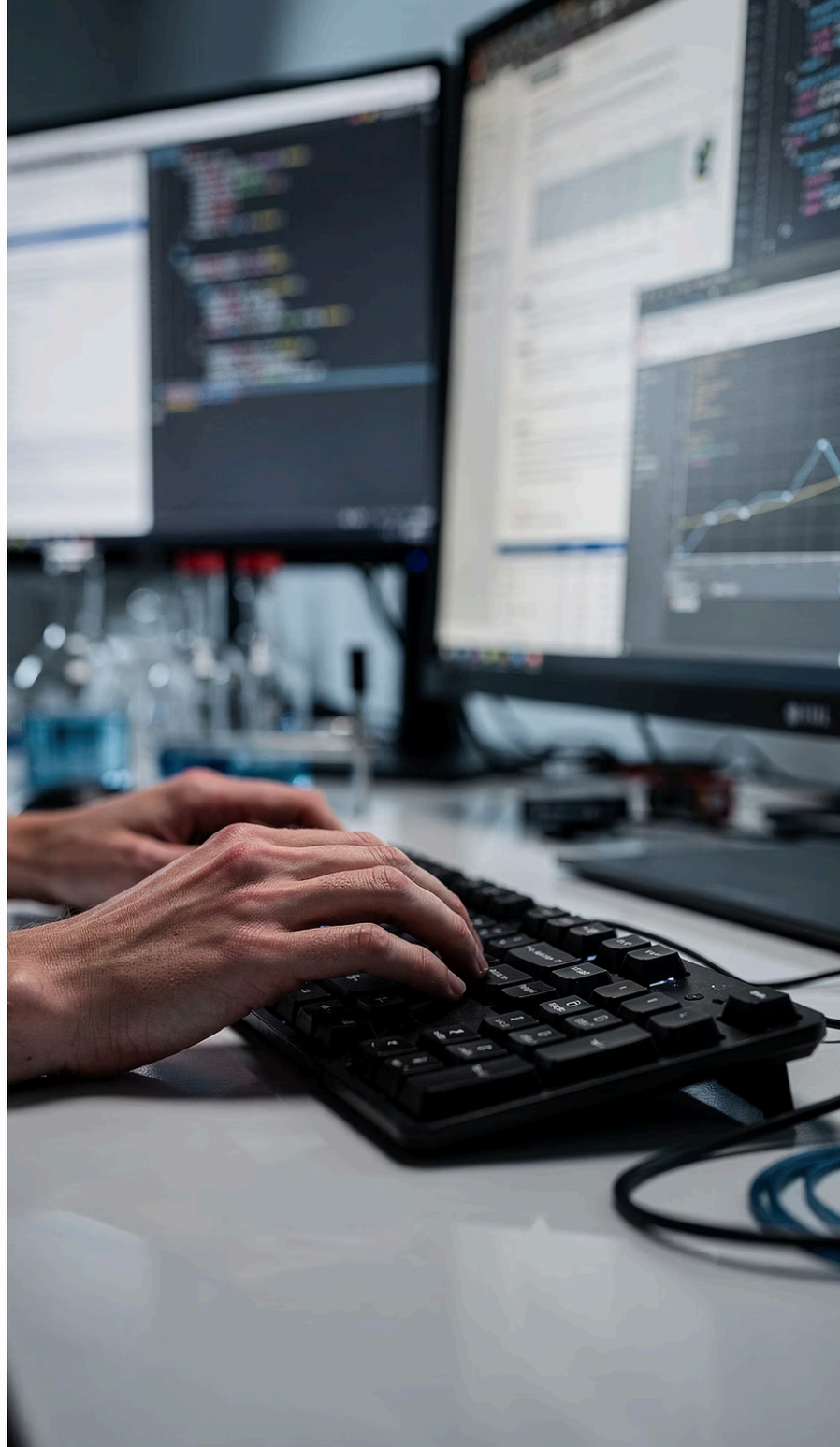
Eksekusi Pengujian Terarah

Uji setiap kerentanan yang dipelajari secara sistematis menggunakan payload yang sesuai.

04

Dokumentasi Temuan

Catat setiap temuan dengan bukti yang jelas: URL, payload, screenshot, dan dampak yang terjadi.



Analisis Bug & Validasi Temuan

Tidak semua temuan adalah kerentanan yang valid. Kemampuan menganalisis dan memvalidasi temuan adalah keterampilan kritis yang membedakan peneliti keamanan yang baik dari yang biasa.

Checklist Validasi

- ✓ Apakah kerentanan dapat direproduksi?
- ✓ Apakah ada dampak nyata terhadap pengguna atau data?
- ✓ Apakah ini dalam scope program?
- ✓ Apakah sudah pernah dilaporkan sebelumnya?
- ✓ Apakah CVSS score dapat dihitung?

Elemen Laporan yang Kuat

- Langkah reproduksi yang jelas dan terurut
- Screenshot atau video sebagai bukti
- Penjelasan dampak potensial secara bisnis
- Saran perbaikan (rekomendasi)
- Klasifikasi tingkat keparahan yang tepat

Laporan yang baik dan terstruktur meningkatkan peluang temuan diterima sebagai valid dan mendapatkan reward yang sesuai.

Outcome & Ringkasan Modul 4

Setelah menyelesaikan Modul 4, peserta diharapkan telah memiliki fondasi yang kuat untuk mengenali dan melaporkan kerentanan umum dalam aplikasi web.

Mengenali Pola

Mampu mengidentifikasi 8 jenis kerentanan umum dalam skenario nyata.

Praktik Lab

Telah berlatih langsung melalui simulasi lab dengan target yang representatif.

Validasi Temuan

Mampu membedakan temuan valid dari false positive dan mendokumentasikannya.

Siap Bug Hunting

Siap mengaplikasikan ilmu pada program bug bounty nyata secara etis dan legal.

Edy Susanto — Founder C-SIX Security