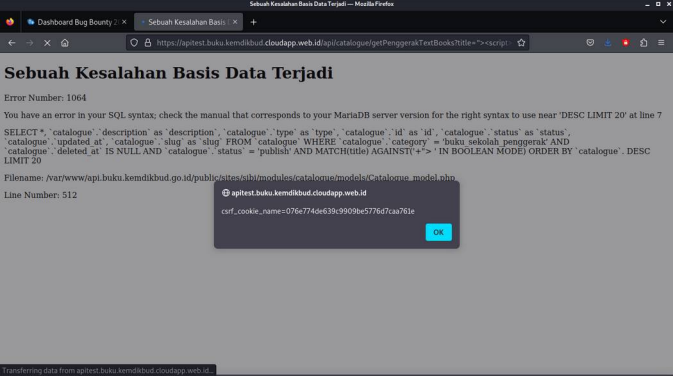


## Laporan Pengujian atau Hasil Pencarian Celah Keamanan oleh Peserta Bug Bounty 2024

No.	Poin (Indikator)	Hasil
1.	Nama Target	SIBI (website API)
2.	Nama Kerentanan	Reflected XSS
3.	Jenis Kerentanan	A03 Injection
4.	Deskripsi Kerentanan	<p>Kerentanan Reflected XSS (Cross-Site Scripting) adalah jenis kerentanan keamanan pada aplikasi web yang memungkinkan penyerang menyisipkan skrip berbahaya ke dalam halaman web yang kemudian dieksekusi oleh pengguna yang melihat halaman tersebut. Skrip ini biasanya disisipkan melalui parameter URL atau input pengguna lainnya yang kemudian direfleksikan kembali ke halaman web tanpa pengolahan atau validasi yang memadai. Ketika pengguna menyentuh halaman yang terinfeksi, skrip yang dimasukkan oleh penyerang akan dijalankan dalam konteks sesi pengguna tersebut, yang dapat mengakibatkan berbagai konsekuensi berbahaya seperti pencurian informasi pribadi, pengalihan sesi, atau manipulasi konten.</p>
5.	Lokasi/URL	<p><a href="https://apitest.buku.kemdikbud.cloudapp.web.id/api/catalogue/getPenggerakTextBooks?title=%22%3E%3Cscript%3Ealert(document.cookie)%3C/script%3E&amp;limit=20&amp;offset=0">https://apitest.buku.kemdikbud.cloudapp.web.id/api/catalogue/getPenggerakTextBooks?title=%22%3E%3Cscript%3Ealert(document.cookie)%3C/script%3E&amp;limit=20&amp;offset=0</a></p>

6.	<i>IP Address Source</i> (IP Address Peserta)	125.166.118.135
7.	Dampak	Kerentanan ini dapat dimanfaatkan oleh penyerang untuk melakukan serangan phishing, merusak reputasi situs web, atau mencuri informasi sensitif seperti kuki sesi, yang dapat memberi akses tanpa izin ke akun pengguna. Untuk mengurangi risiko kerentanan Reflected XSS, praktik pengembangan yang baik meliputi sanitasi dan validasi input, penggunaan header keamanan HTTP seperti Content Security Policy (CSP), serta pendidikan pengguna tentang praktik pengamanan internet yang aman.
8.	Langkah Penetrasi dan Tangkapan Layar (Screenshots) Temuan	<ol style="list-style-type: none"> <li><b>Identifikasi Potensi Kerentanan:</b> Saya mulai dengan mengidentifikasi bahwa ada potensi kerentanan dalam aplikasi web yang saya uji. Saya fokus pada mencari titik-titik input yang mungkin dapat disisipi dengan kode JavaScript untuk mengevaluasi apakah ada celah yang dapat dimanfaatkan.</li> <li><b>Pengamatan Hasil:</b> Saya melaksanakan serangkaian pengujian dengan menyisipkan kode JavaScript ke dalam parameter query title dan memuat ulang halaman web. Melalui pengamatan saya, saya menyadari bahwa kode JavaScript yang saya sisipkan berhasil dieksekusi saat halaman web dimuat ulang, mengkonfirmasi adanya kerentanan Reflected XSS.</li> </ol>

		
9.	Rekomendasi	<ol style="list-style-type: none"> <li>1. <b>Sanitasi Input:</b> Pastikan untuk menyaring dan membersihkan semua input dari pengguna sebelum menggunakannya dalam halaman web. Ini dapat dilakukan dengan menggunakan teknik seperti fungsi escape pada bahasa pemrograman atau menggunakan pustaka sanitasi input yang tersedia.</li> <li>2. <b>Penggunaan Content Security Policy (CSP):</b> Terapkan kebijakan keamanan konten (Content Security Policy) yang ketat di server Anda. Konfigurasi CSP dapat membantu membatasi sumber yang diizinkan untuk memuat atau menjalankan skrip di halaman web, sehingga mengurangi risiko dari serangan XSS.</li> </ol>
10.	Referensi	OWASP TOP TEN