Muhamad Hidayat Follow
Feb 19 · 4 min read

OWASP API Security (Offensive Prespective) : Improper Assets Management #9

IDN Times

Assalamualaikum Wr Wb.

Apa itu Improper Assets Management?

Kesalahan terjadi pada kelalaian management asset dimana assets non-production tersingkap ter-ekspose ke internet public, dan memiliki resiko dapat diakses dengan mudah oleh Threat Actor.

Tidak hanya persoalan Assets production / non production, berikut hal-hal termasuk dalam Improper Assets Management:

- Siapa yang harus memiliki akses jaringan ke API (misalnya publik, internal, mitra)?
- Versi API mana yang sedang berjalan? Data apa yang dikumpulkan dan diolah oleh API (misalnya PII)?
- Bagaimana alur data-nya? Tidak ada dokumentasi, atau dokumentasi yang ada tidak diperbarui.
- Tidak ada rencana pembaruan untuk setiap versi API.
- Inventaris host hilang atau usang
- Inventaris layanan terintegrasi, baik pihak pertama maupun pihak ketiga,

Berikut salah satu contoh penyerangan terkait Improper Assets Management yang umum terjadi pada REST API.

#418823 Reflected XSS on developers.zomato.com (hackerone.com)

Contoh case berikut terdapat sebuah vulnerability Reflected XSS pada https://developers.zomato.com/documentation yang menggunakan Old version Swagger UI.

Dengan membuat sebuah endpoint berisi payload XSS **<script>alert(document.cookie)</script>:**

{"swagger":"2.0","info":{"description":"This is a sample server Petstore server

XSS pun berhasil tereksekusi setelah melakukan input endpoint pada Swagger UI.

Mitigasi terkait permasalahan ini adalah dengan melakukan update versi Swagger UI, dikarenakan di versi tersebut masih memiliki beberapa vulnerability.

Mitigasi

- Inventarisasi semua host API dan dokumentasikan aspek penting dari masing-masing, dengan fokus pada Environment API (misalnya produksi, staging, tes, development), siapa yang harus memiliki akses jaringan ke host (misalnya publik, internal, mitra) dan versi API.
- Inventarisasi layanan terintegrasi dan dokumentasikan aspek penting seperti peran mereka dalam sistem, data apa yang ditukar (alur data), dan sensitivitasnya.
- Dokumentasikan semua aspek dari API Anda seperti otentikasi, kesalahan, pengalihan, pembatasan tingkat permintaan, kebijakan berbagi resource cross-origin (CORS) dan endpoint, termasuk parameter, permintaan, dan responsnya. Buat dokumentasi secara otomatis dengan mengadopsi standar terbuka.
- Sertakan pembuatan dokumentasi dalam pipa CI/CD Anda. Jadikan dokumentasi API tersedia untuk mereka yang berwenang untuk menggunakan API.
- Gunakan tindakan perlindungan eksternal seperti firewall keamanan API untuk semua versi API yang terbuka, bukan hanya untuk versi produksi saat ini. Hindari menggunakan data produksi dengan penyebaran API non-produksi. Jika hal ini tidak dapat dihindari, endpoint ini harus mendapatkan perlakuan keamanan yang sama dengan produksi.
- Ketika versi API yang lebih baru mencakup perbaikan keamanan, lakukan analisis risiko untuk membuat keputusan tindakan mitigasi yang diperlukan

untuk versi yang lebih lama: misalnya, apakah mungkin untuk mengembalikan perbaikan tanpa memecahkan kompatibilitas API atau Anda perlu segera menghapus versi lama dan memaksa semua klien untuk beralih ke versi terbaru.

Api Development Owasp Top 10 Xss Vulnerability Rest Api Hacking