BAB I
PENDAHULUAN

I.1 Latar Belakang

Di era globalisasi saat ini, layanan pertukaran data dan informasi antar aplikasi telah menjadi faktor utama dalam memenuhi kebutuhan pengguna. *Application Programming Interface* (API) adalah antarmuka yang digunakan untuk mengakses aplikasi atau layanan dari sebuah program. API memungkinkan pengembang untuk memakai fungsi yang sudah ada dari aplikasi lain sehingga tidak perlu membuat ulang dari awal. Pada konteks *website*, API merupakan pemanggilan fungsi melalui *Hyper Text Transfer Protocol* (HTTP) dan mendapatkan respon berupa *Extensible Markup Language* (XML) atau *JavaScript Object Notation* (JSON) (Afriansyah et al., 2021). Saat ini, terdapat berbagai jenis API yang digunakan, salah satunya adalah GraphQL. GraphQL adalah bahasa *query* untuk API dan *runtime* untuk memenuhi *query* tersebut dengan data yang ada (Akmal & Dasaprawira, 2022). Di sisi lain, *information disclosure vulnerability* pada sistem perangkat lunak, terutama pada GraphQL API, merupakan masalah yang sering terjadi dan dapat mempengaruhi keamanan aplikasi. Hal ini dapat membuat informasi data *private* bocor seperti data bisnis, kekayaan intelektual, dan informasi pelanggan. Oleh karena itu, penting untuk melindungi informasi *private* dari serangan yang dapat mengungkap vektor serangan tambahan.

Penelitian ini dilakukan karena GraphQL adalah teknologi API yang relatif baru yang sedang berkembang pesat, dan telah diadopsi oleh perusahaan besar, seperti Facebook, GitHub, dan GitLab. Pada tahun 2023, penggunaan GraphQL API mengalami pertumbuhan yang signifikan, dengan sekitar 40% perusahaan teknologi melaporkan peningkatan dalam adopsi GraphQL dibandingkan tahun sebelumnya. Selain itu juga GraphQL menawarkan model permintaan data yang fleksibel, sehingga dengan popularitasnya yang semakin meningkat menunjukkan potensi besar dalam pengembangan aplikasi modern, sekaligus menekankan pentingnya pengembangan praktik terbaik keamanan dan alat untuk melindungi API GraphQL. Penelitian ini berfokus pada metode dan *tools* yang berhasil dan membutuhkan waktu paling efisien untuk mendapatkan *information disclosure*

vulnerability pada eksploitasi terhadap GraphQL. Pada tahap *reconnaissance* penelitian ini menggunakan bantuan Burp Suite untuk menganalisa kerentanan pada GraphQL.Penelitian ini juga menggunakan dua metode eksploitasi dengan masing-masing bantuan *tools*, yaitu *introspection method* dengan bantuan InQL *tool* dan *field suggestion method* dengan bantuan Clairvoyance *tool*. *Introspection method* padaGraphQL memungkinkan untuk memeriksa atau melihat struktur serta *schema* data yang dimilikinya. InQL *tool* yang digunakan dalam *intospection method* membantu dalam mengotomatisasi proses ini dengan memberikan gambaranlengkap *schema* GraphQL termasuk tipe data, dan *query* yang tersedia. Penyerangdapat menggunakan informasi ini untuk mencari celah keamanan atau *endpoint* yang kurang terlindungi. Di sisi lain, *field suggestion method* adalah teknik yang digunakan untuk mendeteksi dan memanfaatkan informasi yang mungkin tidak disadari oleh *developer* sebagai vektor serangan yang potensial. Clairvoyance *tool* digunakan untuk memanfaatkan kelemahan ini dengan mengidentifikasi *field- field* tersembunyi yang dapat diakses oleh penyerang.

I.2 Perumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan permasalahan untuk penelitian ini adalah:

1. Bagaimana mencari dan mengeksploitasi *Information Disclosure Vulnerability* pada GraphQL API?
2. Metode mana yang paling efisien dalam menemukan kerentanan *Information Disclosure* pada GraphQL API dengan keamanan yang rendah dan tinggi?

I.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka tujuan dari penelitian ini adalah:

1. Mengidentifikasi kerentanan dengan bantuan Burp Suite dan mengeksploitasi kerentanan nya dengan *methods* dan *tools Introspection*, InQL, *Field Suggestion*, dan Clairvoyance.
2. Mengidentifikasi dan menyajikan hasil metrik *time* dari setiap serangan *Information Disclosure Vulnerability* pada GraphQL API.

I.4 Batasan Penelitian

Adapun batasan pada penelitian ini adalah sebagai berikut:

1. Penelitian ini hanya berfokus pada GraphQL API versi 3.9.
2. Penelitian ini menggunakan DVGA yang bersifat *open source* dan dapat ditemukan di (DVGA).

I.5 Manfaat Penelitian

Adapun manfaat yang didapatkan dengan adanya penelitian ini adalah sebagai berikut:

1. Secara teoritis
   a. Dapat menambah pengetahuan tentang *Information Disclosure Vulnerability* pada GraphQL API.
   b. Dapat memberikan pemahaman terkait struktur *schema* GraphQL dan teknik-teknik serangan yang dapat dieksploitasi.
2. Secara praktis
   a. Dapat mengetahui bagaimana melakukan serangan secara manual dan menggunakan *tools* terhadap GraphQL API.
   b. Dapat mengidentifikasi dan mengekstrak informasi *private* dari aplikasi yang menggunakan GraphQL.

I.6 Sistematika Penulisan

Penelitian ini diuraikan dengan sistematika penulisan sebagai berikut:

Bab I Pendahuluan

Bab ini berisi uraian mengenai konteks permasalahan, latar belakang permasalahan, perumusan masalah dengan tujuan mencari *information disclosure vulnerability* pada API GraphQL dan membandingkan metode yang paling efisien dari tiap serangan, tujuan penelitian, batasan penelitian, manfaat penelitian, dan sistematika penulisan.

Bab II Tinjauan Pustaka

Bab ini berisi literatur yang relevan dengan permasalahan yang diteliti. Referensi seperti buku, jurnal, dan sumber lainnya digunakan untuk memperoleh dasar ilmu yang diperlukan dalam penelitian ini. Bebarapa metode dan *tools* yang digunakan seperti *Introspection*, *Field Suggestion*, InQL, dan Clairvoyance.

Bab III Metodologi Penelitian

Bab ini berisi langkah-langkah Tugas Akhir secara rinci meliputi pembuatan Model Konseptual serangan *Information Disclosure Vulnerability*, Sistematika Penyelesaian Masalah, Pengumpulan Data, Pengolahan Data, dan Metode Evaluasi.

Bab IV Perancangan dan Skenario Pengujian

Bab ini secara detail menjelaskan proses-proses perancangan dan skenario pengujian seperti Spesifikasi Perangkat Keras, Spesifikasi Perangkat Lunak, Diagram Konektivitas, *Platform* Eksperimen, Skenario Pengujian, Data Eksperimen, dan Perumusan Implementasi dengan *Data Flow Diagram* berdasarkan Eksploitasi.

Bab V Analisis

Bab ini berisi pembahasan hasil pengujian *serangan Information Disclosure Vulnerability* pada GraphQL API yaitu Analisis *Attack*

Tree, Pengukuran *Time* Pada Eksperimen Eksploitasi dan Hasil Analisis *Attack Tree* Berdasarkan Eksploitasi dengan Metrik *Time.*

Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dari penyelesaian masalah serta jawaban terhadap perumusan masalah yang diajukan. Selain itu, terdapat saran dan perbaikan untuk tugas akhir berikutnya.