ABSTRAK

Graph Query Language (GraphQL) adalah bahasa *query* yang menentukan bagaimana klien berinteraksi dengan *Application Programming Interface* (API). Tujuan utama dibuatnya GraphQL adalah untuk mempermudah komunikasi data antara *backend* dan *frontend* yang dimana GraphQL dapat memberikan deskripsi data yang lengkap dan mudah dipahami. Seiring waktu, popularitas GraphQL terus meningkat dikarenakan kemampuannya yang kuat dalam mengelola dan mengambil data API. Seperti dengan teknologi lainnya, GraphQL juga memiliki beberapa titik kelemahan yang dimana dapat menimbulkan kerentanan salah satunya adalah kerentanan injeksi. Penelitian ini dilakukan dengan tujuan untuk menemukan teknik injeksi mana yang paling efektif dengan dua mode keamanan yaitu *hardening* dan sebelum *hardening* dengan melakukan perbandingan teknik injeksi mana yang membutuhkan waktu lebih cepat sehingga kedepannya dapat menemukan solusi keamanan yang paling efektif. Dalam penelitian eksploitasi menggunakan tiga teknik injeksi yang berbeda yaitu *Command Injection*, *Log Injection* dan *Spoof Injection*. Eksploitasi terhadap GraphQL menggunakan teknik injeksi dirumuskan dalam bentuk *diagram attack tree* dengan tujuan untuk mengetahui relasi eksploitasi *attack tree* yang berdasarkan metrik *time*. Dari hasil waktu eksploitasi tiga teknik injeksi tersebut telah didapatkan bahwa *spoof injection* membutuhkan waktu paling singkat dengan mode keamanan sebelum *hardening* dengan total waktu 32,63*s*. Dan teknik injeksi yang membutuhkan waktu paling lama yaitu *command injection* dengan total waktu 60,15*s*. Pada keamanan setelah *hardening*, terjadi perubahan waktu yang dimana *log injection* berada di urutan pertama yang efektif dengan jumlah waktu 38,19*s* dan pada urutan terakhir, teknik injeksi yang membutuhkan waktu yang cukup lama yaitu *command injection* dengan jumlah waktu 53.52*s*. Sehingga dapat disimpulkan bahwa serangan injeksi yang efektif sebelum *hardening* adalah *spoof injection* sedangkan serangan injeksi yang efektif setelah *hardening* adalah *log injection*, dimana dari serangan injeksi tersebut kedepannya dapat ditemukan solusi keamanan yang paling efektif.

Kata kunci – ***attack tree,* eksploitasi, graphql, injeksi, time**