Analisis isu Memory Safety dalam bahasa pemrogramman Rust berdasarkan catatan CVE = Analysis of Memory-Safety issues in the Rust programming language based on CVEs

Paskalis Abhista Bagaskara Yustiyanto, author

Abstrak

Rust adalah bahasa pemrograman yang dirancang untuk mengatasi permasalahan memory safety tanpa mengorbankan performa. Penelitian terdahulu telah membuktikan bahwa Rust memiliki CPU Time yang lebih cepat dibandingkan bahasa C, meskipun penggunaan memorinya sedikit lebih besar. Namun demikian, bukan berarti Rust hadir tanpa cacat, masih banyak catatan CVE yang dipublikasikan atas nama bahasa pemrograman ini tanpa terkecuali mengenai permasalahan memory safety. Pada tugas akhir ini, penulis melakukan penyelidikan terhadap catatan-catatan CVE mengenai Rust dengan tujuan untuk memahami karakteristik dari isu memory safety pada bahasa pemrograman Rust. Penelitian ini menemukan bahwa Uninitialized Memory Access menjadi kategori may- oritas diantara tiga kategori lainnya dalam dataset yang digunakan. Temuan ini berbeda dengan hasil yang diperoleh penelitian terhadulu, yang telah melakukan penyelidikan serupa mengenai hal ini. Selain itu, penelitian ini juga menemukan bahwa mayoritas kode proof-of-concept yang disediakan dalam catatan CVE mampu untuk memvalidasi dan mereproduksi permasalahan yang dimaksud. Dengan dibuatnya penelitian ini, diharapkan supaya pembaca dapat memahami kode Rust yang berpotensi menyebabkan isu memory safety sehingga dapat membuat kode Rust yang lebih aman.

.....Rust is a programming language design to prevent memory-safety issues without sacri cing performance. Previous research has proven that Rust has a faster CPU Time compared to the C language, although it’s memory usage is slightly larger. However, this does not mean that Rust is without aws, as there are still many CVE records published in the name of this programming language, including those regarding memory safety issues. In this nal project, the author investigates CVE records related to Rust with the aim of understanding the characteristics of memory safety issues in the Rust programming language. This research found that Uninitialized Memory Access is the majority category among three other categories in the dataset used. This result is differs from the previous research, which conducted similar investigations on this matter. Additionaly, this research also found that the majority of proof-of-concept codes provided in the CVE records are capable to validating and reproducing the mentioned issue. With this work, it is hoped that readers can understand Rust code that has the potential to cause memory safety issues, so that enabling them to write safer Rust code.