Oleg Bobrov

LinkedIn | oleg.bobrov.m@gmail.com | t.me/olezhkabobrov | olezhabobrov

Education

Constructor University **Sept. 2022 – Aug. 2023**
Bachelor of Computer Science *Bremen, Germany*

Relevant Coursework:
ML, Kotlin, Database internals, Parallel programming, Mathematical Statistics

National Research University Higher School of Economics **Sept. 2020 – June 2022**
Bachelor of Applied Mathematics and Computer Science *St. Petersburg, Russia*

Relevant Coursework:
C++, Java, Algorithms and Data Structures, Calculus, Linear Algebra, Functional Programming

Experience

Exasol | *C++, Compilers, Databases, SQL, Yacc* **Dec. 2023 – Present**
Software Engineer *Nuremberg, Germany*

- Contributed to the development of the Compiler and Engine components of the Exasol relational analytics database, focusing on query graph optimizations
- Implemented an algorithm to detect and materialize identical subqueries, improving performance by up to 65% on some TPC-DS benchmark queries
- Developed and optimized query execution strategies for complex joins with unions
- Participated in the implementation and testing of a new data type, updated embedded libraries, and performed normal maintenance work

JetBrains | *Kotlin, Kotlin Compiler, Fuzzing, Vert.x* **Nov. 2022 – Aug. 2023**
Research intern *Munich, Germany*

- Redesigned and reimplemented the existing **Kotlin compiler fuzzer** as a loosely-coupled system to support easier extension, experiments and evaluation
- Added support for the Kotlin/Native compiler and implemented klib-specific transformations to discover problems with ABI of evolving libraries
- Revealed and reported over **15 new vulnerabilities** using Kotlin fuzzer

Huawei | *C++, KLEE, gRPC* **Feb. 2022 – June 2022**
Research intern *St. Petersburg, Russia*

- Developed a **static code analyzer** for C/C++ based on KLEE (symbolic execution engine)
- Result represented in SARIF format, showing a full stack trace of a possible error with the initial values, which would cause this
- On tested projects showed **10 times more** mistakes than PVS-studio and **8 times more** mistakes more than CppCheck, however it's more time-consuming

Pet Projects

Implementation of the board game Quoridor with a bot | *C++, Qt*

Cooperative party game Code-team | *C++, Qt, winsock*

Technical Skills

Languages: C++, C, Kotlin, Java, Python
Technologies/Frameworks: Vert.x, Qt, SQL, KLEE, Git, CMake