The Parfait Static Code Analysis Framework
Lessons Learnt

Cristina Cifuentes, Nathan Keynes, John Gough, Diane Corney, Manuel Valdiviezo
Oracle Labs

Extended Abstract

The Parfait static code analyser was conceived at Sun Labs, now Oracle Labs, in 2007. At the time, the project focused on the detection of defects in C/C++ code. Over the next five years, Parfait matured to include detection of vulnerabilities (not just defects) in C/C++ and Java™ while meeting the performance and precision standards expected of a commercial tool: Parfait can analyse 39 of the most common defects in the C language over an operating system codebase of 11 million lines of C code in 1.5 hours with a false positive rate of 10%. Today, Parfait is maintained by Oracle as an internal product and is used by thousands of developers at Oracle worldwide.

Key to Parfait's goals of scaling up to millions of lines of code as well as obtaining high precision of analysis is it's design [CS08] – a waterfall, layered program analysis approach that makes use of demand-driven analyses. Different analyses are used based on the type of defect or vulnerability that is being analysed; at the core is a data flow framework, a symbolic analysis, taint analysis and escape analysis. The demand-driven analyses have provided for efficiencies in scalability.

Parfait is implemented in C++ on top of the LLVM [LLV] framework. The parser is a modified version of Clang [Cla]. Parfait makes use of the LLVM intermediate bitcode representation as the intermediate language used for analysis. Fully linked-in files are analysed, except for cases where the LLVM linker cannot link the files. LLVM bitcode modules, whether linked in or not, are analysed by Parfait. Inter-module analysis support was put in place to analyse large codebases.

To support the Java language, LLVM's intermediate representation was extended with metadata. We describe design considerations taken into account to develop Jaffa, our Java to LLVM IR translator, which was driven by the need to make the IR amenable to analysis (not compilation). Java language support allows Parfait to find access control vulnerabilities in the Java platform [CKG+14], as well as web vulnerabilities [OWA] vulnerabilities in Java SE and Java EE applications, without compromising on scalability and precision. Analysis of 52 defect types (including JNI-specific ones) of the 4.7 million lines of Java code and 1.5 million lines of C++ code in the JDK™ is done in 3.3 hours (70% of the time is spent on Java code), and analysis of 4 OWASP defect types of a commercial cloud JEE application of 1.1 million lines of Java code is done in 5 mins.

This talk summarises lessons learnt in the design and implementation of the Parfait tool, expanding on initial lessons learnt from transferring the project into a production environment [CKL+12]. Insights into what works well in the infrastructure today and what does not will be summarised.

Speaker Biography

Cristina Cifuentes is the Director of Oracle Labs Australia and an Architect at Oracle. Headquartered in Brisbane, the focus of the Lab is Program Analysis as it applies to finding vulnerabilities in software and enhancing the productivity of developers worldwide.

Prior to founding Oracle Labs Australia, Cristina was the Principal Investigator of the Parfait static code analysis project at Sun Microsystems, then Oracle. Today, Oracle Parfait has become the defacto tool used by thousands of Oracle developers for defect and vulnerability detection in real-world, commercially sized C/C++/Java applications. The success of the Parfait tool is founded on the pioneering work in advancing static program analysis techniques carried out by Cristinas team of Researchers and Engineers at Oracle Labs Australia.

Cristina's passion for tackling the big issues in the field of Program Analysis began with her doctoral work in binary decompilation at Queenslands University of Technology, followed by later work on binary translation in the UQBT and Walkabout projects. Prior to her work at Oracle and Sun Microsystems, Cristina held teaching posts at major Australian Universities, co-edited Going Digital, a landmark book on cybersecurity, and served on the executive committees of ACM SIGPLAN and IEEE Reverse Engineering.

Cristina continues to play an active role in the international programming language, compiler construction and software security communities. On the weekends, she channels her interests into mentoring young kids through the CoderDojo network.

References

[CKG+14] Cristina Cifuentes, Nathan Keynes, John Gough, Diane Corney, Lin Gao, and Manuel Valdiviezo. Translating Java into LLVM IR to detect security vulnerabilities. LLVM Developer's Meeting, October 2014. Poster.

[CKL+12] Cristina Cifuentes, Nathan Keynes, Lian Li, Nathan Hawes, and Manuel Valdiviezo. Transitioning Parfait into a development tool. *IEEE Security and Privacy*, 10(3):16–23, May/June 2012.

[Cla] Clang. A C language family frontend for LLVM. `http://clang.llvm.org`.

[CS08] Cristina Cifuentes and Bernhard Scholz. Parfait – designing a scalable bug checker. In *Proceedings of the ACM SIGPLAN Static Analysis Workshop*, pages 4–11, 12 June 2008.

[LLV] LLVM. The LLVM Compiler Infrastructure. `http://www.llvm.org`.

[OWA] OWASP. OWASP Top Ten Project. `https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project`.