

Bibliography in L^AT_EX

Tux et al.

February 16, 2006

1 Introduction

Java programs are compiled into an intermediate language which is interpreted by the Java Virtual Machine (JVM) [5]. The intermediate language, which, following other authors, we call JVMML, is a language of bytecode instructions.

Since JVMML programs can be loaded from the network, security problems may arise. For this reason, JVM embodies a bytecode verifier which performs a set of checks on JVMML programs before their execution. The aim of these checks is to prevent the execution of malicious or wrong code which could corrupt the integrity of the host. The verifier performs a data-flow analysis applied to a type-level abstract interpretation of the JVM.

Given the importance of the bytecode verifier, a lot of research efforts have been dedicated both to its formalization and to study extensions able to accept larger classes of correct programs than the standard verifier does [2, 3, 4].

Finally [1] is very funny.

References

- [1] Papérino, P. and De Paperoni, P. Rich Forever. *The Walt Disney Journal* 1(3), 300–350, 2005.
- [2] Freund, S. N. and Mitchell, J. C. A Formal Framework for the Java Bytecode Language and Verifier. ACM Conference on Object-Oriented Programming Systems, Languages & Applications, 1999.
- [3] Hagiya, M. and Tozawa, A. On a New Method for Dataflow Analysis of Java Virtual Machine Subroutines. Static Analysis Symposium '98, LNCS 1503, Springer, 1998.
- [4] Leroy, X. Java bytecode verification: an overview. In Proceedings of CAV'01, Springer LNCS 2102, 2001.
- [5] Lindholm, T. and Yellin, F. The Java Virtual Machine Specification. The Java Series, Addison-Wesley, 1999.