Implementing a security-typed variant of Scala

Master Thesis

to optain the academic degree
master degree in computer science

Friedrich-Schiller-Universität Jena
Faculty of mathematics and computer science

submitted by Matthias Günther
born 02.09.1985 in Dessau

person in charge: Prof. Dr. Wilhelm R. Rossak
advisor: Assistent Professor Jeffery von Ronne
PD Dr. habil. Wolfram Amme
Andreas Gampe

Jena, May 22, 2010

Abstract

This October sees the release of our first Time of Legends audio drama: Aenarion. Aenarion's story is one of the cornerstones of high elf history, so we've placed it in the capable hands of Black Library stalwart, Gav Thorpe. As with the ToL novel covers we've managed to rope in Jon 'sleep is for wimps' Sullivan. We reminded him that CD covers are actually quite small, but it didn't stop him layering an insane amount of detail onto Aenarion's armour. It seems a crime that we're going to have to show it so small on the cover, so here's the whole piece in all its unadulterated glory.

Was soll ich bloÃŸ noch schreiben. Ich habe echt keine Ahnung was ich dir noch alles so sagen soll.

Nun kann ich eventuell noch einen weiteren Satz schreiben.

Contents

1 Preface

Once upon a time, there was the preface in which Matze starts his last academic works.

2 Scala

Scala is a great language which has many components ...

2.1 Scala in Action

Figure 2.1: *Reihungszugriff fÃ¼r Array A*

References

[Alt06] Philippe Altherr. *A Typed Intermediate Language and Algorithms for Compiling Scala by Successive Rewritings*. PhD thesis, EPFL, March 2006.

[AR80a] Gregory R. Andrews and Richard P. Reitman. An axiomatic approach to information flow in programms. In *ACM Transactions on Programming Languages and Systems*, pages 56–76. ACM, 1980.

[AR80b] Gregory R. Andrews and Richard P. Reitman. An axiomatic approach to information flow in programs. In *ACM Transaction on Programming Languages and Systems*, pages 56–76. ACM, 1980.

[Bib77] K. J. Biba. Integrity considerations for secure computer systems. Technical report, USAF Electronic System Division, April 1977.

[CC04] Hubie Chen and Stephen Chong. Owned policies for information security. In *Proceedings of the 17th IEEE Computer Security Workshop*. IEEE, June 2004.

[CLR01] Thomas A. Cormen, Charles E. Leierson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press, 2 edition, 2001.

[CMS05] Stephen Clarkson, Andrew C. Myers, and Fred B. Schneider. Bleif in information flow. In *Proceedings of the 18th IEEE Computer Security Foundations Workshop*, pages 31–45. IEEE, June 2005.

[Fol91] Simon N. Foley. A taxonomy for information flow policies and models. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 98–108. IEEE, 1991.

[FSBJ97] Elena Ferarri, Pierangela Samarati, Elisa Bertino, and Sushil Jajodia. Providing flexibility in information flow control for object-oriented systems. In *Proc. IEEE Symposium on Security and Privacy*, pages 130–140. IEEE, May 1997.

[HDT87] Susan Horwitz, Alan Demers, and Teitelbaum Tim. An efficient general iterative algorithm for dataflow analysis. *Acta Informatica*, 24:679–694, 1987.

[JG91] Pierre Jouvelot and David K. Gifford. Algebraic reconstruction of types and effects. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 303–310. IEEE, Januar 1991.

[JL75] A. K. Jones and R.J. Lipton. The enforcement of security policies for computation. In *Proc. 5th ACM Symp. on operating System Principles, ACM Operating System Review*, pages 197–206. ACM, November 1975.

[Kri10] Jan Kriesten. *Praxisbuch Scala: Programmieren in Scala für Ein- und Umsteiger*. Cambridge University Press, 1 edition, 2010.

[KW94] Atsushi Kanamori and Daniel Weise. Worklist management strategies for dataflow analysis. Technical report, Microsoft Research, May 1994.

[LABW91] Butler Lampson, Martin Abadi, Michale Burrows, and Edward Wobber. Authentication in distributed systems: Theory and practice. In *Proc. 13th ACM Symp. on Operation System Principles*, pages 165–182. ACM, October 1991.

[LS10] Christos K.K. Loverdos and Apostolos Syropoulos. *Steps in Scala*. Cambridge University Press, 1 edition, 2010.

[LY96] T. Lindholm and F. Yellin. *The Java Virtual Machine*. Addison-Wesley, 1 edition, 1996.

[McL88] McLean. Reasining about security models. In *Proc. IEEE Symposium on Security and Privacy*, pages 123–131. IEEE, 1988.

[McL90] McLean. Security models and information flow. In *Proc. IEEE Symposium on Security and Privacy*, pages 180–187. IEEE, 1990.

[Mil81] Jonathan K. Millen. Information flow analysis for formal specifications. In *Proc. IEEE Symposium on Security and Privacy*, pages 3–8. IEEE, April 1981.

[ML97] Andrew C. Myers and Barbara Liskov. A decentralized model for information flow control. In *Proc. 17th ACM Symposium on Operating System Principles*, pages 129–142. ACM, May 1997.

[ML98] Andrew C. Myers and Barbara Liskov. Complete, safe information flow with decentralized labels. In *Proc. 17th ACM Symposium on Security and Privacy*. ACM, May 1998.

[MPO07] Adriaan Moors, Frank Piessens, and Martin Odersky. Towards equal rights for higher-kinded types. In *6th International Workshop on Multiparadigm Programming with Languages at the European Conference on Object-Oriented Programming*, 2007.

[MPO08] Adriaan Moors, Frank Piessens, and Martin Odersky. Safe type-level abstraction in scala. Technical report, K.U. Leuven and EPFL, January 2008.

[MR79] E. Morel and C. Renvoise. Global optimization by suppression of partial redundancies. *Commun. ACM*, 22:96–103, 1979.

[Mye99a] Andrew C. Myers. *Mostly-Static Decentralized Information Flow Control*. PhD thesis, Massachusetts Institute of Technology, January 1999.

[Mye99b] Andrew C. Myers. Practical mostly-static information flow control. In *Proc. 26th ACM Symposium on Principles of Programming Languages*. ACM, 1999.

[Nec92] George C. Necula. Proof-carrying code. In *Proceedings of the ACM Symp. on Principles of Programming Languages*, pages 106–119. ACM, August 1992.

[OAC$^+$06] Martin Odersky, Philippe Altherr, Vincent Cremet, Dragos Gilles Dubochet, Burak Emir, McDirmid Sean, Stephane Micheloud, Nikolay Mihaylov, Michel Schinz, Erik Stenman, Lex Spoon, and Matthias Zenger. An overview of the scala programming language. Technical report, EPFL, January 2006.

[Ode09a] Martin Odersky. Scala by example. available form http://scala.epfl.ch, March 2009.

[Ode09b] Martin Odersky. The scala language specification, March 2009.

[Pol09] D. Pollak. *Beginning Scala*. Apress, 1 edition, 2009.

[RSC92] Joel Richardson, Peter Schwarz, and Luis-Felip Cabrera. Cacl: Efficient fine-grained protection for objects. In *Proceedings of the 1992 ACM Conference on Object-Oriented Programming Systems, Languages, and Applications*, pages 154–165. ACM, October 1992.

[SM03] Andrei Sabelfeld and Andrew C. Myers. Language-based information-flow security. In *IEEE Journal on Selected Areas in Communication*. IEEE, January 2003.

[SP09] Michel Schinz and Hallerm Philipp. A scala tutorial for java programmers. available form http://scala.epfl.ch, March 2009.

[SS98] Pierangela Samarati and Latanya Sweeny. Generalizing data to provide anonymity when disclosing information. In *IEEE Symposium on Security and Privacy*, pages 0–18. IEEE, 1998.

[Sto81] Allen Stoughten. Access flow: A protection model which integrates access control and information flow. In *IEEE Symposium on Security and Privacy*, pages 9–18. IEEE, 1981.

[Sub09] Venkat Subramaniam. *Programming Scala: Tackle Multi-Core Complexity on the Java Virtual Machine*. The Pragmatic Programmers, 1 edition, 2009.

[SV98] Geoffrey Smith and Dennis Volpano. Secure information flow in a multi-threaded imperative language. In *Proc. 25th ACM symp. on Principles of Programming Languages*. ACM, January 1998.

[TW89] Phil Terry and Simon Wiseman. A new security model. In *Proc. IEEE Symposium Security and Privacy*, pages 215–228. IEEE, 1989.

[VIS96] Dennis Volpano, Cynthia Irvine, and Geoffrey Smith. A sound type system for secure flow analysis. *J. Comput. Secur.*, 4:167–187, 1996.

[WBF97] Dan S. Wallach, Dirk Balfanz, and Edward W. Felten. Extensible security architecture for java. In *Proc. 16 th ACM Symp. on Operation System Principles*, pages 116–128. ACM, January 1997.

[WP09] Dean Wampler and Alex Payne. *Programming Scala: Scalability = Functional Programming + Objects*. O'Reilly Media, 1 edition, 2009.

[ZM07] Lantian Zheng and Andrew C. Myers. Dynamic security labels and static information flow. In *International Journal of Information Security*, pages 2–3. Springer, March 2007.

[ZZNM01] Steve Zdancewic, Lantian Zheng, Nathaniel Nytsrim, and Andrew C. Myers. Secure programm partitioning. In *ACM Transactions on Computing Systems*, pages 283–328. ACM, October 2001.

List of Tables

List of Figures

3 Appendix

A Hey

aa

Listing 1: *Bala*

```
44 match {
  case 44 => true // if we match 44, the result is true
  case _ => false // otherwise the result is false
}


// pattern-Match fuer Klassen
Stuff("David", 45) match {
  case Stuff("David", 45) => true
  case _ => false
}

// koennen den Namen testen, wobei uns der zweite Parameter (age) rille ist
Stuff("David", 45) match {
  case Stuff("David", _) => "David"
  case _ => "Other"
}

// koennen das age field extrahieren und in die howold-Variable schreiben
Stuff("David", 45) match {
  case Stuff("David", howOld) => "David, age: "+ howOld
  case _ => "Other"
}

// koennen einen Guard setzen
Stuff("David", 45) match {
  case Stuff("David", age) if age < 30 => "young David"
  case Stuff("David", _) => "old David"
  case _ => "Other"
}
```

aa

Figure 3.1: *array*

B You

Punkt 1,		2 und	3
Längerer wird brochen	Text um-	und	nicht über den Rest geschrieben

Table 3.1: *bla*

Figure 3.2: *Reihungszugriff für Array A*

Independance declaration

I declare that I have made this work independently, and olny using the specified sources and utilities.

Jena, May 22, 2010