Contract-Based Reasoning for Verification and Certification of Secure Information Flow Policies in Industrial Workflows*

John Hatcliff

SAnToS Laboratory
Kansas State University
Manhattan, KS 66506, USA
hatcliff@cis.ksu.edu
http://www.cis.ksu.edu/santos

Abstract. Successful transfer of formal engineering methods from academia to industrial development depends on a variety of factors: a proper understanding of the industrial development context, effective and usable technology that can be integrated with development workflows to provide a compelling solution to serious development challenges, "buy-in" from industrial developers and management, an appropriate business model for supporting the deployed technology, plus a lot of luck. I describe how many of these factors are manifesting themselves in an effort by our research group to transition rigorous static analyses and novel Hoare-style logics into a large industrial development process for information assurance and security applications.

The applications that we are targeting address the following problem: international infrastructure and defense forces are increasingly relying on complex systems that share information with multiple levels of security (MLS). In such systems, there is a strong tension between providing aggressive information flow to gain operational and strategic advantage while preventing leakage to unauthorized parties. In this context, it is exceedingly difficult to specify and certify security policies, and produce *evidence* that a system provides end-to-end trust.

In the past, verification and certification obligations in this domain have been met by using heavy-weight theorem proving technology that requires many manual steps or by light-weight contract-based static analyses that are too imprecise for specifying and verifying crucial information flow properties. In this talk, I will explain how our research team is (a) building integrated tool support for automatically discovering and visualizing information flows through programs and architectures, and (b) providing code-integrated software contracts for specifying information flow policies, and (c) applying synergistic blends of static analyses and automated reasoning based on weakest-precondition calculi to aid developers in automatically discharging verification obligations. These techniques aim to hit a "sweet spot" that provides greater automation and developer integration

* This work was supported in part by the US National Science Foundation (NSF) awards 0454348 and CAREER award 0644288, the US Air Force Office of Scientific Research (AFOSR), and Rockwell Collins.

S. Liu, T. Maibaum, and K. Araki (Eds.): ICFEM 2008, LNCS 5256, pp. 3–4, 2008.

than previous theorem-proving-based approaches while offering increased precision over previous static-analysis-based frameworks. Throughout the presentation, I will assess approaches/strategies that have been successful in moving our research results into industrial practice and summarize challenges that remain.

Acknowledgments

This talk is based on joint work with researchers from Kansas State including Torben Amtoft, Robby, Edwin Rodríguez, Jonathan Hoag, Todd Wallentine and Loai Zomlot, and with David Greve from Rockwell Collins, Advanced Technology Center.

References

1. Amtoft, T., Bandhakavi, S., Banerjee, A.: A logic for information flow in object-oriented programs. In: 33rd Principles of Programming Languages (POPL), pp. 91–102 (2006)
2. Amtoft, T., Banerjee, A.: A logic for information flow analysis with an application to forward slicing of simple imperative programs. Science of Comp. Prog. 64(1), 3–28 (2007)
3. Amtoft, T., Banerjee, A.: Verification condition generation for conditional information flow. In: 5th ACM Workshop on Formal Methods in Security Engineering (FMSE), pp. 2–11 (2007); A long version, with proofs, appears as technical report KSU CIS TR 2007-2
4. Amtoft, T., Hatcliff, J., Rodríguez, E., Robby,, Hoag, J., Greve, D.: Specification and checking of software contracts for conditional information flow. In: Cuellar, J., Maibaum, T.S.E. (eds.) FM 2008. LNCS, vol. 5014. Springer, Heidelberg (2008)
5. Chapman, R., Hilton, A.: Enforcing security and safety models with an information flow analysis tool. In: SIGAda 2004, Atlanta, Georgia, pp. 39–46. ACM Press, New York (2004)
6. Greve, D., Wilding, M., Vanfleet, W.M.: A separation kernel formal security policy. In: 4th International Workshop on the ACL2 Prover and its Applications (ACL2 2003) (2003)
7. Heitmeyer, C.L., Archer, M., Leonard, E.I., McLean, J.: Formal specification and verification of data separation in a separation kernel for an embedded system. In: 13th ACM Conference on Computer and Communications Security (CCS 2006), pp. 346–355 (2006)
8. Ranganath, V.P., Amtoft, T., Banerjee, A., Hatcliff, J., Dwyer, M.B.: A new foundation for control dependence and slicing for modern program structures. TOPLAS 29(5) (August 2007); In: Sagiv, M. (ed.) ESOP 2005. LNCS, vol. 3444. Springer, Heidelberg (2005)
9. Ranganath, V.P., Hatcliff, J.: Slicing concurrent Java programs using Indus and Kaveri. International Journal on Software Tools for Technology Transfer (STTT) 9(5-6), 489–504 (2007); Special section FASE 2004/2005.
10. Rushby, J.: The design and verification of secure systems. In: 8th ACM Symposium on Operating Systems Principles, vol. 15(5), pp. 12–21 (1981)
11. Rushby, J., DeLong, R.: Compositional security evaluation: The MILS approach, `http://www.csl.sri.com/~rushby/slides/iccc07.pdf`