

Curriculum Vitae

Michael William Hicks

1 Personal Information

Current Position

Professor, University of Maryland
Department of Computer Science (50%)
University of Maryland Institute of Advanced Computer Studies (50%)
Phone: +1 301 405 2710
E-mail: mwh@cs.umd.edu
WWW: <http://www.cs.umd.edu/~mwh/>
Twitter: @michael_w_hicks
Blog: <http://www.pl-enthusiast.net>

Education

University of Pennsylvania, Philadelphia, Pennsylvania. Ph.D. in Computer and Information Science, 2001 (*won ACM SIGPLAN Doctoral Dissertation Award, 2002*); M.Sc. in Computer and Information Science, 1996.

Pennsylvania State University, State College, Pennsylvania. B.Sc. in Computer Science with Honors; *Magna Cum Laude* and Department *Standard Bearer*, 1993.

Employment

Professor, *Department of Computer Science and University of Maryland Institute of Advanced Computer Studies (UMIACS)*, University of Maryland, College Park, Maryland. *July 2013–present*.

Director, *Maryland Cybersecurity Center*, University of Maryland, College Park, Maryland. *November 2011–2013*.

Associate Professor, *Department of Computer Science and University of Maryland Institute of Advanced Computer Studies (UMIACS)*, University of Maryland, College Park, Maryland. *July 2008–2013*.

Visiting Professor, *École Normale Supérieure*, Paris, France. *October 2012*.

Affiliate Associate Professor, *Department of Electrical and Computer Engineering*, University of Maryland, College Park, Maryland. *July 2008–2011*.

Adjunct, *Institute for Defense Analyses (IDA) Center for Computing Sciences (CCS)*, Bowie, Maryland. *October 2006–present*.

Visiting Scholar, *The Computer Laboratory*, University of Cambridge, England. *September 2008–August 2009*.

Visiting Researcher, *Microsoft Research Cambridge*, England. *September 2008–November 2008*.

Assistant Professor, *Department of Computer Science and University of Maryland Institute of Advanced Computer Studies (UMIACS)*, University of Maryland, College Park, Maryland. *July 2002–July 2008*.

Affiliate Assistant Professor, *Department of Electrical and Computer Engineering, University of Maryland*, College Park, Maryland. July 2005–2007.

Post-doctoral Research Associate, *Department of Computer Science, Cornell University*, Ithaca, New York. July 2001–July 2002.

Lecturer, *Department of Computer and Information Science, University of Pennsylvania*, Philadelphia, Pennsylvania. August 1999–December 1999.

Scientist, *NEC Research Institute*, Princeton, New Jersey. May 1998–August 1999 (part time).

Software Engineer, *ARINC, Inc.*, Annapolis, Maryland. June 1993–April 1997.

2 Research, Scholarly, and Creative Activities

[†]Indicates a student or post-doc advised or co-advised by Dr. Hicks.

[‡]Indicates work performed in part while student advised or co-advised by Dr. Hicks.

*Indicates a student advised or co-advised by Dr. Hicks for this project.

b. Articles in Refereed Journals

1. Christopher M. Hayden[‡], Karla Saur[†], Edward K. Smith[†], Michael Hicks, and Jeffrey S. Foster. Efficient, general-purpose dynamic software updating for C. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 36(4):13, October 2014.
2. Piotr Mardziel[†], Stephen Magill[†], Michael Hicks, and Mudhakar Srivatsa. Dynamic enforcement of knowledge-based security policies using probabilistic abstract interpretation. *Journal of Computer Security*, 21:463–532, October 2013.
3. Christopher M. Hayden[†], Edward K. Smith[†], Eric A. Hardisty*, Michael Hicks, and Jeffrey S. Foster. Evaluating dynamic software update safety using efficient systematic testing. *IEEE Transactions on Software Engineering*, 38(6):1340–1354, December 2012.
4. Polyvios Pratikakis[‡], Jeffrey S. Foster, and Michael Hicks. Locksmith: Practical static race detection for C. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 33(1):Article 3, January 2011.
5. Michael Hicks and Jeffrey S. Foster. SCORE: Agile research group management. *Communications of the ACM*, 53(10):30–31, October 2010.
6. Jeffrey A. Meister*, Jeffrey S. Foster, and Michael Hicks. Serializing C intermediate representations for efficient and portable parsing. *Software, Practice, and Experience*, 40(3):225–238, February 2010.
7. Peter Sewell, Gareth Stoye*, Michael Hicks, Gavin Bierman, and Keith Wansbrough. Dynamic re-binding for marshalling and update, via redex-time and destruct-time reduction. *Journal of Functional Programming (JFP)*, 18(4):437–502, July 2008.
8. Saurabh Srivastava[†], Michael Hicks, Jeffrey S. Foster, and Patrick Jenkins*. Modular information hiding and type safe linking for C. *IEEE Transactions on Software Engineering*, 34(3):1–20, May 2008.
9. Gareth Stoye*, Michael Hicks, Gavin Bierman, Peter Sewell, and Iulian Neamtii[†]. *Mutatis Mutandis*: Safe and flexible dynamic software updating (full version). *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 29(4), July 2007.

10. Nikhil Swamy[†], Michael Hicks, Greg Morrisett, Dan Grossman, and Trevor Jim. Safe manual memory management in Cyclone. *Science of Computer Programming (SCP)*, 62(2):122–144, October 2006. Special issue on memory management.
11. Michael Hicks and Scott M. Nettles. Dynamic software updating. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 27(6):1049–1096, November 2005.
12. James Rose[†], Nikhil Swamy[†], and Michael Hicks. Dynamic inference of polymorphic lock types. *Science of Computer Programming (SCP)*, 58(3):366–383, December 2005. Special Issue on Concurrency and Synchronization in Java programs.
13. Michael Hicks, Angelos D. Keromytis, and Jonathan M. Smith. A secure PLAN. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 33(3):413–426, August 2003. Special Issue on Technologies Promoting Computational Intelligence, Openness and Programmability in Networks and Internet Services.
14. D. Scott Alexander, William A. Arbaugh, Michael Hicks, Pankaj Kakkar, Angelos Keromytis, Jonathan T. Moore, Carl A. Gunter, Scott M. Nettles, and Jonathan M. Smith. The SwitchWare active network architecture. *IEEE Network*, 12(3):29–36, 1998. Special issue on Active and Controllable Networks.

b.’ Articles in Refereed Conferences and Workshops

15. Chang Liu[†], Michael Hicks, Austin D Harris, Mohit Tiwari, Martin Maas, and Elaine Shi. Ghost rider: A hardware-software system for memory trace oblivious computation. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, March 2015. Acceptance rate 48/278 (17.3%).
16. Luís Pina[†], Luís Veiga, and Michael Hicks. Rubah: DSU for java on a stock JVM. In *Proceedings of the ACM Conference on Object-Oriented Programming Languages, Systems, and Applications (OOPSLA)*, October 2014. Acceptance rate 52/181 (28.7%).
17. Piotr Mardziel[†], Mário S. Alvim, and Michael Hicks. Adversary gain vs. defender loss in quantified information flow. In *(Unofficial) Proceedings of the International Workshop on Foundations of Computer Security (FCS)*, July 2014.
18. Matthew Hammer[†], Yit Phang Khoo[†], Michael Hicks, and Jeffrey S. Foster. Adaption: Composable, demand-driven incremental computation. In *Proceedings of the ACM Conference on Programming Language Design and Implementation (PLDI)*, June 2014. Acceptance rate 52/287 (18.1%).
19. Chang Liu[†], Yan Huang, Elaine Shi, Jonathan Katz, and Michael Hicks. Automating efficient RAM-model secure computation. In *Proceedings of the IEEE Symposium on Security and Privacy (Oakland)*, May 2014. Acceptance rate 44/324 (13.6%).
20. Aseem Rastogi[†], Matthew A. Hammer[†], and Michael Hicks. Wysteria: A programming language for generic, mixed-mode multiparty computations. In *Proceedings of the IEEE Symposium on Security and Privacy (Oakland)*, May 2014. Acceptance rate 44/324 (13.6%).
21. Piotr Mardziel[†], Mario Alvim, Michael Hicks, and Michael Clarkson. Quantifying information flow for dynamic secrets. In *Proceedings of the IEEE Symposium on Security and Privacy (Oakland)*, May 2014. Acceptance rate 44/324 (13.6%).

22. Andrew Miller[†], Michael Hicks, Jonathan Katz, and Elaine Shi. Authenticated data structures, generically. In *Proceedings of the ACM Conference on Principles of Programming Languages (POPL)*, January 2014. Acceptance rate 51/220 (23.2%).
23. Aseem Rastogi[†], Piotr Mardziel[†], Matthew Hammer[†], and Michael Hicks. Knowledge inference for optimizing secure multi-party computation. In *Proceedings of the ACM SIGPLAN Workshop on Programming Languages and Analysis for Security (PLAS)*, June 2013.
24. Chang Liu[†], Michael Hicks, and Elaine Shi. Memory trace oblivious program execution. In *Proceedings of the Computer Security Foundations Symposium (CSF)*, June 2013. Acceptance rate 19/73 (26%).
Winner of the NSA's 2014 Best Scientific Cybersecurity Paper Competition
25. Luis Pina and Michael Hicks. Rubah: Efficient, general-purpose dynamic software updating for Java. In *Proceedings of the Workshop on Hot Topics in Software Upgrades (HotSWUp)*, June 2013.
26. Yit Phang Khoo,[†] Jeffrey S. Foster, and Michael Hicks. Expositor: Scriptable time-travel debugging with first class traces. In *Proceedings of the International Conference on Software Engineering (ICSE)*, May 2013. Acceptance rate 85/461 (18.5%).
27. Mudhakar Srivatsa and Michael Hicks. Deanonymizing mobility traces: Using a social network as a side-channel. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, October 2012. Acceptance rate 80/423 (18.9%).
28. Christopher M. Hayden[†], Edward K. Smith[†], Michail Denchev^{*}, Michael Hicks, and Jeffrey S. Foster. Kitsune: Efficient, general-purpose dynamic software updating for C. In *Proceedings of the ACM Conference on Object-Oriented Programming Languages, Systems, and Applications (OOPSLA)*, October 2012. Acceptance rate 57/228 (25%).
29. Stephen Magill^{*}, Michael Hicks, Suriya Subramanian^{*}, and Kathryn S. McKinley. Automating object transformations for dynamic software updating. In *Proceedings of the ACM Conference on Object-Oriented Programming Languages, Systems, and Applications (OOPSLA)*, October 2012. Acceptance rate 57/228 (25%).
30. Piotr Mardziel[†], Michael Hicks, Jonathan Katz, and Mudhakar Srivatsa. Knowledge-oriented secure multiparty computation. In *Proceedings of the ACM SIGPLAN Workshop on Programming Languages and Analysis for Security (PLAS)*, June 2012. Acceptance rate 9/12 (75%). Number of submissions lower than usual due to location (China).
31. Edward K. Smith[†], Michael Hicks, and Jeffrey S. Foster. Towards standardized benchmarks for dynamic software updating systems. In *Proceedings of the Workshop on Hot Topics in Software Upgrades (HotSWUp)*, pages 11–15, June 2012. Acceptance rate 9/13 (69%).
32. Christopher M. Hayden[†], Karla Saur[†], Michael Hicks, and Jeffrey S. Foster. A study of dynamic software update quiescence for multithreaded programs. In *Proceedings of the Workshop on Hot Topics in Software Upgrades (HotSWUp)*, pages 6–10, June 2012. Acceptance rate 9/13 (69%).
33. Christopher M. Hayden[†], Stephen Magill[†], Michael Hicks, Nate Foster, and Jeffrey S. Foster. Specifying and verifying the correctness of dynamic software updates. In *Proceedings of the International Conference on Verified Software: Theories, Tools, and Experiments (VSTTE)*, pages 278–293, January 2012. Acceptance rate 20/54 (37%).
34. Nataliya Guts[†], Michael Hicks, Nikhil Swamy, and Daan Leijen. A demo of Coco: a compiler for monadic coercions in ML. In *Informal proceedings of the ML Workshop*, September 2011.

35. Nikhil Swamy, Nataliya Guts[†], Daan Leijen, and Michael Hicks. Lightweight monadic programming in ML. In *Proceedings of the ACM International Conference on Functional Programming (ICFP)*, pages 15–27, September 2011. Acceptance rate 33/92 (35.9%). Featured on the *Lambda the Ultimate* blog, July 2011
36. Kin-Keung Ma[†], Yit Phang Khoo[†], Jeffrey S. Foster, and Michael Hicks. Directed symbolic execution. In Eran Yahav, editor, *Proceedings of the Static Analysis Symposium (SAS)*, volume 6887 of *Lecture Notes in Computer Science*, pages 95–111. Springer, September 2011. Acceptance rate 22/67 (32.8%).
37. Piotr Mardziel[†], Stephen Magill[†], Michael Hicks, and Mudhakar Srivatsa. Dynamic enforcement of knowledge-based security policies. In *Proceedings of the Computer Security Foundations Symposium (CSF)*, pages 114–128, June 2011. Acceptance rate 21/81 (25.9%).
38. Christopher M. Hayden[†], Edward K. Smith[†], Michael Hicks, and Jeffrey S. Foster. State transfer for clear and efficient runtime upgrades. In *Proceedings of the Workshop on Hot Topics in Software Upgrades (HotSWUp)*, pages 179–184, April 2011. Acceptance rate 9/17 (52.9%).
39. Jong hoon (David) An[†], Avik Chaudhuri^{*}, Jeffrey S. Foster, and Michael Hicks. Position paper: Dynamically inferred types for dynamic languages. In *Informal Proceedings of the Workshop on Scripts to Programs (STOP)*, January 2011.
40. Jong hoon (David) An[†], Avik Chaudhuri^{*}, Jeffrey S. Foster, and Michael Hicks. Dynamic inference of static types for Ruby. In *Proceedings of the ACM Conference on Principles of Programming Languages (POPL)*, pages 459–472, January 2011. Acceptance rate 49/209 (23.4%).
41. Piotr Mardziel[†], Adam Bender^{*}, Michael Hicks, Dave Levin^{*}, Mudhakar Srivatsa, and Jonathan Katz. Secure sharing in distributed information management applications: problems and directions. In *Proceedings of the Annual Conference of the International Technology Alliance (ACITA)*, September 2010.
42. Jean-Phillipe Martin, Michael Hicks, Manuel Costa, Periklis Akritidis, and Miguel Castro. Dynamically checking ownership policies in concurrent C/C++ programs. In *Proceedings of the ACM Conference on Principles of Programming Languages (POPL)*, pages 457–470, January 2010. Acceptance rate 39/206 (19%).
43. Yit Phang Khoo[†], Jeffrey S. Foster, Michael Hicks, and Vibha Sazawal. Triaging checklists: a substitute for a PhD in static analysis. In *Proceedings of the Workshop on the Evaluation and Usability of Programming Languages and Tools (PLATEAU)*, October 2009.
44. Yit Phang Khoo[†], Michael Hicks, Jeffrey S. Foster, and Vibha Sazawal. Directing javascript with arrows. In *Proceedings of the ACM SIGPLAN Dynamic Languages Symposium (DLS)*, pages 49–58, October 2009. Acceptance rate 10/27 (37%).
45. Michael Furr^{*}, Jong hoon (David) An^{*}, Jeffrey S. Foster, and Michael Hicks. The Ruby intermediate language. In *Proceedings of the ACM SIGPLAN Dynamic Languages Symposium (DLS)*, pages 89–98, October 2009. Acceptance rate 10/27 (37%).
46. Nikhil Swamy, Michael Hicks, and Gavin S. Bierman. A theory of typed coercions and its applications. In *Proceedings of the ACM International Conference on Functional Programming (ICFP)*, pages 329–340, August 2009. Acceptance rate 26/85 (31%).

47. Pavlos Papageorge[†], Justin McCann^{*}, and Michael Hicks. Passive aggressive measurement with MGRP. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, pages 279–290, August 2009. Acceptance rate 27/270 (10%).
48. Brian J. Corcoran[†], Nikhil Swamy[‡], and Michael Hicks. Cross-tier, label-based security enforcement for web applications. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 269–282. June 2009. Acceptance rate 63/397 (16%).
49. Iulian Neamtiu[†] and Michael Hicks. Safe and timely dynamic updates for multi-threaded programs. In *Proceedings of the ACM Conference on Programming Language Design and Implementation (PLDI)*, pages 13–24. June 2009. Acceptance rate 41/196 (21%).
50. Suriya Subramanian^{*}, Michael Hicks, and Kathryn S. McKinley. Dynamic software updates for Java: A VM-centric approach. In *Proceedings of the ACM Conference on Programming Language Design and Implementation (PLDI)*, pages 1–12. June 2009. Acceptance rate 41/196 (21%).
51. Michael Furr^{*}, Jong hoon (David) An^{*}, Jeffrey S. Foster, and Michael Hicks. Static type inference for Ruby. In *Proceedings of the ACM Symposium on Applied Computing, Object-oriented Programming Languages and Systems Track (OOPS)*, pages 1859–1866, March 2009. Acceptance rate 4/9 (44%) for OOPS track, 315/1084 (29%) overall.
52. Dave King^{*}, Boniface Hicks, Michael Hicks, and Trent Jaeger. Implicit flows: Can’t live with ’em, can’t live without ’em. In R. Sekar and Arun K. Pujari, editors, *Proceedings of the International Conference on Information Systems Security (ICISS)*, volume 5352 of *Lecture Notes in Computer Science*, pages 56–70. Springer, December 2008. Acceptance rate 21/81 (26%).
53. Yit Phang Khoo[†], Jeffrey S. Foster, Michael Hicks, and Vibha Sazawal. Path projection for user-centered static analysis tools. In *Proceedings of the ACM Workshop on Program Analysis for Software Tools and Engineering (PASTE)*, pages 57–63, November 2008. Acceptance rate 13/26 (50%).
54. Polyvios Pratikakis[†], Jeffrey S. Foster, Michael Hicks, and Iulian Neamtiu[†]. Formalizing soundness of contextual effects. In Otmane Ait Mohamed, César Muñoz, and Sofiène Tahar, editors, *Proceedings of the International Conference on Theorem Proving in Higher Order Logics (TPHOLs)*, volume 5170 of *Lecture Notes in Computer Science*, pages 262–277. Springer, August 2008. Acceptance rate 16/45 (36%).
55. Nikhil Swamy[†] and Michael Hicks. Verified enforcement of automaton-based information release policies. In *Proceedings of the ACM SIGPLAN Workshop on Programming Languages and Analysis for Security (PLAS)*, pages 21–32, June 2008. Chosen as one of two *Distinguished papers* for the workshop. Acceptance rate 13/24 (54%).
56. Nikhil Swamy[†], Brian Corcoran[†], and Michael Hicks. Fable: A language for enforcing user-defined security policies. In *Proceedings of the IEEE Symposium on Security and Privacy (Oakland)*, pages 369–383, May 2008. Acceptance rate 28/249 (11.2%).
57. Iulian Neamtiu[†], Michael Hicks, Jeffrey S. Foster, and Polyvios Pratikakis[†]. Contextual effects for version-consistent dynamic software updating and safe concurrent programming. In *Proceedings of the ACM Conference on Principles of Programming Languages (POPL)*, pages 37–50. January 2008. Acceptance rate 35/212 (16.5%).
58. Brian Corcoran[†], Nikhil Swamy[†], and Michael Hicks. Combining provenance and security policies in a web-based document management system. In *On-line Proceedings of the Workshop on Principles of Provenance (ProPr)*, November 2007. <http://homepages.inf.ed.ac.uk/jcheney/propr/>.

59. Nick L. Petroni, Jr.[†] and Michael Hicks. Automated detection of persistent kernel control-flow attacks. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, pages 103–115, October 2007. Acceptance rate 55/303 (18%).
60. Nikhil Swamy[†], Michael Hicks, and Simon Tsang. Verified enforcement of security policies for cross-domain information flows. In *Proceedings of the 2007 Military Communications Conference (MILCOM)*, October 2007.
61. Jeffrey S. Foster, Michael W. Hicks, and William Pugh. Improving software quality with static analysis. In *Proceedings of the ACM Workshop on Program Analysis for Software Tools and Engineering (PASTE)*, pages 83–84, June 2007. Acceptance rate 4/4 (100%) for research presentations category.
62. Trevor Jim, Nikhil Swamy[†], and Michael Hicks. Defeating scripting attacks with browser-enforced embedded policies. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 601–610, May 2007. Acceptance rate 10/63 (16%) for the security track, 111/740 (15%) overall.
63. Saurabh Srivastava[†], Michael Hicks, and Jeffrey S. Foster. Modular information hiding and type safety for C. In *Proceedings of the ACM Workshop on Types in Language Design and Implementation (TLDI)*, pages 3–14, January 2007. Acceptance rate 6/12 (50%).
64. Polyvios Pratikakis[†], Jeffrey S. Foster, and Michael Hicks. Existential label flow inference via CFL reachability. In Kwangkeun Yi, editor, *Proceedings of the Static Analysis Symposium (SAS)*, volume 4134 of *Lecture Notes in Computer Science*, pages 88–106. Springer-Verlag, August 2006. Acceptance rate 23/80 (29%).
65. Nikhil Swamy[†], Michael Hicks, Stephen Tse^{*}, and Steve Zdancewic. Managing policy updates in security-typed languages. In *Proceedings of the Computer Security Foundations Workshop (CSFW)*, pages 202–216, July 2006. Acceptance rate 25/96 (26%). Note: though termed a workshop, based on citation rate and number of submissions, CSFW has effectively been a conference for many years; as of 2007, CSFW has been renamed CSF and is termed a symposium.
66. Iulian Neamtiu[†], Michael Hicks, Gareth Stoye^{*}, and Manuel Oriol[†]. Practical dynamic software updating for C. In *Proceedings of the ACM Conference on Programming Language Design and Implementation (PLDI)*, pages 72–83, June 2006. Acceptance rate 36/174 (21%).
67. Polyvios Pratikakis[†], Jeffrey S. Foster, and Michael Hicks. Context-sensitive correlation analysis for detecting races. In *Proceedings of the ACM Conference on Programming Language Design and Implementation (PLDI)*, pages 320–331, June 2006. Acceptance rate 36/174 (21%).
68. Boniface Hicks, Dave King, Patrick McDaniel, and Michael Hicks. Trusted declassification: high-level policy for a security-typed language. In *Proceedings of the ACM SIGPLAN Workshop on Programming Languages and Analysis for Security (PLAS)*, pages 65–74, June 2006. Acceptance rate 10/18 (56%).
69. Michael Hicks, Jeffrey S. Foster, and Polyvios Pratikakis[†]. Inferring locking for atomic sections. In *On-line Proceedings of the ACM SIGPLAN Workshop on Languages, Compilers, and Hardware Support for Transactional Computing (TRANSACT)*, June 2006. <http://www.cs.purdue.edu/homes/jv/events/TRANSACT/transact-06.tgz>. Acceptance rate 10/19 (53%).
70. Michael Hicks, Boniface Hicks, Stephen Tse, and Steve Zdancewic. Dynamic updating of information-flow policies, March 2005. In *Proceedings of the International Workshop on Foundations of Computer Security (FCS)*, pages 7–18, June 2005. Acceptance rate 11/30 (37%).

71. Iulian Neamtiu[†], Jeffrey S. Foster, and Michael Hicks. Understanding source code evolution using abstract syntax tree matching. In *Proceedings of the International Workshop on Mining Software Repositories (MSR)*, pages 1–5, May 2005. Acceptance rate 22/38 (58%).
72. Manuel Oriol[†] and Michael Hicks. Tagged sets: a secure and transparent coordination medium. In Jean-Marie Jacquet and Gian Pietro Picco, editors, *Proceedings of the International Conference on Coordination Models and Languages (COORDINATION)*, volume 3454 of *Lecture Notes in Computer Science*, pages 252–267. Springer-Verlag, April 2005. Acceptance rate 19/88 (22%).
73. Pavlos Papageorgiou[†] and Michael Hicks. Merging network measurement with data transport (extended abstract). In *Proceedings of the IEEE Passive/Active Measurement Workshop (PAM)*, volume 3431, pages 368–371. Springer-Verlag, March 2005. Acceptance rate 36/84 (43%).
74. Gareth Stoyle^{*}, Michael Hicks, Gavin Bierman, Peter Sewell, and Iulian Neamtiu[†]. *Mutatis Mutandis*: Safe and flexible dynamic software updating. In *Proceedings of the ACM Conference on Principles of Programming Languages (POPL)*, pages 183–194, January 2005. Acceptance rate 31/172 (18%).
75. Polyvios Pratikakis[†], Jaime Spacco^{*}, and Michael Hicks. Transparent proxies for Java futures. In *Proceedings of the ACM Conference on Object-Oriented Programming Languages, Systems, and Applications (OOPSLA)*, pages 206–223, October 2004. Acceptance rate 27/173 (16%).
76. Michael Hicks, Greg Morrisett, Dan Grossman, and Trevor Jim. Experience with safe manual memory management in cyclone. In *Proceedings of the ACM International Symposium on Memory Management (ISMM)*, pages 73–84, October 2004. Acceptance rate 15/43 (35%).
77. James Rose[†], Nikhil Swamy[†], and Michael Hicks. Dynamic inference of polymorphic lock types. In *Proceedings of the ACM Conference on Principles of Distributed Computing (PODC) Workshop on Concurrency and Synchronization in Java Programs (CSJP)*, pages 18–25, July 2004. Acceptance rate 11/24 (46%).
78. Gavin Bierman, Michael Hicks, Peter Sewell, Gareth Stoyle, and Keith Wansbrough. Dynamic rebinding for marshalling and update with destruct-time λ . In *Proceedings of the ACM International Conference on Functional Programming (ICFP)*, pages 99–110, August 2003. Acceptance rate 24/95 (25%).
79. Gavin Bierman, Michael Hicks, Peter Sewell, and Gareth Stoyle. Formalizing dynamic software updating. In *On-line Proceedings of the Second International Workshop on Unanticipated Software Evolution (USE)*, April 2003. <http://www.informatik.uni-bonn.de/~gk/use/2003/Papers/papers.html>.
80. Michael Hicks, Adithya Nagarajan[†], and Robbert van Renesse. User-specified adaptive scheduling in a streaming media network. In *Proceedings of the IEEE Conference on Open Architectures and Network Programming (OPENARCH)*, pages 87–96. April 2003. Acceptance rate 12/50 (24%).
81. Seo-Kyu Song, Stephen Shannon, Michael Hicks, and Scott Nettles. Evolution in action: Using active networking to evolve network support for mobility. In James Sterbenz, Osamu Takada, Christian Tschudin, and Bernhard Plattner, editors, *Proceedings of the Fourth International Working Conference on Active Networks (IWAN)*, volume 2546 of *Lecture Notes in Computer Science*, pages 146–161. Springer-Verlag, December 2002. Acceptance rate 20/53 (38%).
82. Trevor Jim, Greg Morrisett, Dan Grossman, Michael Hicks, James Cheney, and Yanling Wang. Cyclone: A safe dialect of C. In *Proceedings of the USENIX Annual Technical Conference*. USENIX, pages 275–288. June 2002. Acceptance rate 25/107 (23%).

83. Dan Grossman, Greg Morrisett, Trevor Jim, Michael Hicks, Yanling Wang, and James Cheney. Region-based memory management in Cyclone. In *Proceedings of the ACM Conference on Programming Language Design and Implementation (PLDI)*, pages 282–293. ACM, June 2002. Acceptance rate 28/169 (17%).
84. Michael Hicks, Jonathan T. Moore, David Wetherall, and Scott Nettles. Experiences with capsule-based active networking. In *Proceedings of the DARPA Active Networks Conference and Exposition (DANCE)*, pages 16–24. IEEE, May 2002.
85. Michael Hicks, Jonathan T. Moore, and Scott Nettles. Compiling PLAN to SNAP. In Ian W. Marshall, Scott Nettles, and Naoki Wakamiya, editors, *Proceedings of the Third International Working Conference on Active Networks (IWAN)*, volume 2207 of *Lecture Notes in Computer Science*, pages 134–151. Springer-Verlag, October 2001. Acceptance rate 10/22 (45%).
86. Michael Hicks, Jonathan T. Moore, and Scott Nettles. Dynamic software updating. In *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, pages 13–23. ACM, June 2001. Acceptance rate 30/144 (21%).
87. Jonathan T. Moore, Michael Hicks, and Scott Nettles. Practical programmable packets. In *Proceedings of the Twentieth IEEE Computer and Communication Society INFOCOM Conference*, pages 41–50. IEEE, April 2001. Acceptance rate 192/830 (23%).
88. Michael Hicks and Scott Nettles. Active networking means evolution (or enhanced extensibility required). In Hiroshi Yashuda, editor, *Proceedings of the Second International Working Conference on Active Networks (IWAN)*, volume 1942 of *Lecture Notes in Computer Science*, pages 16–32. Springer-Verlag, October 2000.
89. K. G. Anagnostakis, M. W. Hicks, S. Ioannidis, A. D. Keromytis, and J. M. Smith. Scalable resource control in active networks. In Hiroshi Yashuda, editor, *Proceedings of the Second International Working Conference on Active Networks (IWAN)*, volume 1942 of *Lecture Notes in Computer Science*, pages 343–358. Springer-Verlag, October 2000.
90. Michael Hicks, Stephanie Weirich, and Karl Cray. Safe and flexible dynamic linking of native code. In Robert Harper, editor, *Types in Compilation*, volume 2071 of *Lecture Notes in Computer Science*. Springer-Verlag, September 2000. Selections from the Third ACM SIGPLAN workshop of the same name.
91. Pankaj Kakkar, Michael Hicks, Jonathan T. Moore, and Carl A. Gunter. Specifying the PLAN network programming language. In *Higher Order Operational Techniques in Semantics (HOOTS)*, volume 26 of *Electronic Notes in Theoretical Computer Science*, pages 87–104. Elsevier, September 1999.
92. Michael Hicks, Suresh Jagannathan, Richard Kelsey, Jonathan T. Moore, and Cristian Ungureanu. Transparent communication for distributed objects in java. In *Proceedings of the ACM SIGPLAN Java Grande Conference*, pages 160–170. ACM, June 1999. Acceptance rate 20/38 (53%).
93. Michael Hicks and Angelos D. Keromytis. A secure PLAN. In Stefan Covaci, editor, *Proceedings of the First International Working Conference on Active Networks IWAN*, volume 1653 of *Lecture Notes in Computer Science*, pages 307–314. Springer-Verlag, June 1999. Reprinted with extensions in DARPA Active Networks Conference and Exposition (DANCE).
94. Michael Hicks, Jonathan T. Moore, D. Scott Alexander, Carl A. Gunter, and Scott Nettles. PLANet: An active internetwork. In *Proceedings of the Eighteenth IEEE Computer and Communication Society INFOCOM Conference*, pages 1124–1133. IEEE, March 1999. Acceptance rate 184/600 (31%).

95. Michael Hicks, Luke Hornof, Jonathan T. Moore, and Scott Nettles. A study of large object spaces. In *Proceedings of the ACM SIGPLAN International Symposium on Memory Management (ISMM)*, pages 138–145. ACM, October 1998.
96. Michael Hicks, Pankaj Kakkar, Jonathan T. Moore, Carl A. Gunter, and Scott Nettles. PLAN: A packet language for active networks. In *Proceedings of the Third ACM SIGPLAN International Conference on Functional Programming Languages (ICFP)*, pages 86–93. ACM, September 1998. Acceptance rate 30/77 (39%).
97. D. Scott Alexander, Michael W. Hicks, Pankaj Kakkar, Angelos D. Keromytis, Marianne Shaw, Jonathan T. Moore, Carl A. Gunter, Trevor Jim, Scott M. Nettles, and Jonathan M. Smith. The SwitchWare Active Network Implementation. In *Notes of the ACM SIGPLAN Workshop on ML*, pages 67–76, September 1998.
98. Michael Hicks, Pankaj Kakkar, Jonathan T. Moore, Carl A. Gunter, and Scott Nettles. Network programming with PLAN. In Luca Cardelli, editor, *Proceedings of the IEEE Workshop on Internet Programming Languages*, volume 1686 of *Lecture Notes in Computer Science*, pages 127–143. Springer-Verlag, May 1998.
99. Michael W. Hicks, Jonathan T. Moore, and Scott M. Nettles. The measured cost of copying garbage collection mechanisms. In *Proceedings of the ACM SIGPLAN International Conference on Functional Programming (ICFP)*, pages 292–305. ACM, June 1997. Acceptance rate 25/78 (32%).

c. Monographs, Reports, and Extension Publications

Technical Reports

Most of these papers contain additional material, such as formal proofs or further experiments, referenced by the refereed conference and journal papers listed above.

100. Piotr Mardziel, Mario Alvim, Michael Hicks, and Michael Clarkson. Quantifying information flow for dynamic secrets (extended version). Technical Report CS-TR-5035, Department of Computer Science, the University of Maryland, College Park, May 2014.
101. Aseem Rastogi, Matthew A. Hammer, and Michael Hicks. Wysteria: A programming language for generic, mixed-mode multiparty computations (extended version). Technical Report CS-TR-5034, Department of Computer Science, the University of Maryland, College Park, May 2014.
102. Matthew Hammer, Yit Phang Khoo, Michael Hicks, and Jeffrey S. Foster. Adaption: Composable, demand-driven incremental computation. Technical Report CS-TR-5027, Department of Computer Science, the University of Maryland, College Park, July 2013.
103. Yit Phang Khoo, Jeffrey S. Foster, and Michael Hicks. Expositor: Scriptable Time-Travel Debugging with First-Class Traces. Technical Report CS-TR-5021, Department of Computer Science, University of Maryland, College Park, February 2013.
104. Nataliya Guts, Michael Hicks, Nikhil Swamy, Daan Leijen, and Gavin Bierman. Polymonads. Technical Report XXX, University of Maryland Department of Computer Science, July 2012.
105. Christopher M. Hayden, Stephen Magill, Michael Hicks, Nate Foster, and Jeffrey S. Foster. Specifying and verifying the correctness of dynamic software updates. Technical Report CS-TR-4997, University of Maryland Department of Computer Science, November 2011. Extended version of VSTTE'12 paper with proofs of theorems and additional discussion.

106. Christopher M. Hayden, Edward K. Smith, Eric A. Hardisty, Michael Hicks, and Jeffrey S. Foster. Evaluating dynamic software update safety using efficient systematic testing. Technical Report CS-TR-4993, University of Maryland, Department of Computer Science, September 2011.
107. Jonathan Turpie, Elnatan Reisner, Jeffrey S. Foster, and Michael Hicks. Multiotter: Multiprocess symbolic execution. Technical Report CS-TR-4982, University of Maryland Department of Computer Science, August 2011.
108. Nikhil Swamy, Nataliya Guts, Daan Leijen, and Michael Hicks. Lightweight monadic programming in ML. Technical Report MSR-TR-2011-039, Microsoft Research, May 2011.
109. Kin-Keung Ma, Yit Phang Khoo, Jeffrey S. Foster, and Michael Hicks. Directed symbolic execution. Technical Report CS-TR-4979, University of Maryland Department of Computer Science, April 2011.
110. Piotr Mardziel, Stephen Magill, Michael Hicks, and Mudhakar Srivatsa. Dynamic enforcement of knowledge-based security policies. Technical Report CS-TR-4978, University of Maryland Department of Computer Science, April 2011. Extended version contains proofs of theorems.
111. Jong hoon (David) An, Avik Chaudhuri, Jeffrey S. Foster, and Michael Hicks. Dynamic inference of static types for Ruby. Technical Report CS-TR-4965, University of Maryland Department of Computer Science, July 2010. Extended version of POPL 2011 paper contains proofs of theorems.
112. Michael Hicks and Jeffrey S. Foster. Adapting Scrum to managing a research group. Technical Report CS-TR-4966, University of Maryland, Department of Computer Science, September 2010.
113. Christopher M. Hayden, Eric A. Hardisty, Michael Hicks, and Jeffrey S. Foster. A testing based empirical study of dynamic software update safety restrictions. Technical Report CS-TR-4949, University of Maryland, Department of Computer Science, November 2009.
114. Yit Phang Khoo, Michael Hicks, Jeffrey S. Foster, and Vibha Sazawal. Directing javascript with arrows (functional pearl). Technical Report CS-TR-4923, University of Maryland, Department of Computer Science, August 2008. Extended version of ICFP 2008 poster summary.
115. Yit Phang Khoo, Jeffrey S. Foster, Michael Hicks, and Vibha Sazawal. Path projection for user-centered static analysis tools (long version). Technical Report CS-TR-4919, University of Maryland, Department of Computer Science, August 2008.
116. Nikhil Swamy[†] and Michael Hicks. Verified enforcement of automaton-based information release policies. Technical Report CS-TR-4906, University of Maryland, Department of Computer Science, 2008. Full version of PLAS 08 paper.
117. Nick L. Petroni, Jr.[†] and Michael Hicks. Automated detection of persistent kernel control-flow attacks. Technical Report CS-TR-4880, Department of Computer Science, University of Maryland, October 2007.
118. Iulian Neamtiu[†], Michael Hicks, Jeffrey S. Foster, and Polyvios Pratikakis[†]. Contextual effects for version-consistent dynamic software updating and safe concurrent programming (extended version). Technical Report CS-TR-4875, University of Maryland, Department of Computer Science, July 2007.
119. Nikhil Swamy[†] and Michael Hicks. Fable: A language for enforcing user-defined security policies (extended version). Technical Report CS-TR-4876, University of Maryland, Department of Computer Science, July 2007.

120. Saurabh Srivastava[†], Michael Hicks, and Jeffrey S. Foster. Appendix to CMod: Modular information hiding and type-safe linking for C. Technical Report (TBD), University of Maryland, College Park, 2007.
121. Michael Hicks, Nikhil Swamy[†], and Simon Tsang. Toward specifying and validating cross-domain policies. Technical Report CS-TR-4870, Department of Computer Science, University of Maryland, April 2007.
122. Nikhil Swamy[†], Michael Hicks, Stephen Tse, and Steve Zdancewic. Managing policy updates in security-typed languages (extended version). Technical Report CS-TR-4793, Department of Computer Science, University of Maryland, August 2006. Contains full formal development and proofs for CSFW 06 paper.
123. Saurabh Srivastava[†], Michael Hicks, Jeffrey S. Foster, and Bhargav Kanagal*. Defining and enforcing C's module system. Technical Report CS-4816, Department of Computer Science, University of Maryland, July 2006. Contains full formal development and proofs for TLDI 07 paper.
124. Polyvios Pratikakis[†], Jeffrey S. Foster, and Michael Hicks. Context-sensitive correlation analysis for detecting races (extended version). Technical Report CS-TR-4789, Department of Computer Science, University of Maryland, June 2006. Extends PLDI 2006 paper with full formal development.
125. Boniface Hicks, Dave King, Patrick McDaniel, and Michael Hicks. Trusted declassification: high-level policy for a security-typed language (extended version). Technical Report NAS-TR-033-2006, Department of Computer Science and Engineering, the Pennsylvania State University, June 2006. Extended version of the PLAS 2006 paper with full formal development.
126. Michael W. Hicks, Pankaj Kakkar, Jonathan T. Moore, Carl A. Gunter, and Scott M. Nettles. PLAN: A packet language for active networks (extended version). Unpublished manuscript, May 2006. Unifies and consolidates ICFP 98, IPL 98, and Allerton 99 papers.
127. Iulian Neamtiu[†], Michael Hicks, Gareth Stoye*, and Manuel Oriol[†]. Practical dynamic software updating for C (extended version). Technical Report CS-TR-4790, Department of Computer Science, University of Maryland, March 2006. Extended version of PLDI 06 paper.
128. Polyvios Pratikakis[†], Michael Hicks, and Jeffrey S. Foster. Existential label flow inference via CFL reachability. Technical Report CS-TR-4700, Department of Computer Science, University of Maryland, July 2005. Contains full formal development and proofs of SAS 06 paper.
129. Michael Hicks, Greg Morrisett, Dan Grossman, and Trevor Jim. Safe and flexible memory management in Cyclone. Technical Report CS-TR-4514, University of Maryland Department of Computer Science, July 2003.
130. Michael Hicks, Adithya Nagarajan[†], and Robbert van Renesse. User-specified adaptive scheduling in a streaming media network. Technical Report CS-TR-4430, Department of Computer Science, University of Maryland, March 2003. Extends OPENARCH '03 paper with full algorithmic description and analysis.
131. Dan Grossman, Greg Morrisett, Trevor Jim, Michael Hicks, Yanling Wang, and James Cheney. Formal type soundness for Cyclone's region system. Technical Report CS 2001-1856, Cornell University, November 2001. Proves formal results of PLDI '02 paper.

132. Michael Hicks and Stephanie Weirich. A calculus for dynamic loading. Technical Report MS-CIS-00-07, University of Pennsylvania, April 2000. Formalism and proof for system presented in *Types in Compilation* paper.
133. Michael Hicks. Types and Intermediate Representations. Technical Report MS-CIS-98-05, Department of Computer and Information Science, University of Pennsylvania, April 1998.
134. Michael Hicks. PLAN system security. Technical Report MS-CIS-98-25, Department of Computer and Information Science, University of Pennsylvania, April 1998.
135. Jonathan T. Moore, Michael Hicks, and Scott Nettles. General-purpose persistence using flash memory. Unpublished manuscript, April 1997.

Dissertation

136. Michael Hicks. *Dynamic Software Updating*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, August 2001. **Winner of the 2002 ACM SIGPLAN Doctoral Dissertation award.**

d. Book Reviews, Other Articles, and Notes

Invited Articles

137. Michael Hicks. POPL'12 program chair's report (or, how to run a medium-sized conference). *SIGPLAN Notices*, 47(4), April 2012.
138. Christopher M. Hayden, Eric A. Hardisty, Michael Hicks, and Jeffrey S. Foster. Efficient systematic testing for dynamically updatable software. In *Proceedings of the Workshop on Hot Topics in Software Upgrades (HotSWUp)*, October 2009.
139. Jonathan T. Moore, Michael Hicks, and Scott M. Nettles. Chunks in PLAN: Language support for programs as packets. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing*, September 1999.
140. Dan Grossman, Michael Hicks, Greg Morrisett, and Trevor Jim. Cyclone: a type-safe dialect of C. *C/C++ Users Journal*, 23(1), January 2005.

e. Talks, Abstracts, Tutorials, and Other Professional Papers Presented

i. Invited Talks

“Authenticated Data Structures, Generically,”

- Dagstuhl Seminar on the Synergy between Programming Languages and Cryptography, Wadern, Germany, December 1, 2014.
- Colloquium talk, Max Planck Institute for Software Systems (MPI-SWS), Saarbrücken, Germany, December 5, 2014.

“Memory-Trace Oblivious Program Execution,”

- Dagstuhl Seminar on the Synergy between Programming Languages and Cryptography, Wadern, Germany, December 1, 2014.
- NSA Best Scientific Cybersecurity Award Paper presentation, September 18, 2014.

“Build-it, break-it, fix-it: Recapping the results,” 5th Annual NIST National Initiative on Cybersecurity Education (NICE) Workshop on “Shaping the Future of Cybersecurity Education,” Columbia, Maryland, November 5, 2014.

“Rubah: DSU for Java on a stock JVM,”

- Invited talk, Laboratory for Telecommunications Sciences, June 2014.
- Invited talk, Microsoft Research, Redmond, August 2014.

“Adapton: Composable, Demand-Driven Incremental Computation,” invited talk, Microsoft Research, August 2014.

“Wysteria: A Programming Language for Generic, Mixed-Mode Multiparty Computations”

- OOPSLA PC Workshop talk, May 14, 2014.
- Invited talk, Computer Science Department Seminar, Stevens Institute of Technology, February 24, 2014. See <http://www.stevens.edu/news/content/cs-department-seminar-michael-hi>
- USUKITA “bootcamp” talk, London, England, January 5, 2014.

“Expositor: Scriptable time-travel debugging with first class traces,” Software Reliability Group Seminar, Imperial College, London, UK, January 8, 2014. <http://srg.doc.ic.ac.uk/seminars/14-01-hicks/>

“Authenticated Data Structures, Generically,” joint PL/Systems seminar, Microsoft Research, Cambridge, November 22, 2013. See <http://research.microsoft.com/apps/video/dl.aspx?id=205246>

“Secure Computation: Combining PL and Crypto Research,” University of Maryland Undergraduate Research Seminar series, October 25, 2013.

“Build-it, break-it, fix-it: A new security contest,” 4th Annual NIST National Initiative on Cybersecurity Education (NICE) Workshop on “Shaping the Future of Cybersecurity Education,” Gaithersburg, Maryland, September 18, 2013.

“On-line patching for better security, and other security challenges,” Colloquium talk, Johns Hopkins Applied Physics Lab, May 10, 2013. Cf. <http://www.youtube.com/watch?v=eF1NBkSKUzw>

“Expositor: Scriptable time-travel debugging with first class traces,”

- Colloquium talk, Max Planck Institute for Software Systems (MPI-SWS), Saarbrücken, Germany, November 30, 2012.
- Seminar talk, Microsoft Research, Redmond, WA, December 3, 2012. See <http://research.microsoft.com/apps/video/default.aspx?id=179198>

“Directed Symbolic Execution,” seminar talk, École Normale Supérieure, Paris, France, October 11, 2012.

“Kitsune: Efficient, General-purpose Dynamic Software Updating for C,”

- Colloquium talk, Johns Hopkins University, Department of Computer Science, November 12, 2012.
- Seminar talk, Microsoft Research, Redmond, July 13, 2012. See <http://research.microsoft.com/apps/video/default.aspx?id=169667>
- Invited talk at Symantec Research Labs, Herndon, VA, March 21, 2012.
- Invited Tech. talk at Cyberpoint, Inc., Baltimore, MD, March 6, 2012.

“Polymonads: reasoning and inference,” seminar talk, Madrid Institute for Advanced Studies (IMDEA), Madrid, Spain, June 1, 2012.

“Cybersecurity: Past, Present, Looking ahead,” Keynote, Maryland Francis King Carey School of Law Symposium on *Cybersecurity: Safeguarding Information in a Digital Age*, March 30, 2012.

“Dynamic Enforcement of Knowledge-based Security Policies,”

- Georgia Institute of Technology ISC invited talk, April 12, 2013.
- Dagstuhl Seminar on Quantitative Security Analysis, Wadern, Germany, November 27, 2012.
- Seminar talk, Microsoft Research, Cambridge, UK, October 12, 2012. See <http://research.microsoft.com/apps/video/default.aspx?id=174740>
- Dagstuhl Seminar on Web Application Security, Wadern, Germany, October 4, 2012.
- Seminar talk, École Normale Supérieure, Paris, France, May 30, 2012.
- Technical talk, First Maryland Cybersecurity Center (MC2) Symposium, May 17, 2012.
- Colloquium talk, George Washington University, March 20, 2012.
- USUKITA “bootcamp” talk, Weybridge, England, June 22, 2011.
- Guest speaker at University of Maryland CyberClub, Oct 27, 2011.

“Dynamic inference of static types for Ruby,”

- Dagstuhl Seminar on Foundations of Scripting Languages, Wadern, Germany, January 2012.
- Colloquium talk, Department of Computer Science, Cornell University, October 14, 2011.
- Systems Seminar, University of Delaware, December 3, 2010.
- Computing Laboratory seminar, University of Oxford (United Kingdom), September 17, 2010.

“Software Synthesis for cost-effective development of correct, efficient software,” given at the ISAT outbrief to DARPA program managers and the deputy director on October 4, 2011. I co-chaired this ISAT study with Armando Solar-Lezama of MIT.

“Lightweight monadic programming in ML,”

- Seminar, Institut National de Recherche en Informatique et en Automatique (INRIA) Paris-Rocquencourt, June 24, 2011.
- Automated Reasoning Group seminar, University of Cambridge Computer Laboratory, July 1, 2011.

“Evaluating dynamic software update safety using efficient systematic testing,” Computer Science Department seminar, University of Texas at Austin, April 15, 2010.

“Secure Provenance Policies in SELinks,” Workshop on Provenance in Secure and Advanced Computer Systems, University of Edinburgh, Scotland, May 13, 2009.

“Systematic Testing for Dynamically Updatable Software”, Systems Research Group Seminar, University of Cambridge Computer Laboratory, May 7, 2009.

“A Theory of Typed Coercions and its Applications”,

- Departmental Seminar, Chalmers Institute of Technology (Gothenburg, Sweden), April 28, 2009.
- Automated Reasoning Group seminar, University of Cambridge Computer Laboratory, March 17, 2009.

“Type-directed coercion insertion for security enforcement”, Dagstuhl Seminar on Web Application Security, Wadern, Germany, April 2, 2009.

“Practical Dynamic Software Updating for C”,

- Departmental Seminar (Colloquium talk), University of Kent at Canterbury Computer Laboratory, March 10, 2009.
- Departmental Wednesday Seminar (Colloquium talk), University of Cambridge Computer Laboratory, October 15, 2008.

“Cross-tier, Label-based Security Enforcement for Web Applications”, invited talk, Security and Privacy Day @ Stony Brook, SUNY Stony Brook, May 30, 2008.

“Building Secure Web Applications with SELinks”, invited talk, IARPA STONESOUP workshop, May 9, 2008.

“Automated Detection of Persistent Kernel Control-Flow Attacks”, invited talk, Microsoft Research, Silicon Valley, October 2007.

“SELINKS: A language for provably secure web applications”, Distinguished Lecturer talk, IBM T.J. Watson Research Laboratory, July 2007.

“CMOD: Modular Type Safety and Information Hiding for C”

- Seminar talk, Department of Computer Science, The University of Wisconsin, Madison, March 2007.
- Seminar talk, Department of Computer Science, Stanford University, February 2007.
- Microsoft Research, Redmond, Washington, July 2006.
- Triforce Programming Language Research Group Seminar, Harvard University, November 2006.

“LOCKSMITH: Context-sensitive Correlation Analysis for Detecting Races”

- Colloquium talk, Department of Computer Science, University of Massachusetts (Amherst), November 2006.
- Seminar talk, Department of Computer Science, Cornell University, April 2007.

“Making Concurrent Software Safer”, Summer School on Language-based Techniques for Concurrent and Distributed Software, July 2006.

“Static Analysis to Improve Software Quality” (with Jeff Foster), National Research Council of Canada, Institute for Information Technology invited talk, June 2006.

“Cyclone: A Safe Dialect of C”, Trust and Security Seminar, Department of Computer Science, University of Illinois Urbana-Champaign, November 2005.

“Practical Dynamic Software Updating for C”

- Seminar talk, Department of Computer Science and Engineering, University of California, Berkeley, February 2007.
- Colloquium talk, Department of Computer Science, the University of Virginia, December 2006.
- MIT Programming Language Group Seminar, November 2006.
- Colloquium talk, Department of Computer Science, the Pennsylvania State University, November 2005.
- IFIP Working Group 2.4 meeting (invited participant), Jackson’s Mill, West Virginia, October 2005.
- Microsoft Research, Redmond, Washington, August 2005. (See <http://research.microsoft.com/apps/video/default.aspx?id=104614>.)

- IBM T.J. Watson Research Lab, Hawthorne, New York, June 2005.
- “Analyzing code that changes on the fly”, Dagstuhl Seminar on Multi-version Program Analysis, Wadern, Germany, June 2005.
- “Safe and Predictable Dynamic Software Updating”
- Principles of Programming Seminar, Carnegie-Mellon University, Pittsburgh, July 2004.
 - NTT/Docomo Research Laboratory, San Jose, California, August 2004.
- “Transparent Proxies for Java Futures”, Seminar talk, the Johns-Hopkins University, May 2004.
- “Safe and Flexible Memory Management in Cyclone”
- Microsoft Research, Redmond, Washington, February 2004.
 - Short talk at the Second Workshop on Semantics, Program Analysis, and Computing Environments for Memory Management (SPACE), 2004.
 - Dagstuhl Seminar on Language-based Security, Wadern, Germany, October 2003.
 - Colloquium talk, Department of Computer Science, Purdue University, August 2003.
- “Dynamic Software Updating”
- Colloquium talk, The Computer Laboratory at Cambridge University, Cambridge, England, October 2001.
 - Microsoft Research Lab, Cambridge, England, October 2001.
 - ULTRA Seminar, Heriot-Watt University, Edinburgh, Scotland, October 2001.
 - Principles of Programming Seminar, Carnegie-Mellon University, Pittsburgh, December 2000.
- “PLAN: Programming Language for Active Networks,” Colloquium talk, The Pennsylvania State University, State College, Pennsylvania, December 1997.

ii. Posters and Short Presentations (invited)

Michael Hicks. Programming Languages for Reliable, Available, and Secure Software. Invited poster at the 25th National Academy of Sciences Kavli Frontiers of Science Symposium, November 7–9, 2013.

ii. Posters and Short Presentations (Referreed)

- Robert Maxwell and Michael Hicks. Using students to pen test your network (for credit). Presentation at *EDUCAUSE Security Professionals Conference*, Saint Louis, MO, April 2013.
- Yit Phang Khoo, Michael Hicks, Jeffrey S. Foster, and Vibha Sazawal. Directing javascript with arrows (poster summary). In *Poster Proceedings of the ACM International Conference on Functional Programming (ICFP)*, September 2008.
- Iulian Neamtiu[†] and Michael Hicks. Dynamic Software Updating for the Linux Kernel. Work-in-progress poster/presentation at the *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, November 2006.
- Iulian Neamtiu[†], Michael Hicks, Gareth Stoye*, and Manuel Oriol*. Ginseng: A System for Dynamic Software Updating. Poster presented at the *ACM Conference on Programming Language Design and Implementation (PLDI)*, June 2006.
- Pavlos Papageorgiou[†] and Michael Hicks. Merging network measurement with data transport. Poster presented at the *IEEE Passive/Active Measurement Workshop (PAM)*, March 2005.

Amol Despande and Michael Hicks. Toward on-line schema evolution for non-stop systems. Presented at the 11th High Performance Transaction Systems Workshop, September 2005.

Carl Gunter, Michael Hicks, Pankaj Kakkar, Jonathan Moore, Scott Nettles, and Jonathan Smith. PLAN: Language-Based Safety and Security for Active Networks. Work-in-progress poster/presentation at the *ACM Symposium on Operating Systems Principles (SOSP)*, October 1997.

g. Exhibits, Performances, Demonstrations and Other Creative Activities.

Build-it, Break-it, Fix-it, a security-oriented programming contest. First offered August 2014. The nationwide contest pits student “builder” teams against “breaker” teams, where the former tries to build software resilient against attack from the latter. Intended outcomes are to (a) impress upon students the value of “building security in,” and (b) to gather data to research common errors, and to learn what most often correlates with success. <https://builditbreakit.org/>.

h. Original Designs, Plans, Inventions, Software, and/or Patents

Software

Kitsune, developed October 2010–present; a library and compiler for support dynamic software updating services in C programs. Available at <http://kitsune-dsu.com>.

Expositor, developed 2013–present; a framework for scriptable, time-travel debugging. Available at <https://bitbucket.org/khooy/expositor>.

Adapton, developed 2013–present; a framework for demand-driven, incremental computation. Available at <https://bitbucket.org/khooy/adapton.ocaml>.

Wysteria, developed 2013–present; a programming language for secure multiparty computation. Available at <https://bitbucket.org/aseemr/wysteria/wiki/Home>.

LambdaAuth, developed 2014–present; a programming language for authenticated data structures. Available at <http://amiller.github.io/lambda-auth/>.

LockSmith, a tool for statically discovering race conditions in C programs, at <http://www.cs.umd.edu/projects/PL/locksmith/>, developed 2005–present (released June 2006). LockSmith checks that the program consistently follows the *guarded-by* pattern, in which some mutual exclusion lock is held consistently by any thread that accesses a location that may simultaneously be accessed by other threads.

Rubydust, developed October 2009–2013. Rubydust is a Ruby library that can be used to infer types for Ruby methods based on information gathered during testing.

Measurement Manager Protocol (MGRP), at <http://www.cs.umd.edu/projects/MGRP/>, developed 2007–2009 (released May 2010). MGRP is a Linux-kernel extension that permits network measurement tools to be written so that they transparently piggyback on top of existing traffic, to infer high-quality measurements at lower overhead.

Path Projection, at <http://www.cs.umd.edu/projects/PL/PP/>, developed 2008–2012 (released June 2008). Path Projection is novel user interface toolkit that helps users visualize, navigate, and understand program paths, a common component of many static analysis tools’ error reports.

Arrowlets, at <http://www.cs.umd.edu/projects/PL/arrowlets/>, developed 2008–present (released September 2008). Arrowlets is a JavaScript library that uses the concept of arrows to ease the task of composing event-handlers, a common need in Javascript programming.

SELinks, a secure web programming language, at <http://www.cs.umd.edu/projects/PL/selinks>, developed 2007–2010 (released May 2008). SELINKS extends the Links web programming language type system with support for specifying security policies so that type checking is equivalent to security checking: type-correct programs correct enforce their security policies.

CMod, a module system for C programs, at <http://www.cs.umd.edu/projects/PL/CMod>, developed 2006–2010 (released October 2006). CMod wraps the standard C compiler and linker to enforce programming patterns that effectively define a module system for legacy C programs; type safe linking and information hiding are provable consequences.

BEEP, browser modifications to support Browser-enforced Embedded Policies, at <http://www.research.att.com/~trevor/beep.html>, developed 2006–2007 (released February 2007). BEEP is a system by which a web server can embed a security policy within its pages, implemented as a Javascript function. While rendering a page, the browser will only execute script programs that are approved by this function, defeating script injection attacks.

Ginseng, a system for dynamically updating C programs, at <http://www.cs.umd.edu/projects/dsu/>, developed 2004–2011 (released June 2006). Ginseng compiles a C program to accept a dynamic patch that can replace functions, global variables, and type definitions on the fly; Ginseng generates dynamic patches mostly automatically from the deployed and updated source code for the program.

Cyclone, a safe dialect of C for more reliable and secure systems programming, at <http://cyclone.thelanguage.org/>, developed 2001–present. A hallmark of Cyclone is type safety alongside a high degree of control over data layout and memory management.

MediaNet, an overlay network for user-specified adaptations to streaming media, at <http://www.cs.umd.edu/projects/medianet/>, developed 2002–2004. MediaNet is written in Cyclone and employs many of its advanced memory management features.

Java Transparent Proxy Framework, a toolkit for proxy programming, at <http://www.cs.umd.edu/~polyvios/proxyc/>, developed 2004. The framework uses static analysis to track the flow of proxies beginning from user-annotated program points. The analysis results are used to re-write the program, inserting code to wrap and unwrap proxies as necessary.

The Flashed upgradeable webserver and an implementation for dynamically updateable applications, developed 2000–2001. Superseded by Ginseng; this is the dynamic updating implementation I developed as part of my doctoral dissertation, on top of the Popcorn compiler for Typed Assembly Language.

The SNAP Active (Programmable) Network, at <http://www.cis.upenn.edu/~dsl/SNAP/>, developed 2001–2002. SNAP defines a bytecode language for constructing programs that reside in network packets, e.g., to define their routing or quality-of-service. SNAP is distinguished in being efficient with tight controls over worst-case resource usage.

The PLANet Active (Programmable) Network, at <http://www.cis.upenn.edu/~dsl/PLAN/>, developed 1998–2001. PLANet’s packet language PLAN is an ML-like language that can be compiled to SNAP.

The Oscar Garbage Collection Testbed, at <http://www.cis.upenn.edu/~oscar/>, developed 1997–1998. Defines a heap snapshot and replay system for measuring the costs of garbage collection mechanisms.

i. Contracts and Grants

Current

Boombox: Dynamic Software Updates for Software Defined Networks, Google Research Award (gift), \$51,037. September 2014–2015. PI: Michael Hicks.

Connecting the Theory and Practice of Incremental Computation via Servo, gift from Mozilla Corporation, \$88,294. May 2014–2015. co-PIs: Michael Hicks and Jeff Foster.

Establishing a Science of Security Research Lablet at the University of Maryland, NSA, \$1,487,608. February, 2014–February, 2015. Lead PI: Jonathan Katz, co-PI: Michael Hicks (among 11 others).

NSF EDU-1319147 **EDU: Competing to Build Secure Systems**, \$300,000. September 2013-2015. PI: Michael Hicks, co-PIs: Atif Memon, David M. Levin, and Jandelyn D. Plane.

NSF CNS-1314857 **TWC: Medium: Collaborative: DIORE: Digital Insertion and Observation Resistant Execution**, \$799,499. August 2013-2016. PI: Elaine Shi, co-PIs: Michael Hicks and Bobby Bhattacharjee. (This is part of a \$1.2M multi-institution grant with one other PI at UT Austin.)

NSF CNS-1111599 **Practical Secure Two-Party Computation: Techniques, Tools, and Applications**, \$1M. August 2011-2016. PI: Jonathan Katz, co-PI: Michael Hicks. (This is part of a multi-institution grant.)

Secure Information Flows in Hybrid Coalition Networks, US Army Research Lab and UK Ministry of Defence International Technology Alliance in Network and Information Science (USUKITA) program BPP13, \$356,615 PI: Michael Hicks, co-PI Jonathan Katz. May 2013–2015.

Completed

University of Maryland Partnership with the Laboratory of Telecommunications Sciences, Contract Number H9823013D00560002, **Protecting against Malware on Android**, \$371K. April 5, 2012–September 30, 2014. PI: Jeffrey S. Foster, co-PI: Michael Hicks.

NSF CCF-0910530 **SHF: Large: Collaborative Research: Ever Ready: Perpetually Available Software Systems**, \$642K. August 2009–2014. PI: Michael Hicks. (This is part of a multi-institutional team grant with two other PIs for a total award of \$2.3M.)

NSF CCF-0915978 **SHF: Small: User-Centered Software Analysis Tools**, \$500K. September 2009–2013. PI: Jeffrey S. Foster, co-PI: Michael Hicks.

Securing Information Flows, US Army Research Lab and UK Ministry of Defence International Technology Alliance in Network and Information Science (USUKITA) program BPP11, \$333K. co-PIs: Michael Hicks and Jonathan Katz. May 2011–2013.

University of Maryland Partnership with the Laboratory for Telecommunications Sciences (LTS), **Dynamic Software Update Automation for Servers and Event-Driven Programs**, \$407K. January 2010–2013. PI: Michael Hicks, co-PI: Jeff Foster.

NSF CNS-0905419 **RECOVERY: TC: Medium: Collaborative Research: Techniques to Retrofit Legacy Code with Security**, \$300K. September 2009–2012. PI: Michael Hicks. (This is part of a multi-institutional team grant with three other PIs for a total award of \$1.2M.)

NSF CCF-0541036 **Scalable, Precise, and Effective Analyses for Detecting Race Conditions**, \$360K. September 2006–2010. PI: Michael Hicks, co-PI: Jeffrey S. Foster.

US Army Research Lab and UK Ministry of Defence International Technology Alliance in Network and Information Science (USUKITA) program BPP09, \$197K. co-PIs: Michael Hicks and Jonathan Katz. May 2009–2011.

NSF CNS-0346989 **CAREER: Programming Languages for Reliable and Secure Low-level Systems**, \$550K. June 2004–2010. PI: Michael Hicks.

University of Maryland Partnership with the Laboratory for Telecommunications Sciences (LTS), **Safe and Robust Dynamic Software Updating for Real-time Embedded Applications**, \$327K, August 2007–January 2010. PI: Michael Hicks, co-PI: Jeff Foster.

Army Research Lab Communications and Networking Collaborative Technology Alliance (ARL C&N CTA) program TIP-07-4, **End-to-End Security of Cross-Domain Information Flows**, \$245K (funds to Maryland). October 2006–October 2009. PI: Michael Hicks.

ARL C&N CTA program (supported by DARPA DSO), **Provably Enforcing Decentralized, Multi-level Security Policies in Distributed, Web-based Applications**, \$83K. March 1, 2007–September 30, 2009. PI: Michael Hicks.

NSF CCF-0524036 **Collaborative Research: CT-T: Flexible, Decentralized Information-flow Control for Dynamic Environments**, \$280K. September 2005–2009. PI: Michael Hicks. (This is part of a multi-institutional team grant with three other PIs for a total award of \$1.1M.)

DARPA #HR00110610019 **Computer Science Study Group Panelist**, \$77K. March 2006–2007. PI: Michael Hicks.

NSF IIS-0613601 **SoD-HCER: Evaluation of Complex Designs—A Comparative Study**, \$100K. July 2006–2008. PI: Michael Hicks (transferred from Vibha Sazawal in January 2007).

j. Fellowships, Prizes, and Awards

Co-author of paper winning the NSA's Best Scientific Cybersecurity Competition, 2013.

NSF CAREER Award, 2004.

ACM SIGPLAN Doctoral Dissertation award, 2002.

Information Assurance Institute (IAI) Postdoctoral Research Fellowship, Department of Computer Science, Cornell University, 2001.

ARINC Technical Excellence Award, 1995.

Member of Phi Beta Kappa National Honors Society, 1993.

k. Editorships, Editorial Boards, and Reviewing Activities for Journals and other Learned Publications

Associate Editor, ACM Transactions on Programming Languages and Systems (February 2012–2015)

Journal guest co-editor, Logical Methods in Computer Science (LMCS), Special issue on the best papers of the 2010 European Symposium on Programming

Journal reviewer

- ACM Transactions on Programming Languages and Systems (TOPLAS)
- ACM Transactions on Code Optimization (TACO)
- IEEE Transactions on Software Engineering (TSE)
- IEEE Security and Privacy Magazine (S&P)
- ACM Transactions on Modeling and Computer Simulation (TOMACS)
- Elsevier Information Processing Letters (IPL)
- Elsevier Theoretical Computer Science (TCS)
- Elsevier Journal of Systems Software (JSS)
- Elsevier Journal of Object Technology (JOT)
- Springer Acta Informatica
- Software: Practice and Experience (SPE)

Conference external reviewer

- International Conference on Concurrency Theory (CONCUR)
- IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)
- ETAPS International European Conference on Object Oriented Programming (ECOOP).
- ACM International Conference on Functional Programming (ICFP)
- International Conference on Parallel Processing (ICPP)
- IEEE International Parallel and Distributed Processing Symposium (IPDPS)
- International Working Conference on Active Networks (IWAN)
- ACM Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA)
- USENIX Symposium on Operating System Design and Implementation (OSDI)
- USENIX Annual Technical Conference
- USENIX Security Symposium
- ACM Conference on Programming Language Design and Implementation (PLDI)
- ACM Conference on the Principles of Programming Languages (POPL)
- ACM Conference of the Special Interest Group in Communications (SIGCOMM)
- ACM Symposium on Operating Systems Principles (SOSP)

3 Teaching, Mentoring, and Advising

a. Courses Taught

1. Specialized Courses

Semester	Course	# Students (to start)	Title
Fall 2014	CMSC 433	50	Programming Language Technologies and Paradigms
Spring 2014	CMSC 838G	10	Software Security
Fall 2013	CMSC 433	50	Programming Language Technologies and Paradigms
Spring 2013	CMSC 631	26	Program Analysis and Understanding
Spring 2013	CMSC 330	113	Organization of Programming Languages
Fall 2012	CMSC 498L	25	Cybersecurity lab (co-taught with 4 other profs)
Spring 2012	CMSC 498B	7	Secure Maryland (penetration testing)
Fall 2011	CMSC 631	13	Program Analysis and Understanding
Spring 2011	CMSC 838G	5	Software Security
Fall 2010	CMSC 433	40	Programming Language Technologies and Paradigms
Spring 2010	CMSC 330	66	Organization of Programming Languages
Fall 2009	CMSC 631	9	Program Analysis and Understanding
Fall 2007	CMSC 631	16	Program Analysis and Understanding
Spring 2007	CMSC 412	42	Operating Systems
Fall 2006	CMSC 631	20	Program Analysis and Understanding
Spring 2006	CMSC 433	44	Programming Language Technologies and Paradigms
Fall 2005	CMSC 412	50	Operating Systems
Spring 2005	CMSC 838Z	7	Language-based Security
Fall 2004	CMSC 412	45	Operating Systems
Spring 2004	CMSC 838Z	8	Tools and Techniques for Software Dependability
Fall 2003	CMSC 433	48	Programming Language Technologies and Paradigms
Spring 2003	CMSC 838Y	10	Agile and Adaptive Programming
Fall 2002	CMSC 433	47	Programming Language Technologies and Paradigms

2. Independent Study

Semester	Course	# Students	Title
Spring 2014	CMSC 498A	1	Research on authenticated data structures
Fall 2011	CMSC 498A	1	Research on dynamic software updating
Fall 2011	CMSC 498A	1	Program Analysis and Understanding
Fall 2005	CMSC 598A	1	Eclipse IDE support for Cyclone
Spring 2004	CMSC 390	1	Module system support for Cyclone
Spring 2004	CMSC 838A	1	Building operating systems with Cyclone
Spring 2003	CMSC 390	1	Supporting asynchronous methods in Java
Spring 2003	CMSC 390	1	Supporting dynamic software updating in Java

b. Course or Curriculum Development

Courses

Software Security - I developed a massively-open on-line course (MOOC) under the UMD-Coursera agreement, on this topic, as part of a specialization on cybersecurity. I fully developed the content

for this course, including videos, lecture notes, and projects. See <https://www.coursera.org/course/softwaresec>.

CMSC 330 - Organization of Programming Languages (2013)

Reintroduced logic programming to the course, and redesigned projects and lecture material to make better connections between course elements, including this new unit.

CMSC 498L - Cybersecurity Lab (2012)

This new course considered various aspects of cybersecurity, from cryptography, to application security, to network analysis, to hardware security, presented from a hands-on perspective. I organized the creation of this course and recruited four other professors to teach it with me; my part was focused on web application security, and I developed two labs for this topic.

CMSC 498B - Secure Maryland (2012)

This is a new course on penetration testing. Seven students piloted the course. We train them in scanning, vulnerability assessment, and exploitation, and educate them in the ethics of penetration testing. The most novel feature of the course is that they are permitted to attack the University's own networks in a supervised setting via public-facing interfaces. When vulnerabilities are discovered, they contact the system owners to work to fix the problems. In The 2012 iteration of the course revealed literally hundreds of vulnerabilities, the majority of which were fixed before the semester concluded.

CMSC 412 - Operating Systems (2004,2005,2007)

I substantially revised the content of the course—I created new lecture notes, new projects, and shifted focus more toward core operating systems and systems-building concepts (and away from networking and distributed systems, taught in other courses). I added an introductory project that uses Cyclone, a research programming language, and extended the remaining projects so that Cyclone could be used optionally; several students used Cyclone for their projects and two started independent study projects involving Cyclone with me afterward.

CMSC 433 - Programming Language Technologies and Paradigms (2002,2003,2006,2010,2013,2014)

I substantially reworked the course content twice. In 2006, I focused the material on advanced programming techniques for sequential, concurrent, and distributed software. I initially revised most of the lecture notes, and selectively created new lecture notes and new projects with each offering of the course. In 2010, I focused the material almost entirely on concurrent/parallel programming, with some material on distributed programming. I completely redesigned all projects and lecture notes.

CMSC 631 - Program Analysis and Understanding (2006,2007,2009,2011,2013)

This course focuses on programming language semantics, and foundations and implementation techniques of program analyses. Some of the material of the course was inherited from my earlier course, *CMSC 838Z* (see below). For the first offering I added some new lecture material, and for the second I reorganized the material to be more bottom-up. In both cases I designed new projects and homework assignments. For the most recent, I developed new material involving the use of a *proof assistant* to ensure that students' formalizations and proofs are checked for correctness automatically.

CMSC 838Y - Agile and Adaptive Programming (2003)

This course focuses on novel techniques for building reusable, reconfigurable, and adaptable software components and systems. I developed the lecture material and helped students design research projects.

CMSC 838Z - Tools and Techniques for Software Dependability (2004), Language-based Security (2005)

The theme of these courses is programming language-oriented techniques for building reliable and secure software. I developed the course content, lectures, and several substantial projects. Most of the project materials and some of the lecture materials from the 2004 offering were subsequently included in *CMSC 631, Program Analysis and Understanding*.

Tutorials

Using LLVM for Program Analysis and Transformation (co-presented with Andrew Ruef): This tutorial was given at the Conference on Programming Language Design and Implementation (PLDI) in June 2013, having 47 attendees. The response was so positive that we were invited to give it again in 2014. It was organized to give programming language researchers and implementers with information about how to use the Low Level Virtual Machine (LLVM) project to approach program transformation and analysis. Cf. <http://pldi2013.ucombinator.org/tutorials.html>

Using Symbolic Execution to Find Software Vulnerabilities: This tutorial was given at the Second Maryland Cybersecurity Center Symposium in May 2013, having 25 attendees. It presented an overview of symbolic execution technology and a tutorial on how to use the KLEE tool in particular. Cf. <http://www.cs.umd.edu/~mwh/se-tutorial/>

Curriculum

Member, Committee to develop curriculum for *Advanced Cybersecurity Experience for Students (ACES)* honors program, to begin Fall 2013.

Member, College committee to develop Oral communications class in response to new University requirement, August 2010—October 2011.

Member, Undergraduate Curriculum Evaluation committee, Spring 2007–2008.

Chair, Education sub-committee to examine the upper-level undergraduate curriculum for computer science, with a focus on broadening participation; University of Maryland, College Park, Fall 2005–Spring 2006.

Member, Education sub-committee to revise the upper-level undergraduate course requirements for computer science; University of Maryland, College Park, Fall 2003–Spring 2004.

Member, Education sub-committee to revise the introductory undergraduate curriculum for computer science, University of Maryland, College Park, Fall 2002.

d. Teaching Awards and Other Special Recognition

Department of Computer Science Faculty Teaching Award, Spring 2014.

Department of Computer Science Faculty Teaching Award, Spring 2007.

Department of Computer Science Faculty Teaching Award, Spring 2005.

Department of Computer Science Faculty Teaching Award, Spring 2003.

e. Advising: Other Than Research Direction

1. Masters Dissertation Committee member

- Surupa Biswas, *Memory Overflow Protection for Embedded Systems using Run-time Checks, Reuse and Compression*, Department of Electrical and Computer Engineering. Defended August 2004, graduated August 2004.
- (Co-chair) Jong hoon (David) An, *Dynamic Inference of Static Types for Ruby*, defended July 2010, graduated August 2010.

2. Doctoral Proposal Committee member (not including my own students)

- Jeff Stuckman, *Developing and Validating Security Vulnerability Indicator Metrics*. Proposed May 2013.
- Aaron Schulman, *Add Wireless Broadcast to the Internet*. Proposed May 2012.
- Peter Fontana, *Towards a Unified Theory of Timed Automata*. Proposed April 2012.
- Matthew Simpson, *Optimizing Memory Safety Checks Using Dynamic Pointer Disambiguation*. Proposed August 2008.
- Boniface Hicks (Pennsylvania State University), *Bridging the Semantic Gap in Security Policy*. Proposed December 2005.
- Yu David Liu (Johns Hopkins University), *Assemblages: Modules for Large-scale Distributed Software*. Proposed May 2004.

3. Doctoral Dissertation Committee member

- (Chair) Piotr Mardziel, *Modeling, Quantifying, and Limiting Adversary Knowledge*. Defended December 2014, graduated January 2015.
- Peter Fontana, *Towards a Unified Theory of Timed Automata*. Defended February 2014.
- (Opponent) Arnar Birgisson (Chalmers University of Technology, Sweden), *Tracking Dependencies for Security and Privacy*. Defended November 2013. The opponent is a special role in Swedish defenses: he begins the defense by presenting context on the research area, and then engages the candidate with roughly one hour of probing questions following the candidate's talk.
- (Co-Chair) Khoo Yit Phang, *User-Centered Program Analysis Tools*, defended June 2013, graduated August 2013.
- Aaron Schulman, *Add Wireless Broadcast to the Internet*. Defended August 2012.
- (Co-chair) Christopher M. Hayden, *Clear, Correct, and Efficient Dynamic Software Updates*, defended April 2012, graduated August 2012. This thesis was one of two winners of the first Larry S. Davis Doctoral Dissertation award; nominated for the ACM Doctoral Dissertation Award and ACM SIGPLAN Doctoral Dissertation Award.
- (Chair) Justin N. McCann, *Automating Performance Diagnosis in Networked Systems*, defended April 2012, graduated August 2012.
- (Co-chair) Kin Keung (Martin) Ma, *Improving Program Testing and Understanding via Symbolic Execution*, defended December 2011, graduated December 2011.
- Adam Bender, *An Accountability Architecture for the Internet*, defended November 2010, graduated December 2010.

- Matthew Simpson, *Runtime Enforcement of Memory Safety for the C Programming Language*, defended November 2010, graduated May 2011.
- Nathaniel Ayewah. *Static Analysis in Practice*. Defended July 2010, graduated August 2010.
- (Co-chair) Saurabh Srivastava, *Satisfiability-based Program Reasoning and Synthesis*. Defended April 2010, graduated May 2010.
- Suriya Subramanian (University of Texas at Austin), *Dynamic Software Updates: a VM-centric Approach*. Defended April 2010, graduated May 2010.
- Michael Furr. *Combining Static and Dynamic Typing in Ruby*. Defended October 2009, graduated December 2009.
- (Chair) Pavlos Papageorgiou (ECE), *The Measurement Manager: Modular and Efficient End-to-end Measurement Services*. Defended November 2008. Graduated December 2008.
- (Chair) Iulian Neamtiu, *Practical Dynamic Software Updating*. Defended July 2008. Graduated August 2008.
- (Co-Chair) Polyvios Pratikakis, *Sound, Precise and Efficient Static Race Detection for Multi-Threaded Programs*. Defended July 2008. Graduated August 2008.
- (Chair) Nikhil Swamy, *Language-Based Enforcement of User-Defined Security Policies as Applied to Multi-tier Web Programs*. Defended July 2008. Graduated August 2008. Thesis nominated for ACM SIGPLAN Doctoral Dissertation Award, 2009.
- (Co-Chair) Nick Louis Petroni, Jr, *Property-based integrity monitoring of operating system kernels*. Defended December 2007. Graduated Spring 2008. Thesis one of two nominated by the University of Maryland for the ACM Doctoral Dissertation Award.
- Yu David Liu (Johns Hopkins University), *Interaction-Oriented Programming*. Defended September 2007. Graduated December 2007.
- Boniface Hicks (Pennsylvania State University), *Bridging the Semantic Gap in Security Policy*. Defended August 2007. Graduated December 2007.
- Chadd Williams. *Using Historical Data from Source Code Revision Histories to Detect Source Code Properties*. Defended July 2006, graduated August 2006.
- Jeremy Manson. *The Java Memory Model*. Defended September 2004, graduated Winter 2004.
- Rajiv Gandhi, *Broadcast Scheduling*. Defended May 2003, graduated August 2003.

f. Advising: Research Direction

1. High school student research mentor

- Arun Dillipan (Poolesville High School), Summer 2014.
- Edward (Ted) Smith (Walt Whitman High School), Summer 2008.
- William C. Burton (Montgomery Blair High School), Summer 2007.
- Richard Matthew McCutchen (Montgomery Blair High School), Fall 2005.

2. Undergraduate research mentor

- Kyle Headley, June 2014–present (co-mentored with Jeff Foster).
- Ilse Haim, January 2014–present.
- Andrew Ruef, June 2010–2013 (he is now my Ph.D. student).

- James Parker, Spring 2012–2013.
- Edward (Ted) Smith, Summer 2009–2013 (co-mentored with Jeff Foster).
- Ryan Sims, November 2010–December 2011.
- Michail Denchev, June 2010–December 2011.
- QUEST-program mentor, Fall 2010, for a team of five undergraduates: Mitchell Kochman, Sebastian Gomez, Melinda Jih, Brad Klein, Jeremy Prince.
- Andrew Ferguson, Winter 2010–Summer 2010.
- QUEST-program mentor, Fall 2009, for a team of six undergraduates: Jeremy Erdman, Andrew Duch, Cameron Rose, Nicole Thomas, Haren Arcot, and Akshay Goyal.
- Elliott Sprehn, December 2008–May 2008.
- Jeffrey S. Meister, Summer 2007–Summer 2008 (co-mentored with Jeff Foster).
- Neha Gupta, Spring 2005.
- Hariharan Sivaramakrishnan (ECE), Spring 2004–Spring 2005.
- Christopher Wamble, Summer 2003. This was as part of the Ronald E. McNair Post-Baccalaureate Achievement Program, which aims to increase the participation of under-represented groups in graduate school.
- Martha Gebremichael, Spring 2003.
- Michael Nelson, Spring 2003.

3. Masters student research advisor

- James Parker, September 2013–December 2014. Graduated December 2014.
- Elnatan Reisner (co-advised with Jeff Foster). Graduated August 2011.
- Jonathan Turpie (co-advised with Jeff Foster). Graduated August 2011.
- David Jong-Hoon An (co-advised with Jeff Foster). Graduated August 2010.
- Adithya Nagarajan (SE). Graduated Spring 2003.
- James Rose. Graduated Spring 2004.

4. Doctoral student research advisor

- Andrew Ruef, Spring 2014–present.
- Luis Pina, Fall 2012–present (co-advised; Luis is a Ph.D. student at Instituto Superior Tecnico, Lisbon, Portugal, visiting my group).
- Aseem Rastogi, Fall 2012–present.
- Karla Saur, Spring 2012–present. Advanced to candidacy May 2014. Co-advised with Jeff Foster.
- Chang Liu, Fall 2012–present. Co-advised with Elaine Shi.
- (graduated) Piotr Mardziel, Summer 2009–January 2015. Advanced to candidacy March 2012. Graduated January 2015.
- (graduated) Khoo Yit Phang, Spring 2007–Summer 2013. Advanced to candidacy May 2010. Graduated August 2013. Co-advised with Jeff Foster.
- (graduated) Justin McCann, Spring 2010–May 2012. Advanced to candidacy December 2010. Graduated August 2012.

- (graduated) Chris Hayden, Spring 2007–May 2012. Advanced to candidacy May 2010. Graduated August 2012. Co-advised with Jeff Foster.
- (graduated) Kin Keung (Martin) Ma, Spring 2009–December 2011. Graduated December 2011. Co-advised with Jeff Foster.
- (graduated) Saurabh Srivastava, Spring 2005–2010. Advanced to candidacy Winter 2008. Graduated Spring 2010. Co-advised with Jeff Foster.
- (graduated) Pavlos Papageorgiou (ECE), Summer 2004–Fall 2008. Advanced to candidacy May 2007. Graduated Fall 2008.
- (graduated) Iulian Neamtiu, Fall 2003–present. Received Masters degree Spring 2005. Advanced to candidacy Fall 2006. Defended July 2008. Graduated Summer 2008.
- (graduated) Polyvios Pratikakis, Summer 2003–Summer 2008. Received Masters Degree Spring 2005. Advanced to candidacy Fall 2007. Defended July 2008. Graduated Summer 2008. Co-advised with Jeff Foster.
- (graduated) Nikhil Swamy, Spring 2004–Summer 2008. Received Masters degree Fall 2005. Advanced to candidacy Spring 2007. Defended July 2008. Graduated Summer 2008.
- (graduated) Nick L. Petroni, Jr., Spring 2006–December 2007. Received Masters degree Spring 2003. Advanced to candidacy Spring 2006. Graduated Spring 2008. Co-advised with Bill Arbaugh.
- (former) Jonathan Turpie, Winter 2010–August 2011.
- (former) Brian Corcoran, Summer 2007–Summer 2009.
- (former) Eric Hardisty, Winter 2008–Summer 2009. Co-advised with Jeff Foster.
- (former) Gary Jackson, Fall 2005–Fall 2007. Co-advised with Jeff Foster.
- (former) Mujtaba Ali, Spring 2004–Spring 2005.

5. Post-doctoral research advisor

- Piotr Mardziel, January 2015–present.
- Matthew Hammer, September 2012–present.
- Nataliya Guts, January 2011–July 2012.
- Stephen Magill, January 2010–August 2011. Currently a researcher at Galois, in Portland, OR.
- Manuel Oriol, Spring 2004–Spring 2005. Currently a senior lecturer at the University of York, United Kingdom.

4 Service

a. Professional

1. Blog: <http://www.pl-enthusiast.net> : *The PL Enthusiast*; co-organized with Swarat Chauhuri (Rice), since June 2014.
2. Unpaid Reviewing Activities for Agencies
 - NSF SaTC panelist, Fall 2013.
 - NSF SaTC panelist, Spring 2012.

- NSF CCF panelist, Spring 2010.
- NSF CCF panelist, Spring 2008.
- NSF CNS panelist, Spring 2006.
- NSF CCF panelist, Spring 2005.
- NSF CCF panelist, Spring 2004.
- NSF SBIR panelist, Fall 2003.
- NSF SBIR panelist, Fall 2002.
- External reviewer, NSF CCR INT panel, Fall 2002.

3. Other non-University Panels

- Program Co-Chair, Computer Security Foundations Symposium (CSF), June 2014.
- Program Committee (PC) member, IEEE Symposium on Security and Privacy, May 2015.
- Co-organizer, Dagstuhl Seminar on *the synergy between programming languages and cryptography*, November 30–Dec 5, 2014.
- Selection Committee, 2014 IBM "PL Day", October 2014.
- PC Member, ACM SIGPLAN Conference on Object-Oriented Programming, Languages, and Applications (OOPSLA), October 2014.
- PC member, Computer Security Foundations Symposium (CSF), June 2014.
- PC member, ACM SIGPLAN "Off the Beaten Track" Workshop, January 2014.
- Steering Committee, ACM SIGPLAN-SIGACT Symposium on the Principles of Programming Languages (POPL), January 2011–2014.
- Invited participant, National Academy of Sciences Kavli Frontiers of Science Symposium (one of five computer scientists invited, out of more than 100 participants), November 2013.
- Co-organizer, ISAT workshop on Fostering Programming Language Adoption, February 2013.
- PC member, ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), June 2013.
- External Review Committee (ERC), ACM SIGPLAN-SIGACT Symposium on the Principles of Programming Languages (POPL), January 2013.
- Invited participant, National Science Foundation (NSF) Workshop on *Formal Methods: Future Directions and Transitions to Practice*, San Diego, CA, December 5–6, 2012.
- Member, CyberMaryland 2.0 Task Force, April–October 2012. Panel commissioned out of the Maryland Department of Business and Economic Development (DBED) to examine progress toward goals for increasing Maryland's impact and standing in the cybersecurity arena.
- PC Member, International Workshop on Hot Topics in Software Upgrades, June 2012.
- Program Chair, ACM SIGPLAN-SIGACT Symposium on the Principles of Programming Languages (POPL), January 2012. PC Chair precluded from submitting to the conference.
- Organizer, Principles of Programming Languages Mini-Workshop, held at the University of Maryland, October 2, 2011.
- PC Member, ACM SIGPLAN Conference on Object-Oriented Programming, Languages, and Applications (OOPSLA), October 2011.

- Co-organizer (PC Chair and General Chair), International Workshop on Hot Topics in Software Upgrades, April 2011.
- PC Member, ACM SIGPLAN Workshop on Types in Language Design and Implementation (TLDI), January 2011.
- Local arrangements chair, ACM SIGPLAN International Conference on Functional Programming (ICFP), September 2010.
- Co-organizer, ISAT study on Software Synthesis, August 2010–2011
- PC Member, ACM SIGPLAN International Conference on Functional Programming (ICFP), August 2010.
- ERC Member, ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), June 2010.
- Tutorials Chair, ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), June 2010.
- PC Member, ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering (PASTE), June 2010.
- Participant, National Science Foundation, Computer Systems Research Future Directions Workshop, March 2010.
- Invited participant, NSF workshop on *Virtual Execution Environments for Scientific Computing*, September 2010.
- Invited participant, NSF workshop on *Trustworthy Computing*, November 2010.
- PC Member, European Symposium on Programming (ESOP), March 2010.
- Member of DARPA Information Science and Technology Board (ISAT), September 2009–2012.
This panel consists of 30 individuals (largely faculty and researchers) from U.S. institutions which advises DARPA by conducting studies on topics of increasing relevance and impact.
- PC Member, IEEE Symposium on Security and Privacy (Oakland), May 2009.
- Steering Committee, 1st Workshop on Theory and Practice of Provenance (TaPP), February 2009.
- PC Member, ACM SIGPLAN-SIGACT Symposium on the Principles of Programming Languages (POPL), January 2009. PC members were held to a higher standard for paper acceptance.
- PC Member, ACM SIGSAC Conference on Computer and Communications Security (CCS), 2008.
- PC Member, First Workshop on Compiler and Architectural Techniques for Application Reliability and Security (CATARS), 2008 (in conjunction with the International Conference on Dependable Systems and Networks (DSN 2008))
- RC member, International Symposium on Memory Management (ISMM), 2008.
- PC Member, International Conference on Coordination Models and Languages (COORDINATION), 2008.
- Member of DARPA Information Processing Technology Office (IPTO) Futures Panel, Fall 2007.
This panel consists of a few select faculty gathered to help IPTO chart its research goals and to develop potential programs.
- Member of DARPA Computer Science Study Panel (CS2P), 2006–2009. This panel consists of a dozen select junior faculty from U.S. institutions, convening four times during 2006, and twice during 2007.

- Co-organizer, the Mid-Atlantic Programming Languages Seminar (MAPLS), held in conjunction with the New Jersey Programming Languages Seminar (NJPLS) at the University of Maryland, November 2007.
- PC Member, ACM SIGPLAN Conference on Object-Oriented Programming Languages, Systems, and Applications (OOPSLA), 2007. PC members were held to a higher standard for paper acceptance.
- PC Member, International Conference on Coordination Models and Languages (COORDINATION), 2007.
- PC Member, ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), 2007. PC members were precluded from submitting to the conference.
- General Chair and Technical Program Chair, Workshop on Programming Languages and Analysis for Security (PLAS), 2007.
- PC Member, Workshop on Programming Languages and Analysis for Security (PLAS), 2006.
- PC Member, Workshop on Formal Techniques for Java-like Languages (FTfJP), 2006.
- PC Member, ACM SIGPLAN Workshop on Semantics, Program Analysis, and Computing Environments for Memory Management (SPACE), 2006.
- Co-organizer, the Mid-Atlantic Programming Languages Seminar (MAPLS), held in conjunction with the New Jersey Programming Languages Seminar (NJPLS) at the University of Maryland, November 2005.
- PC Member, ACM Symposium on Applied Computing (SAC), Object-oriented Programming Systems (OOPS) track, 2005.
- PC Member, ACM Workshop on Synchronization and Concurrency in Object-Oriented Languages (SCOOL) 2005.
- PC Member, ACM/USENIX Virtual Execution Environments Conference (VEE) 2005.
- PC Member, Workshop on Formal aspects of Unanticipated Software Evolution (FUSE) 2004.
- PC Member, International Conference on Parallel Processing (ICPP) 2004.
- PC Member, International Working Conference on Active Networking (IWAN) 2004.
- PC Member, International Working Conference on Active Networking (IWAN) 2003.
- PC Member, Workshop on Unanticipated Software Evolution (USE) 2003.
- PC Member, International Working Conference on Active Networking (IWAN) 2002.
- PC Member, Workshop on Unanticipated Software Evolution (USE) 2002.
- PC Member, International Working Conference on Active Networking (IWAN) 2001.

b. Campus

Department

- Chair, Strategic planning subcommittee on undergraduate education, 2014–2015.
- Co-Director (with Adam Porter), CS Education for Tomorrow, 2014–2015.
- Member, Department Council, 2014–2015.
- Chair, Undergraduate Scholarships and awards committee, 2014–2015.
- Member, Admissions Committee, 2014
- Member, Teaching Evaluation Committee, 2011–2013.

- Member, Tenure Committee for Assistant Professor Nick Feamster, Fall 2012.
- Member, Faculty hiring committee for positions in cybersecurity and software systems, Fall 2011–Spring 2012.
- Member, Teaching Evaluation Committee, 2011–2012.
- Member, Faculty hiring committee for director of Maryland Cybersecurity Center, Fall 2010–Spring 2011.
- Member, Faculty hiring committee for positions in cybersecurity, Fall 2010–Spring 2011.
- Member, Honors Program committee, Fall 2010–Spring 2011.
- Member, Teaching Evaluation Committee, 2010–2011.
- Elected Member, Department Council, 2010–2011. 6 professors are elected annually by the rest of the faculty to sit on this committee.
- Member, Honors Program committee, Fall 2009–Spring 2010.
- Member, Friday Faculty Lunch committee, Fall 2009–Spring 2010.
- Member, Teaching Evaluation Committee, 2009–2010.
- Elected Member, Department Council, 2009–2010.
- Member, Graduate Admissions committee, Spring 2008.
- Faculty speaker, undergraduate recruiting day, March 13, 2008.
- Elected Member, Department Council, 2007–2008.
- Chair, Graduate Student Evaluation committee, 2008.
- Member, Faculty panel to advise undergraduates interested in pursuing graduate school, November 8, 2007.
- Member, Undergraduate Curriculum Evaluation committee, Spring 2007–present.
- Chair, Graduate Student Evaluation committee, 2007.
- Chair, Faculty Retreat Committee on Undergraduate Education, Fall 2005–Spring 2006.
- Chair, Graduate Student Evaluation committee, 2006.
- Member, Graduate Admissions committee, Spring 2005.
- Chair, Graduate Student Evaluation committee, 2005.
- Member, Faculty Retreat Committee on Graduate Student Recruiting, Admissions, and Retention, Fall 2003–Fall 2004.
- Member, Friday Faculty Lunch committee, Fall 2003–Spring 2004.
- Faculty judge, High-school programming contest, Spring 2003.
- Member, Graduate Admissions committee, Spring 2003.
- Member, Lab committee, Fall 2002–Spring 2003.
- Member, Education committee, Fall 2002–present.
- Member, Programming Languages, Software Engineering, and Human-Computer Interaction Field Committee, Fall 2002–present.

College

- Member, Hiring committee for the Maryland Cybersecurity Center Director (hosted by UMI-ACS), January–October 2013.

- Member, Committee to develop Oral communications class in response to new University requirement, August 2010—October 2010.
- Member, UMIACS APT Committee, 2009–2010.
- Member, UMIACS APT Committee, 2006–2007.
- Member, Business Office Coordinator search committee, July 2005.

University

- Chair, Security Enhancements Subcommittee, President’s Task Force on Cybersecurity, April–June 2014
- Chair, Infrastructure Focus Area, IT Strategic Planning task force. One of four Chairs; output was a document presenting the University’s strategic plan for IT, 2013.
- Director, Maryland Cybersecurity Center (MC2) (a joint effort of the CMNS and Engineering colleges), Oct 2011–2013.
- Team mentor, QUEST program (jointly run by the Smith school, the Engineering school, and CMNS), Fall 2010.
- Team mentor, QUEST program, Fall 2009.
- Panelist for CAREER proposal preparation workshop, June 13, 2005.

University System

- Member, USM Chancellor’s Task Force on Cyber Security, November 2010–February 2011.

c. Public

- Committee Chair, Cub Scout Pack 1250, Laurel, MD, August 2012–present.
- Youth Soccer coach, Greater Laurel United Soccer Club, August 2013–present.
- Career day speaker, Eastern Middle School, May 2, 2014