DAFTAR PUSTAKA

[1] D. Bruening, Q. Zhao and S. Amarasinghe, "Transparent Dynamic Instrumentation," in *VEE '12 Proceedings of the 8th ACM SIGPLAN/SIGOPS conference on Virtual Execution Environments*, New York, 2012.

[2] M. Probst, "Dynamic Binary Translation," *CD Laboratory for Compilation Techniques*, 2003.

[3] N. Mavrogiannopoulos, N. Kisserli and B. Preneel, "A taxonomy of self-modifying code for obfuscation," *Computers and Security,* vol. 30, no. 8, pp. 679-691, 2011.

[4] N. Nethercote, "Dynamic Binary Analysis and Instrumentation," University of Cambridge Computer Laboratory, 2004.

[5] "The DynamoRIO API," Google, 8 October 2015. [Online]. Available: http://dynamorio.org/docs/.

[6] K. Anand, M. Smithson, K. Elwazeer, A. Kotha, J. Gruen, N. Giles and R. Barua, "A Compiler-level Intermediate Representation based Binary Analysis and Rewriting System," in *EuroSys '13 Proceedings of the 8th ACM European Conference on Computer Systems*, New York, 2013.

[7] "ELF," OS Development, 7 August 2015. [Online]. Available: http://wiki.osdev.org/ELF.

[8] X. Xie, F. Liu, B. Lu and F. Xiang, "Mixed Obfuscation of Overlapping Instruction and Self-Modify Code Based on Hyper-Chaotic Opaque Predicates," in *Computational Intelligence and Security (CIS)*, Kunming, 2014.

[9] S. Blazy, V. Laporte and D. Pichardie, “Verified Abstract Interpretation Techniques for Disassembling Low-level Self-modifying Code,” in *Interactive Theorem Proving*, Springer International Publishing, 2014, pp. 128-143.

[10] B. Anckaert, M. Madou and K. D. Bosschere, “A Model for Self-Modifying Code,” in *Information Hiding*, Springer Berlin Heidelberg, 2007, pp. 232-248.

[11] T. Touili, B. Cook and P. Jackson, “Computer Aided Verification,” in *22nd International Conference, CAV 2010*, Edinburgh.

[12] N. M. Hai, M. Ogawa and Q. T. Tho, Obfuscation code localization based on CFG generation of malware, Clermont-Ferrand: Springer, 2015.

[13] R. Mankadan, P. A. Kumar and R. Govindaraju, “CODE SECURITY BY CONFUSING LOGIC FLOW USING SELF MODIFYING CODE,” *International Journal of Innovative Trends and Emerging Technologies*, vol. I, no. 2, 2015.

[14] J. Pogulis, “Generation of dynamic control-dependence graphs for binary programs,” Linköping University, Department of Computer and Information Science, Database and information techniques, 2014.

[15] M. Payer, A. Barresi and R. T. Gross, “Lockdown: Dynamic Control-Flow Integrity,” ETH, Zürich, 2014.

[16] W. Arthur, B. Mehne, R. Das and T. Austin, “Getting in control of your control flow with control-data isolation,” in *CGO '15 Proceedings of the 13th Annual IEEE/ACM International Symposium on Code Generation and Optimization*, 2015.

[17] V. Pappas, M. Polychronakis and A. D. Keromytis, “Dynamic Reconstruction of Relocation Information for Stripped Binaries,” in *Research in Attacks, Intrusions and Defenses*, Springer International Publishing, 2014, pp. 68-87.

[18] T. Vilkeliskis, “Automated unpacking of executables using Dynamic Binary Instrumentation,” Department of Computer Science, Stevens Institute of Technology, 2009.

[19] . A. Roy, S. Hand and T. Harris, “Hybrid binary rewriting for memory access instrumentation,” in *VEE '11 Proceedings of the 7th ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*, New York, 2011.

[20] J. Ming, D. Xu, L. Wang and D. Wu, “LOOP: Logic-Oriented Opaque Predicate Detection in Obfuscated Binary Code,” in *CCS '15 Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015.

[21] H. Guo, J. Pang, Y. Zhang, F. Yue and R. Zhao, “HERO: A novel malware detection framework based on binary translation,” in *Intelligent Computing and Intelligent Systems (ICIS)*, Xiamen, 2010.

[22] H. Guan, E. Zhu, H. Wang, R. Ma, Y. Yang and B. Wang, “SINOF: A dynamic-static combined framework for dynamic binary translation,” *Journal of Systems Architecture,* vol. 58, no. 8, pp. 305-317, 2012.

[23] M. Braun, M. Buchwald and A. Zwinkau, “FIRM: A Graph-Based Intermediate Representation,” Karlsruhe Institute of Technology, 2011.