References

[1] Uhlig, R., Neiger, G., Rodgers, D., Santoni, A. L., Martins, F., Anderson, A. V., ... & Smith, L. (2005). Intel virtualization technology. Computer, 38(5), 48-56.

[2] McCabe, Thomas J. "A complexity measure." Software Engineering, IEEE Transactions on 4 (1976): 308-320.

[3] E. Berlinger, "An Information Theory Based Complexity Measure",Proceedings of the 1980 National Computer Conference. pp. 773-779.

[4] Atkinson, Colin, and Thomas Kühne. "Reducing accidental complexity in domain models." Software & Systems Modeling 7.3 (2008): 345-359.

[5] Cook, C. "Information theory metric for assembly language." Proceedings of Third Annual Oregon Workshop on Software Metrics. 1991.

[6] Shannon, Claude Elwood. "A mathematical theory of communication." ACM SIGMOBILE Mobile Computing and Communications Review 5.1 (2001): 3-55.

[7] Oh, N., Shirvani, P. P., & McCluskey, E. J. (2002). Control-flow checking by software signatures. Reliability, IEEE Transactions on, 51(1), 111-122.

[8] Bardas, A. G. (2010). Static code analysis. Journal of Information Systems & Operations Management, 4(2), 99-107.

[9] Hsi, C. H., Bresee, R. R., & Annis, P. A. (2000). Characterizing fuzz on fabrics using image analysis. Textile Research Journal, 70(10), 859-865.

[10] Eder, J., Kappel, G., & Schrefl, M. (1994). Coupling and cohesion in object-oriented systems. Technical Reprot, University of Klagenfurt, Austria.

[11] Shin, Y., & Williams, L. (2008, October). Is complexity really the enemy of software security?. In Proceedings of the 4th ACM workshop on Quality of protection (pp. 47-50). ACM.

[12] Intel® 64 and IA-32 Architectures Software Developer’s Manual Volume 3B: System Programming Guide, Part 2.

[13] Barth, Adam, Collin Jackson, and William Li. "Attacks on javascript mashup communication." Proceedings of the Web. Vol. 2. 2009.

[14] Goktas, E., Athanasopoulos, E., Bos, H., & Portokalidis, G. (2014, May). Out of control: Overcoming control-flow integrity. In Security and Privacy (SP), 2014 IEEE Symposium on (pp. 575-589). IEEE.

[15] Poon, Wing-Chi, and Aloysius K. Mok. "Improving the Latency of VMExit Forwarding in Recursive Virtualization for the x86 Architecture." System Science (HICSS), 2012 45th Hawaii International Conference on. IEEE, 2012.

[16] Sheldon, M., and Ganesh Venkitachalam Boris Weissman. "Retrace: Collecting execution trace with virtual machine deterministic replay." Proceedings of the Third Annual Workshop on Modeling, Benchmarking and Simulation (MoBS 2007). 2007.

[17] Dinaburg, A., Royal, P., Sharif, M., & Lee, W. (2008, October). Ether: malware analysis via hardware virtualization extensions. In Proceedings of the 15th ACM conference on Computer and communications security (pp. 51-62). ACM.

[18] Shannon, C. E. (1957). A universal Turing machine with two internal states. Automata studies, 34, 157-165.

[19] Nikhil, R. S. (1989, April). Can dataflow subsume von Neumann computing?. In ACM SIGARCH Computer Architecture News (Vol. 17, No. 3, pp. 262-272). ACM.

[20] Lee, D., Choi, Y., Jung, J., Kim, J., & Won, D. (2015). An efficient categorization of the instructions based on binary excutables for dynamic software birthmark. International Journal of Information and Education Technology, 5(8), 571.

[21] Dandamudi, Sivarama P. Introduction to assembly language programming: from 8086 to Pentium processors. Springer Science & Business Media, 2013.

[22] Boggs, D., Baktha, A., Hawkins, J., Marr, D. T., Miller, J. A., Roussel, P., ... & Venkatraman, K. S. (2004). The Microarchitecture of the Intel Pentium 4 Processor on 90nm Technology. Intel Technology Journal, 8(1).

[23] Russinoff, D. M. (2000, November). A Case Study in Formal Verification of Register-Transfer Logic with ACL2: The Floating Point Adder of the AMD Athlon TM Processor. In Formal Methods in Computer-Aided Design (pp. 22-55). Springer Berlin Heidelberg.

[24] Petroni Jr, N. L., & Hicks, M. (2007, October). Automated detection of persistent kernel control-flow attacks. In Proceedings of the 14th ACM conference on Computer and communications security (pp. 103-115). ACM.

[25] Bhandarkar, D., & Clark, D. W. (1991, April). Performance from architecture: comparing a RISC and a CISC with similar hardware organization. In ACM SIGARCH Computer Architecture News (Vol. 19, No. 2, pp. 310-319). ACM.

[26] Hennessy, J., Jouppi, N., Przybylski, S., Rowen, C., Gross, T., Baskett, F., & Gill, J. (1982, October). MIPS: A microprocessor architecture. In ACM SIGMICRO Newsletter (Vol. 13, No. 4, pp. 17-22). IEEE Press.

[27] Tanenbaum, A. S., & Bos, H. (2014). Modern operating systems. Prentice Hall Press.

[28] Schneider, F. B., Morrisett, G., & Harper, R. (2001). A language-based approach to security. In Informatics (pp. 86-101). Springer Berlin Heidelberg.

[29] Andronick, J., Greenaway, D., & Elphinstone, K. (2010, October). Towards Proving Security in the Presence of Large Untrusted Components. In SSV.

[30] Liedtke, J. (1995). On micro-kernel construction (Vol. 29, No. 5, pp. 237-250). ACM.

[31] Roch, B. (2004). Monolithic kernel vs. Microkernel. TU Wien.

[32] Abramson, D., Jackson, J., Muthrasanallur, S., Neiger, G., Regnier, G., Sankaran, R.,& Wiegert, J. (2006). Intel Virtualization Technology for Directed I/O. Intel technology journal, 10(3).

[33] Chiueh, T. C., Venkitachalam, G., & Pradhan, P. (1999). Integrating segmentation and paging protection for safe, efficient and transparent software extensions. ACM SIGOPS Operating Systems Review, 33(5), 140-153.

[34] van de Ven, A., Patel, B. V., Mallick, A. K., Neiger, G., Coke, J. S., Dixon, M. G., & Brandt, J. W. (2011). U.S. Patent Application No. 13/997,857.

[35] Arce, I. (2004). The shellcode generation. Security & Privacy, IEEE, 2(5), 72-76.

[36] Xiong, Haiquan, and Zhiyong Liu. "The Architectural Based Interception and Identification of System Call Instruction within VMM." (2013).

[37] Molnar, David, et al. "The program counter security model: Automatic detection and removal of control-flow side channel attacks." Information Security and Cryptology-ICISC 2005. Springer Berlin Heidelberg, 2005. 156-168.

[38] Peleg, A., & Weiser, U. (1996). MMX technology extension to the Intel architecture. Micro, IEEE, 16(4), 42-50.

[39] Katona, G. O., & Nemetz, T. O. (1976). Huffman codes and self-information. Information Theory, IEEE Transactions on, 22(3), 337-340.

[40] Crampton, J. (2005, June). A reference monitor for workflow systems with constrained task execution. In Proceedings of the tenth ACM symposium on Access control models and technologies (pp. 38-47). ACM.

[41] Smith, S. (2013). Trusted computing platforms: design and applications. Springer.

[42] Pfleeger, C. P., & Pfleeger, S. L. (2002). Security in computing. Prentice Hall Professional Technical Reference.

[43] Lin, T. Y. (1989, December). Chinese wall security policy-an aggressive model. In Computer Security Applications Conference, 1989., Fifth Annual (pp. 282-289). IEEE.

[44] Elliott Bell, D. (2011). Bell–La Padula Model. Encyclopedia of Cryptography and Security, 74-79.

[45] Ge, X., Polack, F., & Laleau, R. (2004, June). Secure databases: an analysis of Clark-Wilson model in a database environment. In Advanced Information Systems Engineering (pp. 234-247). Springer Berlin Heidelberg.

[46] Abadi, M., Budiu, M., Erlingsson, U., & Ligatti, J. (2005, November). Control-flow integrity. In Proceedings of the 12th ACM conference on Computer and communications security (pp. 340-353). ACM.

[47] Lombardi, F., & Di Pietro, R. (2011). Secure virtualization for cloud computing. Journal of Network and Computer Applications, 34(4), 1113-1122.

[48] Tsifountidis, Fotis. "Virtualization Security: Virtual Machine Monitoring and Introspection." Signature (2010).

[49 Smirnov, Alexey, and Tzi-cker Chiueh. "DIRA: Automatic Detection, Identification and Repair of Control-Hijacking Attacks." NDSS. 2005.

[50] Zhang, Chao, et al. "Practical control flow integrity and randomization for binary executables." Security and Privacy (SP), 2013 IEEE Symposium on. IEEE, 2013.

[51] Griswold, William G., et al. "Modular Software Design with Crosscutting Interfaces."

[52] Tsoutsos, N. G., & Maniatakos, M. (2014). Fabrication attacks: Zero-overhead malicious modifications enabling modern microprocessor privilege escalation. Emerging Topics in Computing, IEEE Transactions on, 2(1), 81-93.

[53] Davi, Lucas, et al. "Poster: control-flow integrity for smartphones." Proceedings of the 18th ACM conference on Computer and communications security. ACM, 2011.

[54] Davi, Lucas, et al. "MoCFI: A Framework to Mitigate Control-Flow Attacks on Smartphones." NDSS. 2012.

[55] Bermudo, Nerina, Andreas Krall, and Nigel Horspool. "Control flow graph reconstruction for assembly language programs with delayed instructions." Source Code Analysis and Manipulation, 2005. Fifth IEEE International Workshop on. IEEE, 2005.

[56] Vigna, Giovanni. "Static disassembly and code analysis." Malware Detection. Springer US, 2007. 19-41.

[57] Abadi, M., Budiu, M., Erlingsson, U., & Ligatti, J. (2005, November). Control-flow integrity. In Proceedings of the 12th ACM conference on Computer and communications security (pp. 340-353). ACM.

[58] Tsifountidis, Fotis. "Virtualization Security: Virtual Machine Monitoring and Introspection." Signature (2010).