Reference List for "Keeping an Eye on Virtualization: VM Monitoring Techniques Applications"

Zak Estrada, Gary Wang, Dr. Zbigniew Kalbarczyk

Key Concepts

[1] A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori, "kvm: the linux virtual machine monitor," in *In Proc. of the Linux Symposium*, vol. 1, 2007, pp. 225–230.

[2] Intel Corporation, *Intel® 64 and IA-32 Architectures Software Developers Manual Volume 3 (3A, 3B & 3C): System Programming Guide*, September 2014.

[3] Advanced Micro Devices Inc, *AMD64 Architecture Programmers Manual Volume 2: System Programming*, May 2013.

[4] R. Uhlig, G. Neiger, D. Rodgers, A. L. Santoni, F. C. Martins, A. V. Anderson, S. M. Bennett, A. Kagi, F. H. Leung, and L. Smith, "Intel virtualization technology," *Computer*, vol. 38, no. 5, pp. 48–56, 2005.

[5] N. Bhatia, "Performance evaluation of intel ept hardware assist," *VMware, Inc*, 2009.

[6] R. P. Goldberg, "Survey of virtual machine research," *Computer*, vol. 7, no. 6, pp. 34–45, 1974.

[7] G. J. Popek and R. P. Goldberg, "Formal requirements for virtualizable third generation architectures," pp. 121–, 1973.

[8] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, pp. 164–177, 2003.

[9] M. Rosenblum and T. Garfinkel, "Virtual machine monitors: Current technology and future trends," *Computer*, vol. 38, no. 5, pp. 39–47, 2005.

[10] M. Rosenblum, "Vmwares virtual platform," in *Proceedings of hot chips*, vol. 1999, 1999, pp. 185–196.

[11] S. Soltesz, H. Pötzl, M. E. Fiuczynski, A. Bavier, and L. Peterson, "Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors," in *ACM SIGOPS Operating Systems Review*, vol. 41, no. 3. ACM, 2007, pp. 275–287.

[12] M. Kerrisk, "Namespaces in operation, part 1: namespaces overview," Online, http://lwn.net/Articles/531114/, 2013.

[13] P. Menage, "CGROUPS," Online, https://www.kernel.org/doc/Documentation/cgroups/cgroups.txt, 2006.

[14] S. Hykers, "What is docker?" Online, https://www.docker.com/whatisdocker/, 2013.

[15] Z. J. Estrada, F. Deng, Z. Stephens, C. Pham, Z. Kalbarczyk, and R. Iyer, "Performance comparison and tuning of virtual machines for sequence alignment software," *Scalable Computing: Practice and Experience*, vol. 16, no. 1, 2015.

[16] Z. J. Estrada, Z. Stephens, C. Pham, Z. Kalbarczyk, and R. K. Iyer, "A performance evaluation of sequence alignment software in virtualized environments," in *Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on*. IEEE, 2014, pp. 730–737.

[17] A. Madhavapeddy, R. Mortier, C. Rotsos, D. Scott, B. Singh, T. Gazagnaire, S. Smith, S. Hand, and J. Crowcroft, "Unikernels: Library operating systems for the cloud," in *ACM SIGPLAN Notices*, vol. 48, no. 4. ACM, 2013, pp. 461–472.

[18] D. P. Bovet and M. Cesati, *Understanding the Linux kernel*. " O'Reilly Media, Inc.", 2005.

Validation

[19] C. Pham, D. Chen, Z. Kalbarczyk, and R. K. Iyer, "Cloudval: A framework for validation of virtualization environment in cloud infrastructure," in *Dependable Systems & Networks (DSN), 2011 IEEE/IFIP 41st International Conference on*. IEEE, 2011, pp. 189–196.

[20] D. T. Stott, B. Floering, D. Burke, Z. Kalbarczpk, and R. K. Iyer, "Nftape: a framework for assessing dependability in distributed systems with lightweight fault injectors," in *Computer Performance and Dependability Symposium, 2000. IPDS 2000. Proceedings. IEEE International*. IEEE, 2000, pp. 91–100.

[21] M. Le, A. Gallagher, and Y. Tamir, "challenges and opportunities with fault injection in virtualized systems," in *1st Int. Workshop on Virtualization Performance: Analysis, Characterization, and Tools*, 2008.

[22] Z. Mwaikambo, A. Raj, R. Russell, J. Schopp, and S. Vaddagiri, "Linux kernel hotplug cpu support," in *Linux Symposium*, vol. 2, 2004.

[23] T. K. Tsai, M.-C. Hsueh, H. Zhao, Z. Kalbarczyk, and R. K. Iyer, "Stress-based and path-based fault injection," *Computers, IEEE Transactions on*, vol. 48, no. 11, pp. 1183–1201, 1999.

[24] A. Stern, "Update from amazon regarding fridays s3 downtime," *Febuary*, vol. 16, p. 2008, 2008.

[25] M.-C. Hsueh, T. K. Tsai, and R. K. Iyer, "Fault injection techniques and tools," *Computer*, vol. 30, no. 4, pp. 75–82, 1997.

[26] D. P. Siewiorek, R. Chillarege, and Z. T. Kalbarczyk, "Reflections on industry trends and experimental research in dependability," *Dependable and Secure Computing, IEEE Transactions on*, vol. 1, no. 2, pp. 109–127, 2004.

[27] W. Gu, Z. Kalbarczyk, R. K. Iyer, and Z. Yang, "Characterization of linux kernel behavior under errors," in *2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE Computer Society, 2003, pp. 459–459.

[28] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds," in *Proceedings of the 16th ACM conference on Computer and communications security*. ACM, 2009, pp. 199–212.

[29] S. K. Sahoo, J. Criswell, C. Geigle, and V. Adve, "Using likely invariants for automated software fault localization," *ACM SIGARCH Computer Architecture News*, vol. 41, no. 1, pp. 139–152, 2013.

[30] R. Abreu, P. Zoeteweij, and A. J. Van Gemund, "On the accuracy of spectrum-based fault localization," in *Testing: Academic and Industrial Conference Practice and Research Techniques-MUTATION, 2007. TAICPART-MUTATION 2007.* IEEE, 2007, pp. 89–98.

[31] C. Liu, L. Fei, X. Yan, J. Han, and S. P. Midkiff, "Statistical debugging: A hypothesis testing-based approach," *Software Engineering, IEEE Transactions on*, vol. 32, no. 10, pp. 831–848, 2006.

[32] V. Dallmeier, C. Lindig, and A. Zeller, "Lightweight defect localization for java," in *ECOOP 2005-Object-Oriented Programming.* Springer, 2005, pp. 528–550.

[33] B.-C. Tak, C. Tang, C. Zhang, S. Govindan, B. Urgaonkar, and R. N. Chang, "vpath: Precise discovery of request processing paths from black-box observations of thread and network activities." in *USENIX Annual technical conference*, 2009.

[34] B. H. Sigelman, L. A. Barroso, M. Burrows, P. Stephenson, M. Plakal, D. Beaver, S. Jaspan, and C. Shanbhag, "Dapper, a large-scale distributed systems tracing infrastructure," *Google research*, 2010.

[35] Y. Dang, R. Wu, H. Zhang, D. Zhang, and P. Nobel, "Rebucket: a method for clustering duplicate crash reports based on call stack similarity," in *Proceedings of the 34th International Conference on Software Engineering.* IEEE Press, 2012, pp. 1084–1093.

[36] K. Glerum, K. Kinshumann, S. Greenberg, G. Aul, V. Orgovan, G. Nichols, D. Grant, G. Loihle, and G. Hunt, "Debugging in the (very) large: ten years of implementation and experience," in *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles.* ACM, 2009, pp. 103–116.

[37] C. Liu and J. Han, "Failure proximity: a fault localization-based approach," in *Proceedings of the 14th ACM SIGSOFT international symposium on Foundations of software engineering.* ACM, 2006, pp. 46–56.

[38] A. Anandkumar, C. Bisdikian, and D. Agrawal, "Tracking in a spaghetti bowl: monitoring transactions using footprints," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 36, no. 1. ACM, 2008, pp. 133–144.

[39] R. Fonseca, G. Porter, R. H. Katz, S. Shenker, and I. Stoica, "X-trace: A pervasive network tracing framework," in *Proceedings of the 4th USENIX conference on Networked systems design & implementation.* USENIX Association, 2007, pp. 20–20.

[40] H. S. Gunawi, T. Do, J. M. Hellerstein, I. Stoica, D. Borthakur, and J. Robbins, "Failure as a service (faas): A cloud service for large-scale, online failure drills," *University of California, Berkeley, Berkeley*, vol. 3, 2011.

[41] R. Chandra, R. M. Lefever, K. R. Joshi, M. Cukier, and W. H. Sanders, "A global-state-triggered fault injector for distributed system evaluation," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 15, no. 7, pp. 593–605, 2004.

[42] X. Ju, L. Soares, K. G. Shin, K. D. Ryu, and D. Da Silva, "On fault resilience of openstack," in *Proceedings of the 4th annual Symposium on Cloud Computing*. ACM, 2013, p. 2.

[43] W. E. Wong and V. Debroy, "A survey of software fault localization," *Department of Computer Science, University of Texas at Dallas, Tech. Rep. UTDCS-45*, vol. 9, 2009.

[44] I. Lee and R. K. Iyer, "Diagnosing rediscovered software problems using symptoms," *Software Engineering, IEEE Transactions on*, vol. 26, no. 2, pp. 113–127, 2000.

Virtual Machine Introspection

[45] M. Bishop, "A model of security monitoring," in *Fifth Annual Computer Security Applications Conference*. IEEE, 1989, pp. 46–52.

[46] B. D. Payne, M. De Carbone, and W. Lee, "Secure and flexible monitoring of virtual machines," in *Computer Security Applications Conference, 2007. ACSAC 2007. Twenty-Third Annual*. IEEE, 2007, pp. 385–397.

[47] K. Kortchinsky, "Cloudburst," *Black Hat USA.[http://www.blackhat.com/presentations/bh-usa-09/KORTCHINSKY/BHUSA09-Kortchinsky-Cloudburst-PAPER.pdf]*, 2009.

[48] T. Garfinkel, M. Rosenblum *et al.*, "A virtual machine introspection based architecture for intrusion detection." in *NDSS*, vol. 3, 2003, pp. 191–206.

[49] B. D. Payne, "Simplifying virtual machine introspection using libvmi," *Sandia Report*, 2012. [Online]. Available: http://prod.sandia.gov/techlib/access-control.cgi/2012/127818.pdf

[50] ——, "Simplifying virtual machine introspection using libvmi." [Online]. Available: https://www.acsac.org/2014/workshops/mmf/Bryan-Payne-An%20Introduction%20to%20Virtual%20Machione%20Introspection%20Using%20LibVMI.pdf

[51] K. Asrigo, L. Litty, and D. Lie, "Using vmm-based sensors to monitor honeypots," in *Proceedings of the 2nd international conference on Virtual execution environments*. ACM, 2006, pp. 13–23.

Hardware Invariant Techniques

[52] S. Bahram, X. Jiang, Z. Wang, M. Grace, J. Li, D. Srinivasan, J. Rhee, and D. Xu, "Dksm: Subverting virtual machine introspection for fun and profit," in *Reliable Distributed Systems, 2010 29th IEEE Symposium on*. IEEE, 2010, pp. 82–91.

[53] X. Jiang, X. Wang, and D. Xu, "Stealthy malware detection through vmm-based out-of-the-box semantic view reconstruction," in *Proceedings of the 14th ACM conference on Computer and communications security*. ACM, 2007, pp. 128–138.

[54] J. Rhee, R. Riley, D. Xu, and X. Jiang, "Defeating dynamic data kernel rootkit attacks via vmm-based guest-transparent monitoring," in *Availability, Reliability and Security, 2009. ARES'09. International Conference on*. IEEE, 2009, pp. 74–81.

[55] J. Butler and G. Hoglund, "Vice–catch the hookers," *Black Hat USA*, vol. 61, 2004.

[56] D. Cotroneo, R. Natella, and S. Russo, "Assessment and improvement of hang detection in the linux operating system," in *Reliable Distributed Systems, 2009. SRDS'09. 28th IEEE International Symposium on.* IEEE, 2009, pp. 288–294.

[57] R. Hund, T. Holz, and F. C. Freiling, "Return-oriented rootkits: Bypassing kernel code integrity protection mechanisms," in *Proceedings of the 18th USENIX Security Symposium*, 2009, pp. 383–398.

[58] B. Dolan-Gavitt, T. Leek, M. Zhivich, J. Giffin, and W. Lee, "Virtuoso: Narrowing the semantic gap in virtual machine introspection," in *Security and Privacy (SP), 2011 IEEE Symposium on.* IEEE, 2011, pp. 297–312.

[59] Y. Fu and Z. Lin, "Space traveling across vm: Automatically bridging the semantic gap in virtual machine introspection via online kernel data redirection," in *Security and Privacy (SP), 2012 IEEE Symposium on.* IEEE, 2012, pp. 586–600.

[60] Q. Liu, C. Weng, M. Li, and Y. Luo, "An in-vm measuring framework for increasing virtual machine security in clouds," *Security & Privacy, IEEE*, vol. 8, no. 6, pp. 56–62, 2010.

[61] O. S. Hofmann, A. M. Dunn, S. Kim, I. Roy, and E. Witchel, "Ensuring operating system kernel integrity with osck," in *ACM SIGPLAN Notices*, vol. 46. ACM, 2011, pp. 279–290.

[62] S. T. Jones, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, "Antfarm: Tracking processes in a virtual machine environment," in *Proceedings of the USENIX Annual Technical Conference*, 2006, pp. 1–14.

[63] ——, "Vmm-based hidden process detection and identification using lycosid," in *Proceedings of the fourth ACM SIGPLAN/SIGOPS international conference on Virtual execution environments.* ACM, 2008, pp. 91–100.

[64] N. L. Petroni Jr and M. Hicks, "Automated detection of persistent kernel control-flow attacks," in *Proceedings of the 14th ACM conference on Computer and communications security.* ACM, 2007, pp. 103–115.

[65] D. Pelleg, M. Ben-Yehuda, R. Harper, L. Spainhower, and T. Adeshiyan, "Vigilant–out-of-band detection of failures in virtual machines," *Operating systems review*, vol. 42, no. 1, p. 26, 2008.

[66] A. Bovenzi, M. Cinque, D. Cotroneo, R. Natella, and G. Carrozza, "Os-level hang detection in complex software systems," *International Journal of Critical Computer-Based Systems*, vol. 2, no. 3, pp. 352–377, 2011.

[67] L. Wang, Z. Kalbarczyk, W. Gu, and R. K. Iyer, "An os-level framework for providing application-aware reliability," in *Dependable Computing, 2006. PRDC'06. 12th Pacific Rim International Symposium on.* IEEE, 2006, pp. 55–62.

[68] H. Moon, H. Lee, J. Lee, K. Kim, Y. Paek, and B. B. Kang, "Vigilare: toward snoop-based kernel integrity monitor," in *Proceedings of the 2012 ACM conference on Computer and communications security.* ACM, 2012, pp. 28–37.

[69] T. R. Flo, "Ninja - privilege escalation detection system for gnu/linux," Ubuntu Manual, http://manpages.ubuntu.com/manpages/lucid/man8/ninja.8.html, 2005.

[70] S. Jana and V. Shmatikov, "Memento: Learning secrets from process footprints," in *Security and Privacy (SP), 2012 IEEE Symposium on*, 2012, pp. 143–157.

[71] X. Blog, "The intel sysret privilege escalation," http://blog.xen.org/index.php/2012/06/13/the-intel-sysret-privilege-escalation/, 2012, [Online; accessed 29-April-2013].

Hook-based Techniques

[72] P. Padala, "Playing with ptrace, part1," *Linux Journal*, no. 103, Nov. 2002. [Online]. Available: http://www.linuxjournal.com/article/6100

[73] R. Krishnakumar, "Kernel korner: kprobes-a kernel debugger," *Linux Journal*, vol. 2005, no. 133, p. 11, 2005.

[74] N. A. Quynh and K. Suzaki, "Xenprobes, a lightweight user-space probing framework for xen virtual machine," in *USENIX Annual Technical Conference Proceedings*, 2007.

[75] B. D. Payne, "Simplifying virtual machine introspection using libvmi," *Sandia Report*, 2012.

[76] M. D. Ernst, J. H. Perkins, P. J. Guo, S. McCamant, C. Pacheco, M. S. Tschantz, and C. Xiao, "The daikon system for dynamic detection of likely invariants," *Science of Computer Programming*, vol. 69, no. 1, pp. 35–45, 2007.

[77] M. Carbin, S. Misailovic, M. Kling, and M. C. Rinard, "Detecting and escaping infinite loops with jolt," in *ECOOP 2011–Object-Oriented Programming*. Springer, 2011, pp. 609–633.

[78] A. Dinaburg, P. Royal, M. Sharif, and W. Lee, "Ether: malware analysis via hardware virtualization extensions," in *Proceedings of the 15th ACM conference on Computer and communications security*. ACM, 2008, pp. 51–62.

[79] "vmsplice(): A linux integer overflow - vmsplice," Online, http://www.win.tue.nl/~aeb/linux/hh/hh.html#toc12.4.

[80] NIST, "Vulnerability summary for cve-2008-0600," Online, https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2008-0600, USA, 2008.

[81] J. Arnold and M. F. Kaashoek, "Ksplice: Automatic rebootless kernel updates," in *Proceedings of the 4th ACM European conference on Computer systems*. ACM, 2009, pp. 187–198.

[82] B. D. Payne, M. Carbone, M. Sharif, and W. Lee, "Lares: An architecture for secure active monitoring using virtualization," in *Security and Privacy, 2008. SP 2008. IEEE Symposium on*. IEEE, 2008, pp. 233–247.

[83] M. I. Sharif, W. Lee, W. Cui, and A. Lanzi, "Secure in-vm monitoring using hardware virtualization," in *In Proc of the 16th ACM Conference on Computer and Communications Security*, ser. CCS '09. New York, NY, USA: ACM, 2009, pp. 477–487.

[84] O. Agesen, J. Mattson, R. Rugina, and J. Sheldon, "Software techniques for avoiding hardware virtualization exits." in *USENIX Annual Technical Conference*, 2012, pp. 373–385.

[85] D. Spinellis, "Trace: A tool for logging operating system call transactions," *ACM SIGOPS Operating Systems Review*, vol. 28, no. 4, pp. 56–63, 1994.

[86] S. J. Vaughan-Nichols, "No reboot patching comes to linux 4.0," Online, http://www.zdnet.com/article/no-reboot-patching-comes-to-linux-4-0/, 2015.

[87] Z. Zhou, V. D. Gligor, J. Newsome, and J. M. McCune, "Building verifiable trusted path on commodity x86 computers," in *Security and Privacy (SP), 2012 IEEE Symposium on*. IEEE, 2012, pp. 616–630.

[88] S. M. Larson, C. D. Snow, M. Shirts *et al.*, "Folding@ home and genome@ home: Using distributed computing to tackle previously intractable problems in computational biology," 2002.