An Effective Scheme for Protecting against Windows Kernel-mode Rootkits

林士嘉、曹偉駿

E-mail: 321521@mail.dyu.edu.tw

ABSTRACT

More and more malicious programs are combined with rootkits to shield their illegal activities so that result makes information security defense encounters a great challenge. It can be observed that most sophisticated kernel mode rootkits are implemented to execute hiding tasks through drivers in Windows Kernel.In the current prevention schemes, the memory shadowing, kernel-mode code signing walkthrough and host-based intrusion prevention system are all to passively protect the operating systems, and they cannot identify whether rootkits intrude in the operating systems. On the other hand, though many companies or individuals have developed rootkit detectors to the public and undoubtedly they can detect known rootkits effectively, they cannot foresee what the result is when meeting unknown rootkits and crashed operating systems. Hence, the thesis will develop a prevention mechanism which can identify driver-hidden rootkits to protect the Windows-based operating systems. Our research constructs an anti-rootkit scheme for protecting Windows kernel to higher system security, especially for safeguarding Windows kernel from the damages of unknown driver-hidden rootkits. Moreover, we also test the proposed prevention scheme by Windows XP SP3 on the Testbed@TWISC platform. We affirm that our efforts are extremely useful for improving the current techniques of preventing Windows driver-hidden rootkits.

Keywords : malware、rootkit、windows、kernel mode、System Security

Table of Contents

REFERENCES

一、中文部份 TWISC@NCKU(2007)，Testbed@TWISC介紹 [線上資料]，來源: http://testbed.ncku.edu.tw/docfortestbed/20071002_Testbed@TWISC_introduction.pdf [2010, January 10]。 陳昱成 (2008)，變形的Windows Kernel Mode Rootkit 分析研究，私立大葉大學資訊管理研究所未出版之碩士論文。 二、英文部份 Andreas, B. (2004). UNIX and Linux based Rootkits Techniques and Countermeasures. In G. Harris & U. Harder (Eds.), Papers presented at the 16th Annual FIRST Conference on Computer Security Incident Handling (pp. 27-34), United Kingdom: London. Bulter, J., Undercoffer, J. L., & Pinkston, J. (2003). Hidden process: the implication for intrusion detection. Papers presented at the IEEE International Workshop on Information Assurance, (pp. 116-121), Germany: Berlin. Baliga, A., Iftode, L., & Chen, X. (2008). Automated containment of rootkits attacks. Computers & Security, 27(7-8), 323-334. Bellard, F. (2005). QEMU, a fast and portable dynamic translator. Proceedings of the USENIX Annual Technical Conference. Barham, P., Dragovic, B., Fraser, K., Hard, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., & Warfield, A. (2003). Xen and the art of virtualization. Proceedings of the Fifth

ACM Symposium on Operating Systems Principles, (pp. 164-177). Germany: Berlin. Bulter, J., & Honglund, G. (2004). Rootkit forum [online]. Available : http://www.rootkit.com [2009, December 14]. Cogswell, B., & Russinovich, M. (2005). RootkitRevealer [online]. Available: http://technet.microsoft.com/en-us/sysinternals/bb897445.aspx.
[2009, December 28]. Chian, K., & Lloyd, L. (2007). A case study of the rustock rootkit and spam bot. Papers presented at the USENIX First Workshop on Hot Topics in Understanding Bonets. Chuvakin, A. (2003). An overview of unix rootkits. iALERT White Paper, iDefense Labs, Chantilly, Virginia. Felten, E. W., & Halderman, J. A. (2006). Digital rights management, spyware, and security. IEEE Security & Privacy, 4(1), 18-23. Florio, E. (2005). When malware meets rootkits [online]. Available: http://www.symantec.com/avcenter/reference/when.malware.meets.rootkits.pdf [2009, December 14]. Fuchsberger, A. (2005). Intrusion detection systems and intrusion Prevention Systems. Information Security Technical Report, 134-139. Geer, D. (2006). Hackers get to the root of the problem. IEEE Computer, 39(5), 17-19. Goldberg, R. (1974). Survey of virtual machine research. IEEE Computer Magazine, 7(6), 34-45. Hoglund, G., & Butler, J. (2005). Rootkits:Subverting the windows kernel. Addison-Wesley. Hunt, G., & Brubacher, D. (1999). Detours: Binary interception of win32 functions. Proceedings of the 3rd USENIX Windows NT Symposium, 135-143. Ierace, N., Urrutia, C., & Bassett, R. (2005). Intrusion prevention Systems ACM Press [online]. Available: http://www.acm.org /ubiquity/views/v6i19_ierace.html?CFID=66131608&CFTOKEN=15275447[2009, December 14]. King, T. S., & Chen, M. P. (2005). Backtracking intrusions. ACM Transactions on Computer Systems, 23(1), 51-76. King, S., Chen, P., Wang, Y., Verbowski, C., Wang, H., & Lorch, J. (2006). SubVirt: Implementing malware with virtual machines. Proceedings of the IEEE Symposium on Security and Privacy, 314-327. Kruegel, C., Robertson, W., & Vigna, G. (2004). Detection kernel-level rootkits through binary Analysis. Proceedings of the Annual Computer Security Applications Conference. Keong, C. (2004). Defeating Kernel Native API Hookers by Direct Service Dispatch Table Restoration [Online]. Available: http://www.packetstormsecurity.org/hitb04/hitb04-chew-keong-tan.pdf [2009, December 14]. Kreibich, C., & Crowcroft, J. (2004). Honeycomb: Creating intrusion detection signatures using honypots. ACM SIGCOMM Computer Communication Review, 34(1), 51-56. Kim, G. H., & Spafford, E. H. (1994). The design and implementation of tripwire: a file system integrity checker. Proceedings of the 2nd ACM Conference on Computer and Communications security (pp. 18-29), USA: Virginia. Kumar, E. (2006). Battle with the unseen – understanding rootkits on Windows. Proceedings of the AVAR International conference (pp. 96-112) United States of America: Washington. Laureano, M., Maziero, C., & Jamhour, E. (2004). Intrusion detection in virtual machine environments. Proceedings of the 30th EUROMICRO Conference (520-525). Laureano, M., Maziero, C., & Jamhour, E. (2007). Protecting Host-Based intrusion detectors through virtual machines. Computer Networks, 51(5), 1275-1283. Molina, D., Zimmerman, M., Roberts, G., Eaddie, M., & Peterson, G. (2008). Timely rootkit detection during live response. IFIP International Federation for Information Processing, 28(5), 139-148. McAfee. (2006). Rootkits, Part 1 of 3: The Growing Threat [Online].Available: http://www.mcafee.com/us/local_content/white_papers/threat_center/wp_akapoor_rootkits1_en.pdf [201 0, January 1]. Microsoft, Kernel-Mode Code Signing Walkthrough [Online]. Available: http://www.microsoft.com/whdc/winlogo/drvsign/kmcs_walkthrough.mspx [2010, January 10]. Microsoft. (1985). VirtualPC [Online] Available: http://w ww.microsoft.com/windows/virtual-pc/ [2009 December 14]. Petroni, N. L., Fraser, T., Molina, J., & Arbaugh, W. A. (2004). Copilot-a coprocessor-based kernel runtime integrity monitor. Proceedings of the 13th Usenix Security Symposium, (179-194), USA. Riley, R., Jiang, X., & Xu, D. (2008). Guest-transparent prevention of kernel rootkits with VMM-based memory shadowing. Proceedings of the International Symposium on Recent Advances in Intrusion Detection (pp. 1-20) United States of America: Pittsburgh. Rutkowska, J. (2006). Subverting Vista Kernel For Fun And Profit [online]. Available: http://www.blackhat.com/presentations /bh-usa-06/BH-US-06-Rutk owska.pdf [2010, January 10]. Ronald, R. (1995). MD5 [Online]. Available: http://en.wikipedia. org/wiki/MD5 [2010, January 10]. Schreiber, S. (2001). Undocumented Windows 2000 Secrets: A Programmer ’ s Cookbook. Addison-Wesley. Solomon, D.A. (1998). The Windows NT Kernel Architecture, computer, 31(10), 40-47. Sequeira D. (2003). Intrusion prevention systems: security's silver bullet?. Business Communications Review (pp. 36-41). Sun Microsystems. (1986). Virtu alBox [online]. Available: http://ww w.virtualbox.org/ [2009, December 14]. Tsaur, W. J., Chen, Y. C., & Tsai, B. Y. (2009). A new windows driver-hidden rootkit based on direct kernel object manipulation. Lecture Notes in Computer Science, 5574, 202-213. University of Cambridge. (1209). Xen [Online]. Available: http://ww w. xen.org/ [2009, December 14]. VMware. (1998). VMware [online] Available: http://www.vmw are.com/ [2009, December 14]. Youseff, L., Wolski, R., Gorda, B., & Krintz C. (2006). Paravirtualization for HPC systems. Lecture Notes in Computer Science, 5(7), 474-486.