Journal of Information Security, 2015, 6, 101-110
Published Online April 2015 in SciRes. http://www.scirp.org/journal/jis
http://dx.doi.org/10.4236/jis.2015.62011

Quantifying Malware Evolution through Archaeology

Jeremy D. Seideman[1*], Bilal Khan[2], Cesar Vargas[3]

[1]The Graduate School and University Center (CUNY), New York, USA
[2]Department of Math and Computer Science, John Jay College (CUNY), New York, USA
[3]NacoLabs Consulting, LLC, New York, USA
Email: [*]jseideman@gradcenter.cuny.edu, bkhan@jjay.cuny.edu, cesar@nacolabs.com

Received 10 January 2015; accepted 27 March 2015; published 31 March 2015

Abstract

Dynamic analysis of malware allows us to examine malware samples, and then group those samples into families based on observed behavior. Using Boolean variables to represent the presence or absence of a range of malware behavior, we create a bitstring that represents each malware behaviorally, and then group samples into the same class if they exhibit the same behavior. Combining class definitions with malware discovery dates, we can construct a timeline of showing the emergence date of each class, in order to examine prevalence, complexity, and longevity of each class. We find that certain behavior classes are more prevalent than others, following a frequency power law. Some classes have had lower longevity, indicating that their attack profile is no longer manifested by new variants of malware, while others of greater longevity, continue to affect new computer systems. We verify for the first time commonly held intuitions on malware evolution, showing quantitatively from the archaeological record that over 80% of the time, classes of higher malware complexity emerged later than classes of lower complexity. In addition to providing historical perspective on malware evolution, the methods described in this paper may aid malware detection through classification, leading to new proactive methods to identify malicious software.

Keywords

Malware, Classification, Evolution, Dynamic Analysis

1. Introduction

When performing analysis on malicious software, or malware, it is important to be able to group similar mal-

[*]Corresponding author.

How to cite this paper: Seideman, J.D., Khan, B. and Vargas, C. (2015) Quantifying Malware Evolution through Archaeology. *Journal of Information Security*, **6**, 101-110. http://dx.doi.org/10.4236/jis.2015.62011

Finally, this method can be used to classify very specific behavior, given that classification of behaviors can be as fine-grained as desired. A detector can be programmed to look for *any* operation within a category of behavior; if we are looking at registry changes, we can design our scheme to look for the most specific registry change we want to use as a basis of classification. This level of customization allows for more targeted detectors which, while not always useful in the real-world, are useful in an isolated setting as part of a reverse-engineering approach.

References

[1] Classification of Species, 2009. https://web.archive.org/web/20120121022919/http://classes.entom.wsu.edu/348/classification.htm

[2] Seideman, J. (2009) Recent Advances in Malware Detection and Classification: A Survey. Technical Report, The Graduate School and University Center of the City University of New York.

[3] Spafford, E.H. (1994) Computer Viruses as Artificial Life. *Artificial Life*, **1**, 249-265. http://dx.doi.org/10.1162/artl.1994.1.3.249

[4] Bailey, M., Oberheide, J., Andersen, J., Morley Mao, Z.Q., Jahanian, F. and Nazario, J. (2007) Automated Classification and Analysis of Internet Malware. *Proceedings of RAID* 2007, 178-197. http://dx.doi.org/10.1007/978-3-540-74320-0_10.

[5] Riau, C. (2002) A Virus by Any Other Name: Virus Naming Practices. http://www.symantec.com/connect/articles/virus-any-other-name-virus-naming-practices

[6] Gandotra, E., Bansal, D. and Sofat, S. (2014) Malware Analysis and Classification: A Survey. *Journal of Information Security*, **5**, 56-64. http://dx.doi.org/10.4236/jis.2014.52006

[7] Lee, T. and Mody, J.J. (2006) Behavioral Classification. *Proceedings of EICAR* 2006, May 2006, 1-17.

[8] Szor, P. (2005) The Art of Computer Virus Research and Defense. Addison-Wesley, New York.

[9] Jacob, G., Debar, H. and Filiol, E. (2008) Behavioral Detection of Malware: From a Survey towards an Established Taxonomy. *Journal in Computer Virology*, **4**, 251-266. http://dx.doi.org/10.1007/s11416-008-0086-0

[10] Andreas Moser, Christopher Kr¨ugel, and Engin Kirda. (2007) Exploring Multiple Execution Paths for Malware Analysis. *Proceedings of the* 2007 *IEEE Symposium on Security and Privacy*, 2007, 231-245. http://dx.doi.org/10.1109/SP.2007.17

[11] (2009) Norman Sandbox. https://web.archive.org/web/20101005091013/http://www.norman.com/technology/norman_sandbox/

[12] Liang, Z.K., Sun, W.Q., Venkatakrishnan, V.N. and Sekar, R. (2009) Alcatraz: An Isolated Environment for Experimenting with Untrusted Software. *ACM Transactions on Information and System Security*, **12**, 1-37.

[13] Buyrukbilen, S. and Deryol, R. (2008) An Automated System for Behavioral Malware Analyis. Technical Report, John Jay College of Criminal Justice, City University of New York.

[14] (2010) The Honeynet Project. http://www.honeynet.org

[15] (2010) Nepenthes—Finest Collection. http://nepenthes.carnivore.it/

[16] (2012) Dionaea-Catches Bugs. http://dionaea.carnivore.it/

[17] (2010) Offensive Computing: Community Malicious Code Research and Analysis. http://www.offensivecomputing.net

[18] VX Heavens, 2010. http://vxheaven.org/

[19] Symantec, 2012. http://www.symantec.com/index.jsp

[20] VirusTotal, 2008. http://www.virustotal.com

[21] (2012) Threat Explorer—Spyware and Adware, Dialers, Hack Tools, Hoaxes and Other Risks. http://www.symantec.com/security_response/threatexplorer/

[22] Oberheide, J., Cooke, E. and Jahanian, F. (2008) Cloudav: N-version Antivirus in the Network Cloud. *Proceedings of the* 17*th USENIX Security Symposium*, 91-106. http://www.usenix.org/events/