Behaviour Based Malware Detection

Tristan Aubrey-Jones
School of Electronics and Computer Science, University of Southampton, UK
taj105@ecs.soton.ac.uk

Abstract

This article looks at the future of antivirus technology in IT security, discussing some of the latest malware threats and counter developments. We specifically examine key developments in proactive malware detection based on real-time behavioural analysis, to combat 0-day threats.

1. Introduction

“Malware” is the generic term for malicious computer programs like Viruses, Worms and Trojans written to make illegitimate use of a computer system, purposed by those without the right to do so. Such programs use a variety of different techniques to access and exploit victim systems but whatever these may be, it is a vital role of IT security to frustrate them. The primary countermeasure against this type of threat is antivirus software; software that seeks to detect malicious code and disable it.

2. The Battle Thus Far

The predominate method of virus detection searches files for “signatures”, binary patterns that occur in the virus, to see if they are infected. Signatures are created manually by antivirus companies as they analyse new viruses, and are distributed over the internet. In the early 1990s the first “polymorphic” viruses emerged. These frustrate this method of detection by disguising themselves as they replicate, so that no two copies of the virus look alike. The first versions of these viruses just encrypted their bodies with random keys, but it wasn’t long before viruses were written that randomly mutated their decryption routines as well, making detection very difficult.

To combat the surge of polymorphic viruses antivirus software now include dynamic analysers that allow potential viruses to unpack in an emulated environment, while scanning the memory that they modify for signatures so that the virus reveals itself and is then detected. Various sophisticated virus unpackers [1] have been developed using this approach, many of which employ heuristics to accelerate analysis, but recent threats present a further problem.

In the first six months of 2007 Symantec detected over 200,000 new malicious code threats, giving a 185% increase over the previous six months [2]. All of the detection techniques above rely on the prompt creation of signatures for each new threat, but this flood of new threats has swamped virus analysts. In response to this vast influx of new malware, research has been made into automated analysis techniques to accelerate the process. Most important is research into automated behaviour analysis tools (such as TTAnalyze [3]) which execute the malicious code in a safe environment, like an emulated or isolated computer to study its behaviour. However even these AV analyst tools, in their current form, are inadequate in addressing some recent more organised threats.

“Botnets”, networks of trojanized machines, are starting to use “offline polymorphism” [4] where they regularly download new updates from the internet before signatures can be created and distributed, thus hiding their mutation algorithm from the AV analyst. For example “Storm” is part Trojan, part Botnet and part Worm and has compromised between 0.25 and 10 million machines [5]. It uses a peer to peer network to distribute new variants faster than the AV companies can react. These Botnets are creating a “dark” backbone to the Internet, which is not easily removed and are being leveraged for profit, for example by distributing the hosting of phishing sites using “Fast-flux” [6], to frustrate their removal.

This offline polymorphism may have found the limit of signature based detection. A key development along the road to solving the problem has been recent research into behaviour based detection techniques.

3. Behaviour Based Detection

For antivirus scanners to detect malware before it has been studied they must perform some sort of automatic analysis themselves. Ultimately analysis must be behaviour based, because no matter the disguise, a piece of malware will behave badly, that is its purpose. One technique for analysing the behaviour of a program is to study the sequence of operating system calls it makes [7]. Antivirus software can intercept these API calls while a program is running, and use heuristics to look for suspicious activity, terminating those with harmful behaviour. Various heuristics have been researched, such as looking for patterns used for self replication [8], but these all rely on monitoring a program once it is running. This is dangerous to rely on because the malware might cause harm to the system before it is recognized as malicious.

Alternatively we can adapt this approach to scan a program by observing its execution it in an emulated machine. TTAnalyse [3], a recently developed analysis tool, uses the QEMU [9] emulator to do this, but hardware emulators have always been very slow, and virus writers have exploited this by using processor hungry routines such as brute forcing their own decryption [10], so that native execution might take 5 seconds but emulation would take 10 minutes.

A development that builds on this research could be set to solve this problem by using "virtualization" as opposed to hardware emulation, so that a large portion of the instructions are executed directly, making behaviour analysis fast enough to include in antivirus software. This is done through "Dynamic Binary Translation" [11] which translates and caches binary code, replacing API calls so that they modify virtual resources rather than the real system. This gives fast execution, but also safely isolates the program and allows intimate observation of its activities. CWSandbox [12] has recently been developed as a tool for AV analysts and uses a similar method. It does use a virtual machine using DBT, but the analyser software itself is also executed in the virtualized environment and uses inline code overwriting to hook the API functions which could allow malware to detect analysis, and change behaviour to avoid detection. Current research in this area is concerned with further accelerating DBT, and enabling it to cope with self modifying and multi-threaded code [13], two important features in contemporary malware. The future of this technology lies in "hardware virtualization", new processor architectures which include instructions to support fast virtualization, such as AMD's AMD-V [14] and Intel's VT-x [15], enabling a new generation of "proactive" antivirus protection. All executables could be analyzed in a virtual machine before they are first executed, and disabled if they are found to have malicious intent.

4. Impact

The deployment of these behaviour based detection developments, could serve to drastically improve the security of an organisation. Real-time behavioural analysis could be initially deployed on email servers and web gateways and serve to detect bespoke Trojans intended for espionage [16], as well as repelling 0-day threats and offline polymorphic viruses. The likely expense of products may limit their initial distribution and therefore fail to deal with existing Botnets of largely unprotected machines, but could drastically limit their rate of growth. Deployment of this technology would no doubt provoke the criminal community to develop new methods to evade detection. For example logic and time bombs that only become malicious under certain circumstances may not be detected, and methods to detect virtualization are bound to increase in sophistication. However verifying the behaviour and intent of a program, rather than just its appearance is certainly a step in the right direction for security in IT.

5. References

[1] S. Josse, "Secure and advanced unpacking using computer emulation," *Journal in Computer Virology*, vol. 3, no. 3, pp. 221-236, Aug. 2007.

[2] Symantec, "Symantec Internet Security Threat Report, Trends for January-June 07," vol. XII, Sept. 2007. [Online] Available: http://www.symantec.com/threatreport/. [Accessed Jan. 15, 2008].

[3] U. Bayer, C. Kruegel, E. Kirda, "TTAnalyze: A Tool for Analyzing Malware," in 15th Annual Conference of the European Institute for Computer Antivirus Research (EICAR), 2006.

[4] M. Schipka, "A road to big money: evolution of automation methods in malware development," in Proceedings VB2007 Conference, Sept. 2007.

[5] P-M. Bureau, A. Lee, "Malware Storms: A Global Climate Change," *Virus Bulletin*, Nov. 2007, pp. 12-16.

[6] A. Solomon, G. Evron, "The world of Botnets," Virus Bulletin, Sept. 2006.

[7] J. Xu, A.H. Sung, P. Chavez, S. Mukkamala, "Polymorphic malicious executable scanner by API sequence analysis," in 4th International Conference on Hybrid Intelligent Systems (HIS), 2004, pp. 378–383.

[8] A. Volynkin, V.A. Skormin, D.H. Summerville, J. Moronski, "Evaluation of Run-Time Detection of Self-Replication in Binary Executable Malware," in IEEE Information Assurance Workshop, 2006, pp. 184-191.

[9] F. Bellard, "QEMU, a Fast and Portable Dynamic Translator," in Proceedings of the USENIX Annual Technical Conference, 2005, pp. 41+.

[10] P. Ször, D. Fellows, "Bad IDEA," *Virus Bulletin*, Apr. 1998, pp. 18-19.

[11] K. Scott, J. Davidson, "Safe Virtual Execution Using Software Dynamic Translation," in Proceedings of the 18th Annual Computer Security Applications Conference, 2002, pp. 209-218.

[12] C. Willems, T. Holz, G. F. Freiling, "Toward Automated Dynamic Malware Analysis Using CWSandbox," *IEEE Security & Privacy*, vol. 5, no. 2, Mar. 2007, pp. 32-39.

[13] J. Wu, "Full Potential of Dynamic Binary Translation for AV Emulation Engine," in VB2006 Conference, Oct. 2006.

[14] AMD, *AMD64 Architecture Programmer's Manual Volume 2: System Programming*, pub no. 24593, Sept. 2007, pp. 367+. Available: http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/24593.pdf

[15] Intel, *Intel® Virtualization Technology for Directed I/O Architecture Specification*, pub no. D51397-003, Sept. 2007, pp 12-13. Available: http://download.intel.com/technology/computing/vptech/Intel(r)_VT_for_Direct_IO.pdf.

[16] Rhys Blakely, Jonathan Richards, James Rossiter, and Richard Beeston, "MI5 Alert on China's Cyberspace Spy Threat," TimesOnline, December 1, 2007. [Online] Available: http://business.timesonline.co.uk/tol/business/industry_sectors/technology/article2980250.ece [Accessed: Dec. 11, 2007].