Biological Aspects of Computer Virology

Vasileios Vlachos[1], Diomidis Spinellis[2], and Stefanos Androutsellis-Theotokis[2]

[1] Department of Computer Science and Telecommunications
Technological Educational Institute of Larissa
vsvlachos@gmail.com
[2] Department of Management Science and Technology
Athens University of Economic Business
{dds,stheotok}@aueb.gr

Abstract. *Recent malware epidemics proved beyond any doubt that frightful predictions of fast-spreading worms have been well founded. While we can identify and neutralize many types of malicious code, often we are not able to do that in a timely enough manner to suppress its uncontrolled propagation. In this paper we discuss the decisive factors that affect the propagation of a worm and evaluate their effectiveness.*

Key words: Malware, Computer Epidemiology, Artificial Immune Systems

1 Introduction

Computer viruses and worms are definitely not a new threat as they exist for several decades. The striking difference between the "ancient" viruses and the modern ones lies in the time-frame in which they operate. Ancient viruses needed weeks or even months to propagate and reach a noticeable level of prevalence because of the completely different means of infection, such as diskettes, on which they relied. On the contrary, modern viruses and worms utilize the Internet and other high-speed networks achieving sizable infection rates. Theoretical studies [44, 51, 52], but also empirical evidence [32] suggests that last generation worms are perfectly capable of infecting a susceptible population in about 15 minutes. The construction of rapid malcode is by no means an easy task. While thousands of worms exist, only a small fraction managed to prevail in an observable level and only a handful of them to create epidemic outbreaks. Similarly, though

In *3rd International Conference on e-Democracy*, 23–25 September 2009, Athens, Greece

thousands biological pathogens survive, just a small percentage of them is able to cause major health threat. Therefore it would be useful to stand on the experience of practical epidemiologists, so as to identify the major factors that dominate the propagation of a biological pathogen and thereafter to try to correlate these factors with components that may affect the virulence of computer malware. The rest of the paper is organized as follows. Section 2 surveys the existing literature on biologically inspired computer security research. In Section 3 we discuss the effective parameters that can lead to infectious diseases epidemics and draw the first analogies between biological and computer virulence. In Sections 4 and 5 we present and analyze these factors, while in Section 6 we fit these findings in the concept of Computer Epidemiology. Section 7 concludes this paper.

2 Related Work

The similarities between biological pathogens and computer malware are semantically evident, as the most popular description of malcode suggests. Computer virus is the term that F. Cohen with his supervisor L. Adleman coined to describe the earliest and simplest forms of malicious software [9, 10]. Even before the formalism that was developed from Adleman's term and Cohen's work, an other category of misbehaving programs was described with another biological analogy as *rabbits* [45]. The similitude of computer malcode and live pathogens was not overlooked by the research community. In particular, researchers [28] looked in great detail characteristics of computer worms and well known biological diseases and tried to compare specific types of pathogens with prominent species of malware and examine their most important properties in regard to their propagation success. Other efforts [54] concentrated on public health policies that are in place against major epidemics (Acquired Immune Deficiency Syndrome – AIDS) and proposed equivalent public policies for computer malware. An extensive review of the literature reveals that two basic strategies are available to tackle the malware problem: A microscopic approach that extends the analogies between the biological viruses and computer malware and tries to develop artificial systems with similar functionality to the human immune system and a macroscopic methodology that employs epidemiological tools to gain insights in the propagation dynamics of rapid malcode as happens with the infectious diseases.

Many efforts focused on the application of the basic mechanisms of the immune system to computer security. Forrest et al. [14, 15, 43] implement some immunological functions paying attention to the mechanism that distinguishes self to non-self elements of the human body so as to embed a similar technique to computer systems that is capable of recognizing legitimate use from abuse. A large part of their work has found application to the 'pH' patch for the Linux kernel [42] with promising results, while some other efforts that also use immunological concepts are in early stages [34]. Aickeling & Greensmith claim

that the insufficiency of various artificial immune systems to address the problem of computer security could be due to use of older immunological models. In their work they employ the most recent immunological theories. In collaboration with practical immunologists they implement two algorithms, the Dendritic Cell Algorithm (DCA) [21] and the Toll-Like Receptor Algorithm (TLR) [1]. These algorithms are part of the Danger Theory Project [11], which adapts the recent theories of immunologists. According to the Danger Theory, a complex signaling mechanism is responsible for the activation of the immune system, rather the simplistic 'non-self' versus 'self' principle [29]. The fact that the Danger Theory is not unanimously accepted in the medical world [31], raises some questions about its effectiveness as viable model for Artificial Immune Systems.

Burgess [8] dealt with the most basic foundations of biology such as health and sickness and tried to express them as security policies. Many of his ideas have been realized in the *cfengine* project [7]. Of particular interest are his remarks about redundancy, which he founds of limited use in computer security. On the other hand, others base their work on this concept and accept as unavoidable the loss of some computer systems. Though this loss is not pleasant at larger scale can act as an alarm for the majority of the networked components [48]. According to previous research [49, 50] principles of Computer Hygiene could slow down the spread of malicious agents. Another biological inspired approach focuses on models that try to mimic the functionality of genomics and protomics. Goel and Bush [19] propose a system that is able to create new virus - signatures by mutation of the existing signatures. Shafi and Abbass [40] present in their work a more holistic approach as they consider several Complex Adaptive Systems with foundations in the physical world. They examine paradigms of security systems that utilize Genetics Based Machine Learning, Swarm Intelligence and Coevolution.

Most of the conducted research tackles specific aspects of computer virulence in order to find appropriate means to minimize the risk of a malware epidemic. This paper investigates the joint effects of the factors that dominate the propagation of malicious agents. Three major components seem to dominate the propagation of a worm, the *Infection Propagator*, the *Target Locator* and the *Worms Virulence*.

3 Infection Propagator

The type of the attacked vulnerability is one of the most critical factors that heavily affect the virulence of a worm. Generally what influences the outcome of that choice, could be best described by the prevalence of the exploited vulnerability, the age of the vulnerability at time of exploitation and the exploitation difficulty of the attacked vulnerability.

3.1 Vulnerability Prevalence

Worm writers normally prefer to infect the largest possible number of susceptible systems. To maximise the number of the contaminated systems it is necessary for a worm to exploit a popular vulnerability. As [18, 17, 20] showed, the homogeneity of operating systems and applications is perfectly suited for worm writers. The fact that up to 95 percent [35] of all the computer systems in the world use some version of Microsoft's Windows operating system and the majority of them have also a version of Microsoft's Office installed make them attractive targets for any kind of attack. Given that Microsoft's operating systems and applications exhibit a large number of vulnerabilities renders the situation even worse. The OpenBSD operating system on the other hand, which according to its authors, suffered 'only two remote holes in the default install, in more than 10 years' and occupies less than 1 percent share is obviously less attractive as a target. This asymmetry leads directly to a constantly increasing number of attacks against the popular operating systems and applications making their use even more insecure. The monocultures have also significant impact on other aspects of the worm's development process. The possibility of contaminating up to 95 percent of the susceptible population using only a single infection vector lowers the bar regarding the required skills of a worm writer. Complicated malcode, such as the Slapper worm [3] necessitate extra work to utilize multiple attack vectors in order to infect a number of different distributions of the Linux operating system thus requiring much more effort from a worm writer to achieve similar effects with a worm that operates in a homogeneous environment. It has yet to be decided whether we prefer to have highly homogeneous environments so as to avoid the cost of portability and to further enhance the standardization of software development, or we should pay more attention to the security disadvantages of that approach and start building more heterogeneous systems. Even if we stick to the current monoculture, whether we have other means to diminish these effects and whether it is possible to obtain software diversity as a countermeasure, are nevertheless open questions.

3.2 Age of the Vulnerability at Time of Exploitation

The cycle from discovering a vulnerability till the development of a patch is a lengthy process that requires a number of intermediate steps. Obviously, recently discovered vulnerabilities are much more promising from a worms' writer perspective, because most users need several days, if not weeks or longer, to update their systems. If a vulnerability is recent enough it is highly probable that a significant number of systems will be unpatched and hence unprotected. Recent evidence indicates that modern malcode tends to minimise the time interval between the disclosure of a vulnerability and its exploitation. The Witty worm [41] took advantage of a vulnerability that was announced only the day before, however the great fear is for worms that will exploit an unknown or *zero-day* vulnerability. To protect better against known threats the standard methodology

followed by vendors, researchers and system administrators involves: *vulnerability discovery, patch development and testing* and *vulnerability announcement and patch deployment*.

This procedure showed positive results, but also highlighted a number of downsides. When a vulnerability is announced, both the legitimate users and the malicious crackers become aware of it. Thus, adversaries start actively seeking susceptible non-patched systems in order to exploit them. It is important to note that the technical skills required to discover a vulnerability are quite different and much higher than to exploit a public announced one. Furthermore, the reverse-engineering of a patch offers valuable information to an adversary allowing him to develop in a lesser time an exploit for the specific vulnerability. Recent research [39] provides provocative but also sound arguments to keep some discovered vulnerabilities secret, questioning the way we handled the vulnerabilities disclosure procedure till now. We are confident that this and other related studies [4] will initiate some very interesting debates in the near future regarding whether, when and who should publicly announce vulnerabilities.

3.3 Exploitation Difficulty

As security becomes a major factor during the software development lifecycle not only the number of vulnerabilities diminishes, but also they become much more difficult to be exploited. Hence we can observe a switch from the traditional and easy to implement *stack smashing* techniques to much more sophisticated *arc injections, pointer subterfuge* and *heap smashing* attacks [38]. While these developments are overall positive, they lead to a new breed of malicious crackers with exceptional skills. Unfortunately these crackers don't limit their operations only to breaking systems but also write highly advanced worms such as the Slapper worm. These advancements seem to conclude the shift of successful worm writers from disgruntled teenagers with limited abilities (also known as *script kiddies*, because they tend to use already available tools and code instead of developing their own) to highly skilled malevolent programmers. It remains to be seen what other measures or designs should be embedded in the future programming languages in order to further limit the space in which the malicious crackers operate.

4 Target Locator

A worm, in order to propagate successfully, should have an efficient target locator. Staniford et al showed the great importance of the propagation strategy of a worm as different propagation dynamics can cause completely different outcomes in the spread of a worm. In their seminal work [44] they presented a short-list of the most eminent target locators namely *Random Scanning*, *Localized Scanning*, *Hit-list Scanning*, *Permutation Scanning*, *Topological Scanning* and argued for or against their efficiency. They also coined the terms *Warhol Worm* and *Flash Worm*.

5 Worm Virulence

Pathogens, microbes and parasites share a common behavior with artificial viruses regarding their propagation, because of their virulence. The virulence of a microorganism (such as a bacterium or virus) is defined as a measure of the severity of the disease it is capable of causing [30]. Ebola hemorrhagic fever has one of the highest mortality and fatality rates and therefore is able to eradicate small villages, but because of the severe symptoms and the short incubation time is almost never spread over large geographic areas [37]. On the contrary the influenza virus has usually mild symptoms and therefore the patients neglect to search for cure during the early phases of the infections, which turns them to a carriers of the disease to a large number of the population. During the 20th century the influenza A is responsible for deaths of 20 to 40 million persons [47], while the *Spanish flu* [6] is still considered as one of the worst pandemics ever. The biological analogy between the destructiveness of the malware, measured by the rate of worm-induced host mortality, and the parasite virulence has brought to general attention the underestimated, but nonetheless critical factor of the effectiveness of worm virulence. Most of the successful worms did not carry an explicitly destructive payload [16]. Undoubtedly the more harmful a worm is, the more attention it attracts. Therefore, it is highly probable that if a worm has been developed just for fun or for surveillance purposes without damaging properties, it will not be easily noticed.

On the other hand, Hofmeyr [22] argues that the virulence of malware is a far more complicated issue and poses interesting questions regarding the interaction between different types of malware coexisting in the same host. Though these interactions have been studied in biology [53], they are still neglected in the case of malware. The consequences of this omission may become evident in the feature as it is quite common for different types of malware to compete for the same resources. Malware writers started to use to their benefit the spread of the other types of malcode as it can be seen from the infection techniques of the Nimda worm which took advantage of the Code Red II and Sadmind backdoors. Often worm writers tend to act antagonistically as was with the heavily noticed Netsky – MyDoom wars, but most of the times malcode works synergistically. An arguable [33] solution to slowdown most of the existing worms proposes the release of 'good' worms that will search and eliminate both the malicious worms by deleting them and simultaneously reduce the number of the susceptible hosts by upgrading specific vulnerabilities making them immune to future attacks that target the specific vulnerabilities [26, 46].

6 Computer Epidemiology

As showed in the previous sections many factors contribute to the success or the failure of a worm. To which extent each one of them affects their overall performance and consequently where we should concentrate our efforts to suppress their spread are still open questions which we will have to focus on in the near

future. Numerous renowned scientists, including Daniel Bernoulli, Ronald Ross, Lowell Reed and Wade Hampton Frost, combined epidemiology with mathematical models to establish Mathematical Epidemiology. William Ogilvy Kermack and Anderson Gray McKendrick [25] however, were responsible for the most widely accepted mathematical model to describe the progress of an epidemic, the *General Epidemic Model*. Based on that model and by using the following three differential equations, where N is the fixed population size, S is the number of the susceptible hosts, I is the number of the infected hosts, R is the number of the recovered, quarantined or deceased individuals, β is the *pairwise rate of infection*, γ is the removal rate and under certain assumptions such as the homogeneous mixing of the population, it is possible to depict accurately the circulation of a disease.

$$\frac{dS}{dt} = -\beta SI \tag{1}$$

$$\frac{dI}{dt} = \beta SI - \gamma I \tag{2}$$

$$\frac{dR}{dt} = \gamma I \tag{3}$$

given that the population size is constant

$$N = S(t) + I(t) + R(t) \tag{4}$$

In our effort to correlate the variables of this model, which is also known as S-I-R (Susceptible-Infective-Recovered) model, with the physical quantities of an epidemic, we will find striking similarities between the spread of biological viruses and the propagation of computer worms. Kephart was the first, who in his seminal work introduced McKendrick's epidemiological models to describe the spread of computer viruses [23, 24]. While he is the founder of computer epidemiology at that time the propagation speed of malicious code did not constitute a major threat. It was only shortly after the malware epidemics of Code Red, Code Red II and Nimda that it was made clear that traditional approaches to protect against malicious code, were no longer sufficient. Hence, Staniford et al [44] started to investigate worms' propagation dynamics under the prism of epidemiology with remarkable success. Since then, a lot of effort has been put into the improvement and finetuning of these models [55].

The following interpretation of biological epidemiology in a computer network context is our own and may only slightly differ from other established approaches, but we believe that are closely related and can sufficiently explain the three essential ingredients of worms' effectiveness, which we presented in the first part of the paper.

- N: the fixed population size. In computer epidemiology, it is usually the total number of hosts connected to the Internet, if the spread of a given worm is to be examined.
- S: the number of the susceptible hosts. In our context this means computers running the application or operating system that the virus targets. As

discussed in the third section of this paper, the more prevalent an operating system or an application is, the more likely to get exploited in case of an vulnerability and the sooner the susceptible population will become infected. Therefore, the diversity in our digital infrastructure is not an unnecessary luxury, but an essential precaution.

- I: the number of the infected hosts. Our collective efforts should focus on minimizing that set.
- R: the number of the recovered, quarantined or deceased individuals. In a malware epidemic R represents patched or well hardened systems, resilient to the exploited vulnerabilities. It is clearly to our best interest to convince users to keep their systems secured and updated and thus to have R maximized. As the age of an vulnerability decreases, is more difficult to have the majority of systems updated. Moreover, if a worm utilizes a *zero day* exploit, the only way to increase R is to rely on external security mechanisms, such as firewalls, in the hope that way a malcode attack can be intercepted. Of course, there is also another aspect of R. In a similar way to the biological death of some part of the population due to a pathogen, also some computer systems can be damaged from a destructive worm. Therefore a super virulent worm might face significant challenges to its further propagation.
- β: the *pairwise rate of infection*. The larger β this is, the more rapidly a worm spreads. In order to increase β malcode writers employ, usually intuitively, various techniques. Characteristic examples are the spawning of multiple threads of the target locator as in the case of the Code Red Worm or fitting the whole worm code in a single UDP packet to eliminate TCP connection latency [32].
- γ: the removal rate, either via disinfection, isolation or death in the physical world. During a malware epidemic a large γ would obviously help the containment of a worm. This can be attributed to either an effective mechanism to timely provide patches to vulnerable systems or to a very destructive payload. Contrary to the common belief, a very harmful worm could hinder its further propagation leading to its extinction.

Another important parameter that does not appear directly to the aforementioned equations is ρ the *relative removal rate* which is defined as

$$\rho \equiv \frac{\gamma}{\beta} \tag{5}$$

An epidemic outbreak is possible only when the number of initial susceptible population $S_0 > \rho$.

Of course that depends also on the underlying network topology, as useful theoretical studies have indicated [27] with implicit implications for scale-free graphs [36], which represent the majority of most technical and technosocial networks [2, 5, 13, 12]. The developments in computer epidemiology allow us to understand, model and accurately predict the spread of malicious software, which is necessary for the implementation of effective network defenses and automatic containment mechanisms capable to suppress its propagation in the available time frame.

7 Conclusion

Given the dependency of modern societies on digital infrastructures the rapid malcode is a serious problem. The most advanced nations strive to implement effective cyber-defences against the new generation of malware-based threats. The development of malcode detection algorithms that are applicable to anti-virus programs or host-based intrusion detection systems have proven useful, but inadequate to contain rapidly spreading malware epidemics. The microscopic analysis is nonetheless essential to disinfect or protect a system, once a worm has gained access to it. On the other hand to secure the operational availability of critical information, communications, and control systems a strategic approach is required. In medicine, microbiologists and epidemiologists act complementary to ensure timely identification of new threats and provide the best possible protection of the susceptible population. In our domain a similar methodology should be applied to fight efficiently digital threats.

References

1. U. Aickelin and J. Greensmith. Sensing danger: Innate immunology for intrusion detection. *Information Security Technical Report*, 12:218–227, 2007.
2. R. Albert and A. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, January 2002.
3. I. Arce and E. Levy. An analysis of the slapper worm. *IEEE Security & Privacy*, 1(3):82–87, January 2003.
4. A. Arora and R. Telang. Economics of software vulnerability disclosure. *IEEE Security & Privacy*, 3(1):20–25, January 2005.
5. A. Barabási, R. Albert, and H. Jeong. Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A*, 281:69–77, 1999.
6. J. Barry. *The Great Influenza*. Penguin Group, New York, NY, 2005.
7. M. Burgess. Probabilistic anomaly detection in distributed computer networs. *Science of Computer Programming*, 1:1–26, 2006.
8. M. Burgess. Biology, immunology and information security. *Information Security Technical Reports*, 12:192–199, 2007.
9. F. Cohen. Computer viruses – theory and experiments. *Computers and Security*, 6:22–35, 1987.
10. F. Cohen. *A Short Course on Computer Viruses*. Wiley Professional Computing. Wiley, Canada, 1994.
11. DangerProject. The danger project. Current on-line (September 2008): http://http://www.dangertheory.com/, September 2008.
12. H. Ebel, L. Mielsch, and S. Bornloldt. Scale-free topology of e-mail networks. *Physical Review*, E 66(035103(R)), September 2002.
13. M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *Proceedings of ACM SIGCOMM*, pages 251–262, Cambridge, MA, USA, 1999.
14. S. Forrest, S. Hofmeyr, and A. Somayaji. Computer immunology. *Communications of the ACM*, 40(10):88–96, 1997.
15. S. Forrest, A. Somayaji, and D. Ackley. Building diverse computer systems. In *IEEE 6th Workshop on Hot Topics in Operating Systems,*, 1997.

16. S. Furnell and J. Ward. The true computer parasite. http://securityfocus.com/infocus/1838, June 2005.
17. D. Geer. Monopoly considered harmful. *IEEE Security & Privacy*, 1(6):14–16, Dec 2003.
18. D. Geer, R. Bace, P. Gutmann, P. Metzger, C.P. Pfleeger, J.S. Quarterman, and B. Schneier. Cyber insecurity: The cost of monopoly. Technical report, Computer & Communications Industry Association, 2003.
19. S. Goel and S. Bush. Biological models of security for virus propagation in computer networks. *;login:*, 29(6), December 2004.
20. G. Goth. Addressing the monoculture. *IEEE Security & Privacy*, 1(6):8–10, Dec 2003.
21. J. Greensmith and U. Aickelin. The deterministic dendritic cell algorithm. In *7th International Conference on Artificial Immune Systems (ICARIS2008)*, Phuket, Thailand, 2008.
22. S. Hofmeyr. On the virulence of malware. Current on-line (June 2007): http://www.nthworld.org/archives/malware/index.htm.
23. J. Kephart. How topology affects population dynamics. In *Proceedings of Artificial Life 3*, New Mexico,USA, June 1992.
24. J. Kephart and S. White. Directed-graph epidemiological models of computer viruses. In *Proceedings of the 1991 Computer Society Symposium on Research in Security and Privacy, California USA*, pages 343–361, 1991.
25. W. O. Kermack and A. G. McKendrick. A contribution to the mathematical theory of epidemics. In *Proceedings of the Royal Society of London. Series A*, volume 115, pages 700–721, 1927.
26. H. Kim and I. Kang. On the functional validity of the worm-killing worm. In *Proceedings of the 2004 IEEE International Conference on Communications*, volume 4, pages 1902– 1906, June 2004.
27. J. Leveille. Epidemic spreading in technological networks. Hpl-2002-287, School of Cognitive and Computing Sciences, University of Sussex at Brighton, Bristol, October 2002.
28. J. Li and P. Knickerbocker. Functional similarities between computer worms and bilogical pathogens. *Computers & Security*, 26:338–347, 2007.
29. P. Matzinger. The danger model: A renewed sense of self. *Science*, 296:301–305, April 2002.
30. MedicineNet. Definition of virulence. Current on-line (December 2008): http://www.medterms.com/script/main/art.asp?articlekey=6911, 2008.
31. R. Medzhitov and C Janeway. Decoding the patterns of self and nonself by the innate immune system. *Science*, 296:298–300, April 2002.
32. D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the slammer worm. *IEEE Security & Privacy*, pages 33–39, July 2003.
33. N.Weaver and D. Ellis. White worms don't work. *;login*, 31:33–38, 2006.
34. T. Okamoto and Y. Ishida. A distributed approach against computer viruses inspired by the immune system. *IEICE Transaction on Communications*, E83-B:908–915, May 2000.
35. OneStat. Microsoft's windows os global market share is more than 97to onestat.com Current on-line (May 2005): http://www.onestat.com/html/press-release-microsoft-windows-vista-global-usage-share-december-2008.html, 2008.
36. R. Pastor-Satorras and A. Vespignani. Epidemic spreading in scale-free networks. *Physical Review Letters*, 86:3200–3203, 2001.

37. S. Pattyn, editor. *Ebola Virus Haemorrhagic Fever*. Elsevier North-Holland, 1977.
38. J. Pincus and B. Baker. Beyond stack smashing: Recent advances in exploiting buffer overruns. *IEEE Security & Privacy*, 2(4):20–27, July 2004.
39. E. Rescorla. Is finding security holes a good idea? *IEEE Security & Privacy*, 3(1):14–19, January 2005.
40. K. Shafi and H. Abbass. Biologically-inspired complex adatpive systems approaches to network intrusion detection. *Information Security Technical Report*, 12:209–217, 2007.
41. C. Shannon and D. Moore. The spread of the witty worm. *IEEE Security & Privacy*, 2(4):46–50, July 2004.
42. A. Somayaji and S. Forrest. Automated response using system-call delay. In *Nith USENIX security symposium*, 2000.
43. A. Somayaji, S. Hofmeyr, and S. Forrest. Principles of a computer immune system. In *Meeting on New Security Paradigms, 23-26 Sept. 1997, Langdale, UK*, pages 75–82. New York, NY, USA : ACM, 1998, 1997.
44. S. Staniford, V. Paxson, and N. Weaver. How to 0wn the internet in your spare time. In *Proceedings of the 11th USENIX Security Symposium*, pages 149–167, August 2002.
45. P. Szor. *The Art of Computer Virus Research and Defense*. Addison-Wesley, Upper Saddle River, NJ, February 2005.
46. S. Tanachaiwiwat and A. Helmy. Modeling and analysis of worm interactions (war of the worms). In *Fourth International Conference on Broadband Communications, Networks and Systems, 2007. BROADNETS 2007.*, pages 649–658, 2007.
47. M. Sabelis U. Dieckmann, J. Metz and K. Sigmund, editors. *Adatpive Studies in Dynamics of Infectious Diseases*. Cambridge University Press, 2002.
48. V. Vlachos, S. Androutsellis-Theotokis, and D. Spinellis. Security applications of peer-to-peer networks. *Comput. Networks*, 45(2):195–205, 2004.
49. V. Vlachos, A. Raptis, and D. Spinellis. PROMISing steps towards computer hygiene. In Steven Furnel, editor, *International Network Conference (INC2006)*, pages 229–236, Plymouth, UK, July 2006.
50. V. Vlachos and D. Spinellis. A PRoactive malware identification system based on the computer hygiene principles. *Information Management and Computer Security*, 15(4):295–312, 2007.
51. N. Weaver, V. Paxson, and S. Staniford. A worst-case worm. In *Proceedings of the Third Annual Workshop on Economics and Information Security (WEIS04)*, May 2004.
52. N. Weaver, V. Paxson, S. Staniford, and R. Cunningham. Large scale malicious code: A research agenda. Current on-line (June 2005): http://www.cs.berkeley.edu/ nweaver/large scale malicious code.pdf, May 2003.
53. P.D. Williams and T. Day. Interactions between mortality sources and the evolution of parasite virulence. In *Proceedings of the Royal Society of London B*, volume 268, pages 2331–2337, 2001.
54. K. Zelonis. Avoiding the cyber pandemic: A public health approach to preventing malware propagation. Master's thesis, Carnegie Mellon University, December 2004.
55. C. Zou, W. Gong, and D. Towsley. Code red worm propagation modeling and analysis. In *Proceedings of the 9th ACM Conference on Computer and Communication Security (CCS)*, Washington DC, USA, November 2002.