PROMISing Steps Towards Computer Hygiene

V. Vlachos, A. Raptis, and D. Spinellis

Department of Management Science and Technology,
Athens University of Economics and Business (AUEB),
Patission 76, GR-104 34, Athens, Greece
email: vbill@aueb.gr

Abstract

Recent worm epidemics proved beyond any doubt that the existing centralized worm containment mechanisms are no longer adequate to protect vulnerable systems, resulting in a shift towards distributed cooperative systems that aim to safeguard and immunize the susceptible population by automatically vaccinating them. We present PROMIS, a P2P based algorithm that provides its participants with early information regarding the existence of a worm epidemic and lets them automatically adjust their security level. We argue that our approach is based on the principles of hygiene: taking the basic precautions to avoid infection when an epidemic is on the rise.

> *The ultimate security is your understanding of reality*
> —H. Stanley Judd

Keywords

Peer to Peer Networks, Computer Worms, Computer Hygiene

1 Introduction

Recent malware epidemics (Moore *et al.*, 2002; Zou *et al.*, 2002; Bailey *et al.*, 2005) demonstrated the enormous harm that modern worms can cause, but latest incidents also showed that new breeds of malcode appear almost concurrently with the announcement of new vulnerabilities (Shannon & Moore, 2004). Thus it is clear that traditional security applications such as firewalls, IDS or anti-viruses, while beneficial, no longer provide sufficient protection against rapidly spreading malware. Therefore, it is useful to explore other possible protective mechanisms that can act complementary to the existing security infrastructure. To design effective defenses for our digital assets it is essential to understand the propagation dynamics of fast spreading worms so as to identify the available time frames in which reaction is both feasible and effective. Staniford et al (Staniford *et al.*, 2002) proved that highly virulent worms are fully capable of infecting the susceptible population in less than 15 minutes, as in the case of a Warhol worm. Empirical evidence confirms the validity of these assumptions: the Slammer worm (Moore *et al.*, 2003) required only 10 minutes to infect the vulnerable population using the simplest propagation strategy (random scanning), while the theoretical limits of an ultra virulent worm fall well below the psychological one minute limit (Weaver *et al.*, 2004; Staniford *et al.*, 2004).

To alleviate the effects of rapid malcode we propose a cooperative containment algorithm based on the following assumptions.

1. The most vulnerable systems are personal computers whose respective owners do not have the time nor the skills to protect them sufficiently. In other words their owners are not security professionals that are constantly aware of the trends of malware activity. The best that is reasonable to expect from them is

   (a) To have enabled the automatic downloading and installation of the new updates and patches on their OS of choice (usually some flavor of Windows).

   (b) To have installed some kind of security application. Most probably an anti-virus with automatic updating of signatures and a roughly configured firewall. Experienced users or small offices with a minimal IT infrastructure may also host an Intrusion Detection System.

2. Fine grained security policies are perfectly suited for large organizations and enterprises that have valuable digital assets to protect in very complex environments with numerous different user groups, each one requiring different resources and access rights to perform their duty. On the other hand, small office or home office (SOHO) users can be sufficiently protected even with less detailed security policies. We argue that by simply changing different predefined security policies it is possible to hold at bay the attacks of most malware. As most attacks have common attack vectors, such as the HTML engine of a popular e-mail client or the scripting abilities of an equally popular browser, by disabling these services only during worm epidemics and re-enabling them after the containment of the epidemic, we can adequately protect them against most threats. SOHO users on the other hand, may agree with a temporary hardening of their system in order to protect themselves against a malcode epidemic, but they are unlikely to accept a permanent disabling of their useful but not necessary favorite services and applications.

Our algorithm is called PRoactive Malware Identification System — PROMIS and is based on a peer-to-peer (P2P) architecture to provide timely information to the members of a specially crafted P2P group.

2 Related Work

P2P networks are widely treated as a potential propagation vector for malicious software. While many worms or viruses so far utilized P2P networks to accelerate their propagation, mostly masquerading as pirated software or media files, we do believe on the contrary that their distributed architecture can offer significant advantages over the traditional centralized or partially centralized architectures. Our research has been greatly influenced by Kephart et al (Kephart *et al.*, 1993; Kephart, 1992; Kephart & White, 1999), who introduced in his seminal work the concept of computer epidemiology by applying the basic epidemiological models to computer viruses.

This work shares more common ground with the Indra project (Janakiraman *et al.*, 2003) and the quarantine reputation-based system of Coull et al (Coull & Szymansky, 2005). Indra's philosophy is quite similar to ours, but with one major difference. Our goal is to give to all participants of the PROMIS system the rate of the ongoing malicious activity and let them decide for the best applicable measures, while the Indra project aims to inform all the participants about specific threats as well as the origin of these threats in order to have them blacklisted. Coull's architecture is also based on a P2P framework, but they work at a router level and each node acts to protect the community in general, where in our case each node aims to protect itself, which diminishes the possibilities of Byzantine situations where malevolent nodes try to use innocent nodes to harm others. We find extremely useful the information provided by DSHIELD (DShield Corporation Web Site, 2006) in which numerous clients submit data from their firewalls and IDS, that are collected and analyzed by a central server to derive attack trends and rates. We could say metaphorically that our design is a fully decentralized DSHIELD in a more general and simple form. Another system that aggregates data from thousands of clients and extracts global attack rates to inform a special members group is the DeepSight (Symantec Corporation Web Site, 2006) system, but as a commercial service it does not provide all the required information to evaluate it properly.

3 Architecture

PROMIS utilizes a P2P architecture (Figure 1). We assume that a special purpose security peer group, named PROMISGROUP, is created. PROMISGROUP contains two types of nodes, the *member* nodes and the *super* nodes. All normal nodes wishing to participate to this P2P group must authenticate themselves to one of the available super nodes. We consider that the authentication procedure takes place using secure out of bounds communication mechanisms. The super-nodes verify all the submitted data of a requesting node before authenticating it. This data may also include the real names, e-mail address and a phone/FAX numbers of the node's owner. Thus all member nodes of the PROMISGROUP are not intentionally malevolent, and more importantly, they can later be contacted if their behavior is unexpected or abnormal. The authentication procedure is outside the scope of our algorithm, as there are a number of excellent trust management schemes for P2P networks available in the literature (Androutsellis-Theotokis & Spinellis, 2004). We do provide, however some very simple mechanisms to exclude misbehaving nodes. We also require from every participating node to host a security application as an anti-virus, firewall or an Intrusion Detection System in order to contribute to the more accurate estimation of the general malware activity, though it is possible for a node to gain from our system even if it doesn't operate any security application.

PROMIS nodes constantly perform two operations (Figure 2). A daemon called Notifier checks repeatedly during predefined short time intervals the log files of the security applications operating on the specific node and extracts the rate of the intercepted malicious activity against this host according to the following formula

$$p_t^n = \frac{h_t^n - \frac{\sum_{i=t-k}^{t-1} h_i^n}{k}}{\frac{\sum_{i=t-k}^{t-1} h_i^n}{k}} \tag{1}$$

where t is the ordinal number of a fixed time interval, n is the node identifier, h_t^n is the number of attacks node n received in the interval t, p_t^n is the percentage increase or decrease in attacks during the current interval t on node n, $k(> 0)$ is the size of the 'time window' used in the number of t time intervals which the malicious activity rate is calculated. The Notifier also sends this local malicious activity rate to a number of randomly chosen participants of the PROMISGROUP.

Another daemon named Handler constantly listens for incoming rates from other peers of the PROMISGROUP and aggregates their messages in order to compute a global malicious activity rate for the PROMISGROUP using the following equation.

$$p_{avg} = \frac{\sum_{i=1}^{n} p_t^i}{n} \tag{2}$$

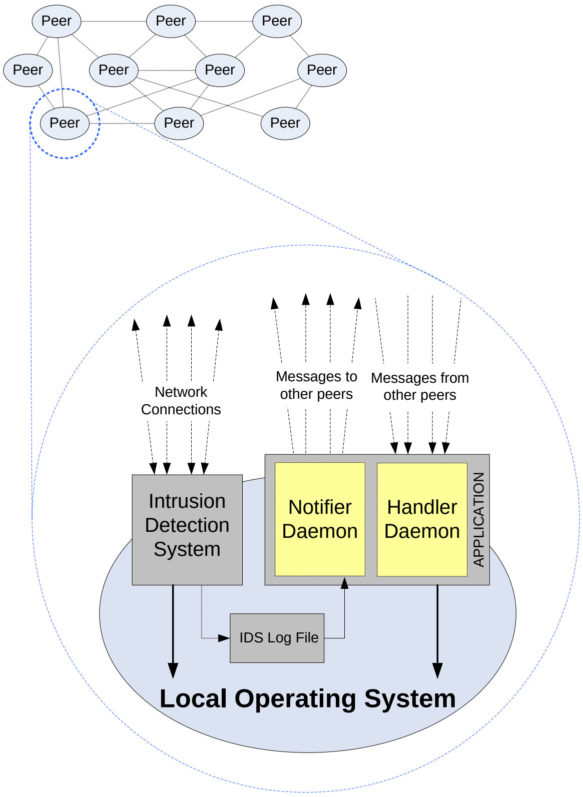

Figure 1: PROMIS architecture

The Handler's main responsibility is to adjust automatically the security level of the local system based on the subsequent directives

- if $p_{avg} > t_{high}$, then increase the security policy by disabling non essential services as for example HTML preview in mail clients or by increasing the security settings of the installed web browser.
- if $p_{avg} < t_{low}$, then decrease the security policy by reactivating the above-mentioned services.
- if $t_{low} \leq p_{avg} \leq t_{high}$ do nothing.

4 Implementation Details

The outcome of a simulation is highly dependent on the graph models that are used to depict the network topology. Therefore, we have developed and made available the NGCE (Vlachos *et al.*, 2005) tool that constructs the most appropriate graphs for the study of the spread of viruses and worms. To check the validity of our results we modeled the uncontrolled propagation of various worms in different homogenous graph environments and compared the results of our simulator with the expected analytical solution of the *General Epidemic Model* (Kermack & McKendrick, 1927).

PROMIS simulator is so far capable of modeling the spread of a worm or a virus using the well known S-I-R (Susceptible-Infected-Recovered) model. The outcome of the simulator depicts in gnuplot-ready files (Figure 3):

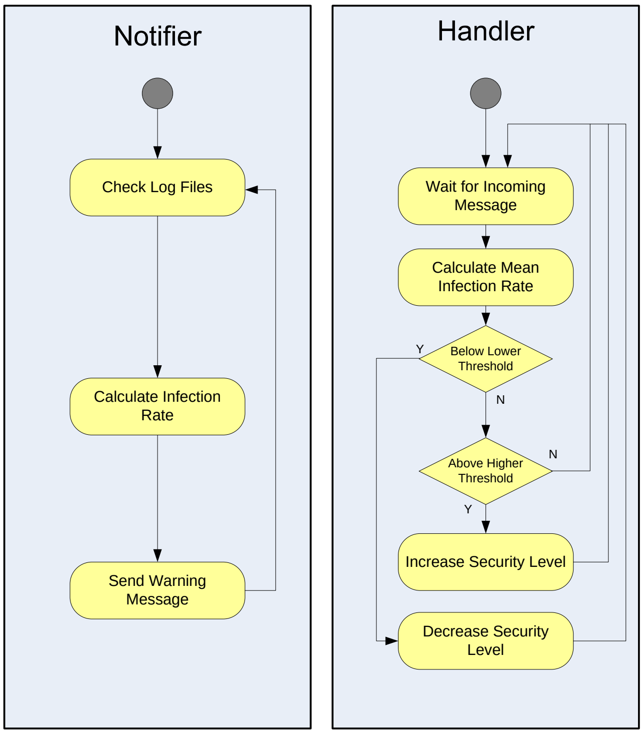

Figure 2: Handler and Notifier activity diagrams

- the percantage of the infected population.
- the epidemic curve, which is in essence the rate of new infections (Daley & Gani, 1999).
- the average locally intercepted malicious activity of any node.
- the average global malicious activity as it was estimated by each node of the PROMISGROUP.

The PROMIS simulator is written completely in Java and operates on graphs that have been generated with the NGCE, which so far covers homogeneous graphs, scale-free graphs, random graphs, lattices and custom graphs with specific properties.

5 Future Work and Concluding Discussion

Our results indicate that each peer of the PROMISGROUP observes a significant increase of the local malicious activity rate during the early phases of the epidemic (Figure 3). This information is dissiminated via the PROMIS system to each member of the PROMISGROUP, which in turn calculates the estimated global malicious activity rate of the peer group. Each peer has a different view of the malicious activity as it sustains a different number of attacks, communicates with a divergent set of nodes and therefore calculates its own local and global malicious activity rate. Therefore, in a non-homogeneous graph, such as the scale-free graph, which we used to perform our simulations, each peer has its own perspective of the local and the global malicious activity which might differ slightly from those of other nodes. The curves in Figure 3 depict the average local malicious activity of all the uninfected nodes and the global malicious activity rate of all active PROMISGROUP nodes respectively. The two horizontal lines indicate the total

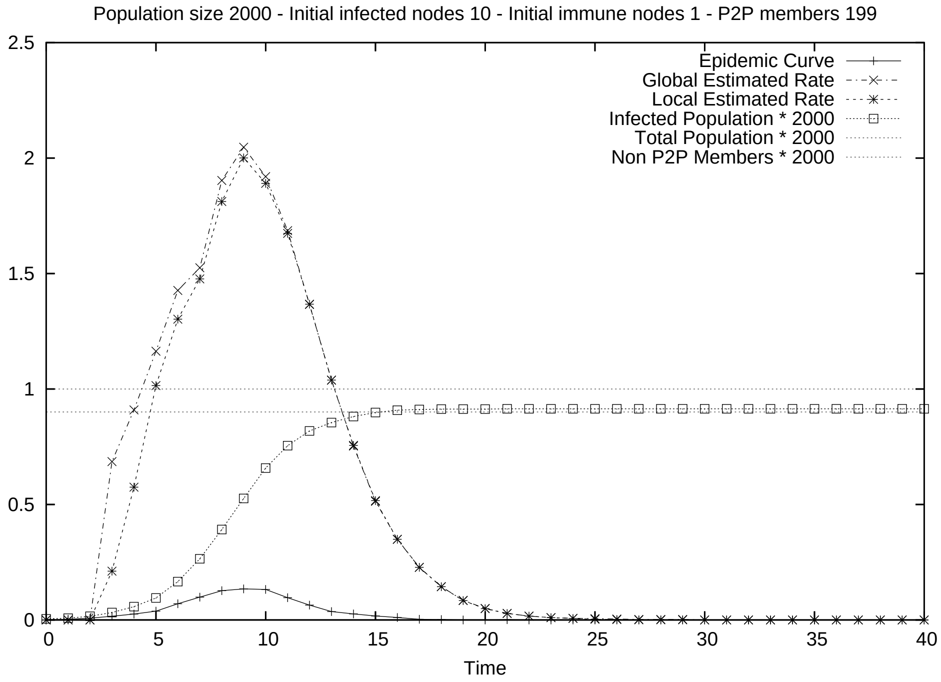

Figure 3: Simulator's output

population and the number of the PROMISGROUP members. As one can see from the graph, only a small percentage of the PROMISGROUP members has actually been infected. Figure 4 shows the transition of the security levels of the PROMISGROUP nodes that remain uninfected during the course of the epidemic.

It is reasonable to expect that by enabling the proper countermeasures during a real-world epidemic the majority of the participating systems will avoid an infection. We have built a proof of concept prototype (Vlachos *et al.*, 2004), but we haven't yet be able to develop a fully functional PROMIS system. Nonetheless we are confident that its realization is possible as all the required technologies such as mature P2P APIs and widely-installed security applications are already available. Having completed the PROMIS simulator the next step is to test the PROMIS algorithm extensively under different circumstances and to evaluate other possible ways to calculate the global malicious activity before we develop a large scale PROMIS system.

PROMIS was designed with two things in mind. First that the spread of the recent worms cannot be suppressed using traditional centralized containment techniques, thus a highly distributed environment based on the P2P networks might be useful, and second that the overwhelming majority of most users do not want the remote installation of any kind of code to their systems from anyone besides the original vendor of their software. Therefore we find all automatic immunization and vaccination systems (Goldenberg *et al.*, 2005) or the concept of *good worms* (Kim & Kang, 2004; Middleton, 2001) are not an applicable solution for the moment. Our intention is not to provide automatic protection for all the existent systems, rather we are confident that by applying basic precautions during a worm epidemic we can achieve significant benefits. During a biological epidemic of an infectious pathogen the first, but most critical line

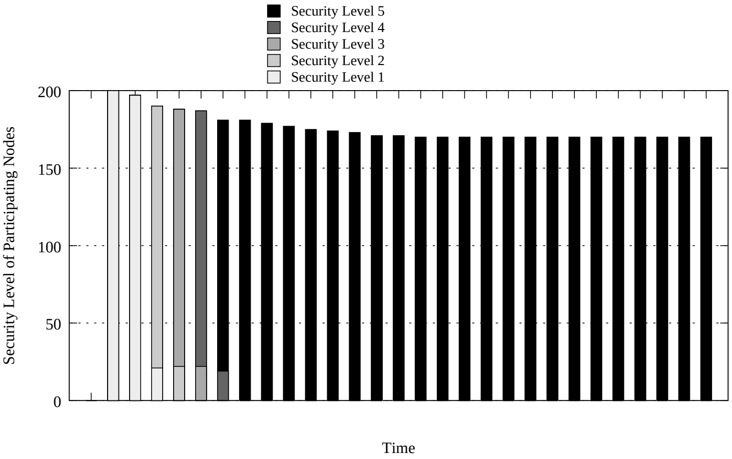

Figure 4: PROMISGROUP's peers security levels

of defense is the strict application of hygiene principles; we don't see a reason why computer epidemics should be treated differently.

Acknowledgments

The work of Vasileios Vlachos is funded under the Iraklitos fellowships for research of Athens University of Economics and Business program and the European Social Fund and partially funded by the European Commission (SQO-OSS IST project 2005-33331).

References

Androutsellis-Theotokis, S., & Spinellis, D. 2004. A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys*, **36**(4), 335–371.

Bailey, M., Cooke, E., Jahanian, F., Watson, D., & Nazario, J. 2005. The Blaster Worm: Then and Now. *IEEE Security & Privacy*, **3**(4), 26–31.

Coull, S., & Szymansky, B. 2005. *A Reputation-based System for the Quarantine of Widespread Malicious Behaviour*. Tech. rept. 05-01. Rensselaer Polytechnic Institute.

Daley, D., & Gani, J. 1999. *Epidemic Modelling*. Cambridge, UK: Cambridge University Press.

DShield Corporation Web Site. 2006. *DShield Distributed Intrusion Detection System*. http://www.dshield.org/. (Accessed March 2006).

Goldenberg, J., Shavitt, Y., Shir, E., & Solomon, S. 2005. Distributive immunization of networks against viruses using the 'honey-pot' architecture. *Nature Physics*, **1**, 184–188.

Janakiraman, R., Waldvogel, M., & Zhang, Q. 2003 (June). Indra: A peer-to-peer aroach to network intrusion detection and prevention. *In: Proceedings of 2003 IEEE WET ICE Workshop on Enterprise Security*.

Kephart, J. 1992 (June). How topology affects population dynamics. *In: Proceedings of Artificial Life 3*.

Kephart, J., & White, S. 1999 (May). Measuring and Modeling Computer Virus Prevalence. *Pages 2–14 of: Proceedings of the 1999 IEEE Computer Society Symposium on Research in Security and Privacy*.

Kephart, J., Chess, D., & White, S. 1993. Computers and Epidemiology. *IEEE Spectrum*, **30**(20).

Kermack, W. O., & McKendrick, A. G. 1927. A Contribution to the Mathematical Theory of Epidemics. *Pages 700–721 of: Proceedings of the Royal Society of London. Series A*, vol. 115.

Kim, H., & Kang, I. 2004 (June). On the functional validity of the worm-killing worm. *Pages 1902–1906 of: Proceedings of the 2004 IEEE International Conference on Communications*, vol. 4.

Middleton, J. 2001 (September). *Anti-worms' fight off Code Red threat*. http://www.vnunet.com/News/1125206. (Accessed November 2005).

Moore, D., Paxson, V., Savage, S., Shannon, C., Staniford, S., & Weaver, N. 2003. Inside the Slammer Worm. *IEEE Security & Privacy*, July 2003, 33–39.

Moore, D., Shannon, C., & Brown, J. 2002. Code Red: a case study on the spread and victims of an Internet worm. *In: Proceedings of the Internet Measurement Workshop.*

Shannon, C., & Moore, D. 2004. The Spread of the Witty Worm. *IEEE Security & Privacy*, **2**(4), 46–50.

Staniford, S., Paxson, V., & Weaver, N. 2002 (August). How to 0wn the Internet in Your Spare Time. *Pages 149–167 of: Proceedings of the 11th USENIX Security Symposium.*

Staniford, S., Moore, D., Paxson, V., & Weaver, N. 2004. The top speed of flash worms. *Pages 33–42 of: WORM '04: Proceedings of the 2004 ACM workshop on Rapid malcode.* New York, NY, USA: ACM Press.

Symantec Corporation Web Site. 2006. *Symantec DeepSight Threat Management System*. http://tms.symantec.com/. (Accessed March 2006).

Vlachos, V., Vouzi, V., Chatziantoniou, D., & Spinellis, D. 2005. NGCE network graphs for computer epidemiologists. *Pages 672–683 of:* Bozanis, Panagiotis, & Houstis, Elias N. (eds), *In Advances in Informatics: 10th Panhellenic Conference on Informatics, PCI 2005, Lecture Notes in Computer Science 3746.* Springer-Verlag.

Vlachos, V., Androutsellis-Theotokis, S., & Spinellis, D. 2004. Security applications of peer-to-peer networks. *Comput. Networks*, **45**(2), 195–205.

Weaver, N., Paxson, V., & Staniford, S. 2004 (May). A Worst-Case Worm. *In: Proceedings of the Third Annual Workshop on Economics and Information Security (WEIS04).*

Zou, C., Gong, W., & Towsley, D. 2002 (November). Code Red Worm Propagation Modeling and Analysis. *In: Proceedings of the 9th ACM Conference on Computer and Communication Security (CCS).*