

Email Worm Mitigation by Controlling the Name Server Response Rate

Nikolaos Chatzis
Fraunhofer FOKUS,
Kaiserin-Augusta-Allee 31,
10589 Berlin, Germany
nikolaos.chatzis@fokus.fraunhofer.de

Enric Pujol
Fraunhofer FOKUS,
Kaiserin-Augusta-Allee 31,
10589 Berlin, Germany
enric.pujol@fokus.fraunhofer.de

Abstract

Email worms and the spam associated with them are one of the main operational security issues today because they waste time, money and resources. The high incidence of email worms today clearly indicates that current network defence mechanisms yield rather meagre results in mitigating this class of self-propagating malicious program. In this work, we build on the observation that email worms rely on the local name servers to propagate, and propose a novel approach to slow down their propagation by means of limiting the response rate of local name servers, which are topologically near the infected user machines. We conduct extensive simulation experiments that involve email network, email user behaviour, email propagation, and physical network modelling, and show that our approach is promising for slowing down email worm epidemics.

1 Introduction

Email worms remain for network operators and end users an ever-evolving threat. They are the dominant method for propagating Trojans, spyware, and phishing attempts [3] and cause network congestion, which results in loss of service or degradation in the performance of network resources [1]. According to the Messaging Anti-Abuse Working Group (MAAWG) abusive email traffic traversing the Internet oscillates above 80% [2]. Rather than exploiting new critical vulnerabilities, email worms mainly rely on social engineering to infect a user machine. Social engineering is a non

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 216585 (INTERSECTION Project).

technical kind of intrusion heavily dependent on human interaction, which often involves victimizing users to break normal security procedures. After infecting a user machine, email worms exploit information found on the machine about the email network of the user, and then to avoid the ever improving filtering on the Simple Mail Transfer Protocol (SMTP) servers they use their built-in SMTP engine to send email messages to the potential victims.

Commonly applied approaches against email worms are antivirus, antispam packages and rule based email filtering, which are deployed on the receiver's domain, and operate on incoming messages. The rationale of these approaches is to classify email messages as malicious or legitimate, and deliver to users only messages that are deemed safe. These approaches suffer from the well-cited disadvantage of poor detection time against novel email worm instances because signatures and rules must be continually updated, which entails non-trivial human labour that results in significant delay. Furthermore, they detect abusive email traffic in the network of the potential victim and go to no lengths to slowing down email worm epidemics. Effectively slowing down email worm epidemics can contribute to reducing the unwanted email traffic traversing the Internet and can result in giving to humans the time to take the necessary actions – e.g. produce a new signature and disseminate it and re-configure routers and firewalls – to contain email worms.

In the past much research effort has been devoted to analyse the traffic that email worm infected user machines generate [20, 21, 12, 13, 10, 9, 5]. These studies share the outcome that once a user machine gets infected its Domain Name System (DNS) traffic characteristics change. The positive contribution of defining such a correlation between email worm infection and DNS traffic is that the communication patterns

between user machines and the local name server can serve as a basis for detecting email worm-infected user machines at the local name server, which is close to the user machines. Detecting email worms as they appear on local name servers offers several advantages. First, the email worm propagation can be slowed down and second, it contributes to reducing the unwanted email traffic traversing the Internet, which constitutes a primary operational security issue.

In this work we take a step beyond related work, and study the effect that controlling the response rate at which the local name server responds to an email worm-infected machine has on the email worm propagation. We show that by rate limiting the response rate of the local name server, the spread of infection and the extent of the epidemic can be reduced. We consider three rate limiting schemes i.e., *blackholing*, *user machine rate limiting* and *aggregate rate limiting*; and conduct extensive simulation experiments to evaluate their efficacy. Our simulation studies account for email network, email user behaviour, email propagation, and physical network modelling. The analysis we present in this paper is the first that we are aware of that offers evaluation of different DNS-based rate limiting schemes to contain email worm epidemics.

The paper is organized as follows. In Section 2, we discuss the necessary background of our study and related work. In Section 3, we explain our approach for slowing down email worm propagation by DNS response rate limiting. In Section 4, we validate our approach by conducting extensive simulation experiments. We present our conclusions in Section 5.

2 Background and Related Work

Our work is motivated by email worm detection by analysing DNS traffic, and draws on the confluence of email worm modelling and defence; we provide in this section the necessary background on these areas.

2.1 DNS-Based Email Worm Detection

A rich body of publications presents methods to detect email worm-infected user machines by analysing DNS traffic [20, 21, 12, 13, 9, 5]. The detection methods in [20, 21, 12, 13] are straightforward and focus on volume-based application-level detection. Their rationale is that many DNS queries for Mail eXchange (MX) resource records (RR) or the relative numbers of queries for pointer (PTR) RR, MX and address (A) RR a user machine generates give a telltale sign for email worm infection. In particular, the authors in [21] analyse measurements of the DNS traffic captured at the caching name server of a campus network during the

outbreak of SoBig.F and MyDoom.A. Similar measurements for the same worm instances present the authors in [12], while they extend their scope by studying Net-sky.Q and Mydoom.S in [13]. In [20] the authors focus on enterprise networks and measure DNS activity of NetSky.Q over ten minutes periods. In [9, 5] the authors propose more sophisticated approaches that build on anomaly detection. In [9] the authors detect email worm-infected user machines by means of Bayesian inference based on a priori knowledge of email worm signature DNS queries. Finally, the authors in [5] report on their unsupervised detection method, which builds on time series analysis of DNS streams at flow level.

2.2 Email Worm Life Cycle

User Machine Infection Email worms rely on a two-phase interaction with the email user to propagate. The first phase relates to the user behaviour in checking emails; the critical parameter is how often a user checks its emails. The second phase involves social engineering to infect a user machine once the abusive email reaches the inbox of a user. Social engineering techniques lure an email recipient to take some action that will infect its system. Email worms provoke the user to open a specially crafted file attachment or to visit a Web site, which installs malicious software on its machine.

The authors in [24, 25] are the only that provide a model for email user behaviour in the email worm modelling and defence literature. This model has been also used by the authors in [23]; therefore we opt to use this model in the work presented here. In particular, in [24] they model the mean email checking time of a user i as a Gaussian-distributed random variable T_i , i.e., $T_i \sim N(\mu_T, \sigma_T^2)$ and the probability P_i that the user gets tricked as a Gaussian-distributed random variable, i.e., $P_i \sim N(\mu_P, \sigma_P^2)$. They extend their model in [25] by defining P_i as $1 - (1 - C_i)^m$, where m is the number of received worm copies and C_i is the probability that the user opens the attachment. They model C_i as a Gaussian-distributed random variable i.e., $C_i \sim N(\mu_C, \sigma_C^2)$.

Email Worm Propagation Email worms self-propagate among social contacts that comprise the email network of the user. So, once an email worm infects a user machine it takes actions to discover the email network of the user. Email worms harvest email addresses from the email address book and files with predefined extensions. Worm writers also use other techniques that involve guessing email addresses to support email worm spreading. First, they arm worms with lists of commonly used names e.g. `webmaster@` or `admin@` to send out emails indiscriminately to any domain they find on the infected machine. Second, email

worms come equipped with specialized Web crawlers, which are automated software that browse Web pages visited by the user looking for email addresses.

Modelling email networks has attracted much research attention [24, 23, 14, 6]. The authors study email network as graphs, where email users are nodes and a vertex among two nodes represents that the two connected users know the email address of each other. In [14] the authors analyse the structure of the network formed by the email user address books using data from a university network. The authors in [6] look at email server log files at their university to find that the email network can be modelled as a scale-free network. Scale-free networks are characterized by power-law distribution of their node degree; node degree is the number of edges incident to the node. In [24] the authors extend the scope of the above models to account for email lists. They examine a large number of email groups in *Yahoo!* to find that their size is heavy-tailed distributed. Based on the observation that if a user has the address of an email group in his computer – from an email worm’s point of view – this user virtually has all the addresses contained in the email group, they argue that email networks are heavy-tailed distributed in terms of node degrees. The authors in [23] state that the node degree in [24] might be underestimated because email addresses can be stored in locations other than the user address book e.g., web browser cache. They call these additional addresses *external addresses* to distinguish them from *internal addresses* – i.e., the email addresses that are stored in the address book – and model them as a uniform distribution $U \sim (0, d_i)$, where d_i denotes the number of internal addresses. Thereby, they consider that the email network has a higher average node degree than this in [24].

Despite, this correction in the average node degree, the authors do not account for the capability of email worms to guess email addresses, which is today a common email worm functionality. In this work we go a step beyond their work and further increase the average node degree by using a simple model to describe guessing. Let d denote the total number of email addresses – both internal and external, as defined in [23] – of a user i , an email worm guesses $d_{ig} = d \times w_{agg}$ addresses; w_{agg} is the email worm *guessing aggressiveness*, which is a measure that quantifies the attempt of a worm to guess email addresses.

2.3 Email Worm defence

Given the analysis of email worm propagation among email contacts presented above, it becomes apparent that user machines infected by email worms do not exhibit abnormal packet-level patterns similar to

worms that scan a set of IP addresses to propagate. This makes detection methods that look for high rates at which unique destination addresses are contacted [18], or high number of failed connections [16], or high self-similarity of packet contents [19] intrinsically incapable of detecting email worms. Likewise, Honeypot-based systems provide a very reliable anti-scanning mechanism, but they are not effective against email worms. Commonly applied approaches for dealing with email worms like antivirus, antispam packages and rule based filtering at the desktops, network servers, mail exchange servers and at the gateways [8] share two deficiencies. First, they are ineffective against novel worm instances because they entail non-trivial human labour that results in significant delay, and second, they do not target reducing the amount of unwanted email traffic traversing the Internet.

This state of affairs stimulated researchers to look at other directions for defending the Internet against email worms. Inspired by work on node immunization to contain scanning worms [15], researchers have proposed adapting this work to contain email worms [25, 23]. These studies are based on the premise that the structure of the email network plays an important role in the epidemic propagation [11]. In particular, in a scale-free network the subset of individuals with higher number of email contacts get infected first, and the individuals with decreasing number of contacts follow [4]. Despite their positive contribution of showing that immunizing the high-degree nodes significantly slows down email worm propagation, selective immunization is not applicable because it relies on a priori knowledge of the email network structure, which in practice is not apparent. Furthermore, it depends on email worm vaccines i.e., signatures, which are developed after the worm has already infected the majority of susceptible user machines.

In this work we go a step beyond related work, and given that detection of email worm by analysing DNS traffic on the local name server is possible, we investigate how controlling DNS responses by means of rate limiting affects email worm propagation. In contrast to immunization rate limiting is a method that is deployable in real networking environments [22].

3 DNS Response Rate Limiting

The DNS is a critical infrastructural component of Internet, since it constitutes the essential first link in the entire chain of Internet connectivity. Name servers are specialized database servers that translate between domain names and their corresponding IP addresses and vice versa. They are present in the vast majority of networking environments and play a significant

role in Internet, since almost all basic and emerging applications depend on them to work. Email worms rely on querying the name servers for information of the domain name space to propagate. Firstly, guessing and Web crawling for email addresses are functionalities heavily dependent on querying the name server for A RRs. Secondly, email worms send out emails to potential victims using their build-in SMTP engine. However, to find the recipients' SMTP server, the worms query the local name server firstly for MX and then for the corresponding A RR.

That said, we present here a novel approach to slow down email worm propagation. We investigate how controlling by means of rate limiting local name server responses affects email worm propagation. The scope of this study is twofold. Assuming that email worm detection on the local name server is possible, we show first that controlling local name server responses slows down global email worm propagation once an email infected user machine has been identified. Second, we evaluate the relative impact of detection and rate limiting time, which hereafter we call mitigation time, on the email worm epidemics. Rate limiting is a mechanism by which an element in the network can restrict the rate of communication with other network elements. We consider three rate limiting schemes to modify the responses rate of local name servers: *blackholing*, *user machine rate limiting* and *aggregate rate limiting*.

Two time parameters are important for our study: the detection time and the mitigation time. We define detection time t_d as the time between when an infected user machine sends out its first infectious DNS query and when the name server identifies the user machine as infected, and mitigation time t_m , as the time within which the name server rate limits its outgoing DNS responses. Given these two time parameters, blackholing refers to not responding to incoming DNS queries from an infected user machine within t_m , user machine rate limiting to responding only to a limited percentage of queries within t_m from an infected user machine and aggregate rate limiting to restricting to a certain rate DNS responses to all infected user machines for time t_m . In our framework, the latter two approaches implement no buffering. After time t_m , the local name server responds again to user machines in a best effort manner. Since local name servers are critical infrastructure components, on which a large number of user applications depend, we opt to use these gentler rate limit alternatives that release rate limiting after t_m , and thus render our approach more palatable to actual deployment. In Fig. 1 we show an example of how blackholing and user machine rate limiting work.

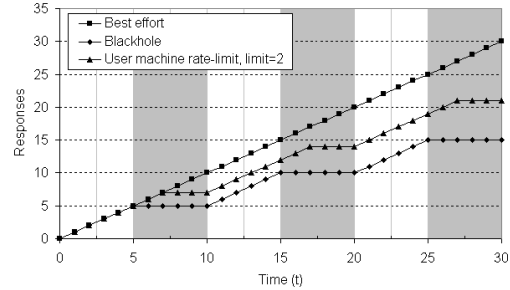


Figure 1. DNS responses sent to a single user machine, while rate limiting. After detection time $5t$, a rate limit scheme is applied for time $5t$ and then is released. Mitigation periods are highlighted.

4 Email Worm Simulation Studies

In this section we conduct extensive simulation studies of free email worm propagation, and show how it is affected by DNS response rate limiting schemes. We discuss the experimental setup, as well as our results.

4.1 Experimental Setup

In order to validate our hypothesis that email worm mitigation is possible through DNS-based response rate limiting, we conducted an extensive evaluation in Georgia Tech Network Simulator (GTNetS). GTNetS was designed for efficient simulation of large-scale networks, as it achieves good scalability by using distributed simulation methods, as well as an efficient design for both memory and computational resources [17]. Analogously to epidemiology studies that look at the number of infections by a contagious illness as a function of time, we use as performance measures the average number of infected user machines per time and the average number of active email worms per time. We derive these measures by averaging from 100 simulation runs for each rate limiting scheme.

In contrast to [25, 24], we account for the network physical topology. We consider underlying physical topology of 200 Campus Network instances (CN) connected in a slightly modified ring topology. This network topology model has been extensively used in other large-scale simulations [7]. Each CN comprises of 504 user machines, one name server and one SMTP server. Moreover, regarding the email network we opt to use for comparison reasons, the 100.000 nodes email network provided by the authors in [24]. This network is characterized by average node degree 8 and power-law exponent -1.7. Finally, we assign randomly email network nodes to physical user machines.

Email users check their email with a Poisson distribution, where the parameter λ is approximated by a Gaussian distribution $N \sim (40, 20^2)$. Moreover, a user

Table 1. Experimental Parameters

Parameter	Value	Explanation
k	-1.7	Power law exponent
d_i	x^{-k}	# of internal addresses
d_e	$[0 - d_i]$	# of external addresses
μ_T	40	Mean email checking time
σ_T^2	20^2	Variance of email checking
μ_P	0.5	Mean opening probability
σ_P^2	0.3^2	Variance of opening probability
w_{agg}	0.5	Email worm guessing aggressiveness
$I(t=0)$	2	# of initially infected nodes
$D_i(t=0)$	8	Initially infected nodes degree
a_q	$[0 - 20]$	# of additional queries per successful query

i gets infected with probability $1 - (1 - C_i)^m$, where m is the number of received worm copies and C_i is obtained from a Gaussian distribution $P \sim (0.5, 0.3^2)$. The number of external addresses d_e for a user i is obtained from a uniform distribution $U \sim (0, d_i)$, where d_i is the number of internal addresses. We assume that a worm has the capability of guessing a number of email addresses d_{ig} equal to $w_{agg} \times (d_i + d_e)$, where we set the guessing aggressiveness to 0.5. Finally, for guessing we assume that for every successful query the worm performs a_q additional queries, where a_q is obtained from a uniform distribution $U \sim (0, 20)$.

Email users use the local name server and the mail server within their CN. Mail servers do not support any kind of email filtering, and local name servers support mitigation schemes. Both for user machine and aggregate rate limiting within the mitigation time t_m , the local name servers respond to 10% of the received queries. Regarding the initial infected user machines, we randomly select two user machines from the set of user machines, which have node degree equal to the average node degree of the email network. Thereby, we avoid selecting high-degree or low-degree nodes that can bias our simulation results. In Table 1 we list our experimental parameters and their values.

4.2 Evaluation of DNS Response Rate Limiting

In epidemiology the graph that shows the evolution of the average number of infected nodes in time is called *epidemic curve*. In Fig. 2 we present the epidemic curves that display the magnitude of the email worm outbreak for free email worm propagation, and how this is affected, when various rate limit schemes are applied on the local name server responses. The shape of the epidemic curves are consistent with the results in [24, 23]. Fig. 2 shows that the epidemic goes through two distinct phases. In the early phase, the number of infectives is a small fraction of the population, and the growth is approximately exponential. In

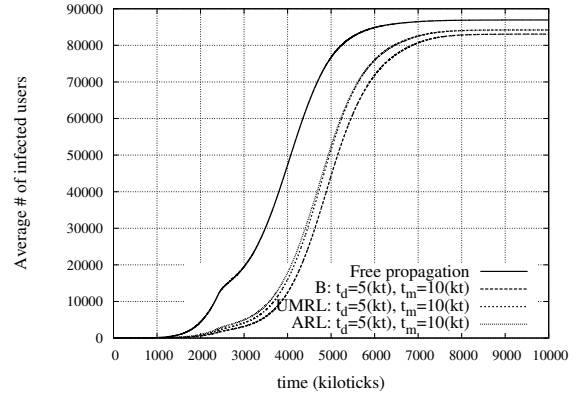


Figure 2. Epidemic curves comparing blackholing (B), user machine rate limit (UMRL) and aggregate rate limit (ARL). All three schemes slow down the propagation and reduce the number of user machines the email worm can reach.

the later phase, the rate of spreading slows down until it stabilizes as infectives saturate the population.

A closer look at Fig. 2 indicates that even if DNS responses are rate limited an epidemic takes place. This is because email worms repeatedly query local name servers until they eventually get the information about all potential victims. However, in Fig. 2 the shift to the right of the epidemic curves that correspond to the propagation in the presence of rate limiting reveals that blackholing, user machine rate limiting and aggregate rate limiting can reduce the propagation speed of email worms. Furthermore, rate limiting DNS responses reduces in the early phase the slope of the epidemic curve, which measures how the propagation accelerates. Finally, Fig. 2 shows that in the later phase the epidemic curves associated with DNS responses rate limiting stabilize around smaller infected user machine values than that of free propagation. This means that in the early – fast – phase of the email worm propagation the worm can reach fewer nodes.

In the interest of space and given that all the rate limiting schemes exhibit similar performance, in what follows we present results that relate only to user machine rate limiting.

4.3 Effect of Detection & Mitigation Time Parameters

In this section we concentrate on user machine rate limit scheme, and study the effect of various values for the detection and mitigation time parameters. We show how modifying only one of the detection and mitigation time and keeping the other constant affects email worm propagation to assess the relative impact of these two parameters.

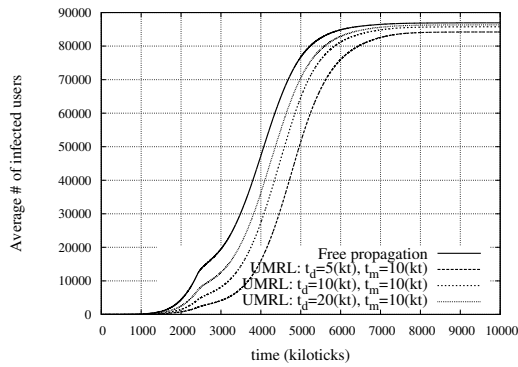


Figure 3. Epidemic curves for user machine rate limiting (UMRL) with linearly increasing t_d and constant t_m . As t_d decreases the impact on slowing down email worm propagation increases.

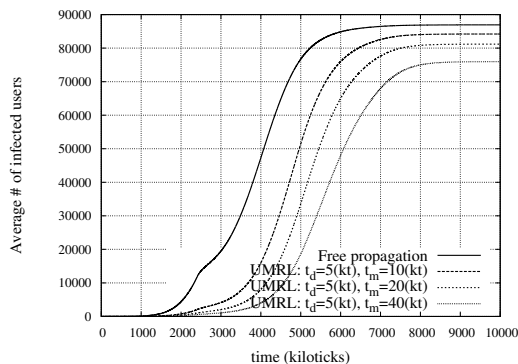


Figure 4. Epidemic curves for user machine rate limiting (UMRL) with linearly increasing t_m and constant t_d . As t_m increases the impact on slowing down email worm propagation increases and the number of user machines the email worm can reach decreases.

In Fig. 3 we depict the epidemic curves for linearly increasing detection time, while the mitigation time remains constant. As the detection time decreases, the epidemic curve shifts to the right. This implies that faster detection times have a larger impact on slowing down the propagation. This becomes clearer in Fig. 5, where the curve of the average number of active worms shifts to later times as the detection time narrows.

Fig. 4 shows the epidemic curves for linearly increasing mitigation time, while retaining the detection time constant. As mitigation time increases, the epidemic curve shifts to the right, which, as in the case of decreasing detection time, means that the email worm propagation slows down. In addition, in Fig. 4 we show that as the mitigation time increases the epidemic curves stabilize around significantly lower values of average infected user machines. This implies that apply-

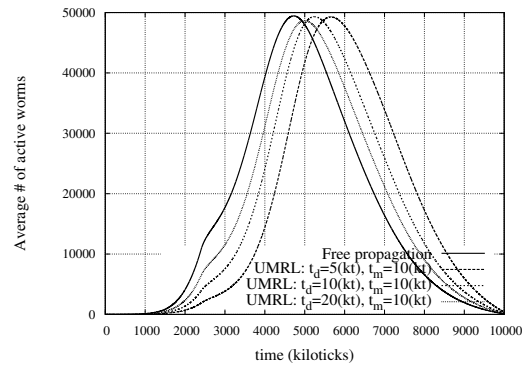


Figure 5. Average number of active worms per time for user machine rate limiting (UMRL) with linearly increasing t_d and constant t_m . As t_d decreases email worm activity is delayed.

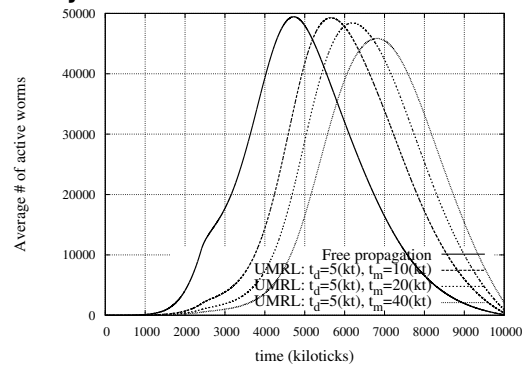


Figure 6. Average number of active worms per time for user machine rate limiting (UMRL) with linearly increasing t_m and constant t_d . As t_m increases email worm activity is delayed and the maximum number of active worms is less than this of free propagation.

ing large mitigation times effectively reduces the number of the user machines the worm can reach during its early – fast – phase. This observation is supported by Fig. 6, which shows that increasing mitigation time results in lower maximum values of average number of active worms.

5 Conclusions

Email worms remain a serious security concern, and their high prevalence is paired with serious monetary loss. Current in network defence mechanisms against this class of malicious software yielded so far only meagre results in defending the Internet and reducing abusive email traffic. Much research effort has been devoted in the past to show that a promising solution is to detect email worms at the name server, which is

located topologically near the infected user machine.

In this paper, we build on this observation and present a novel method, which is based on controlling the local name server response rate to contain email worm propagation. We conduct extensive simulation experiments that involve user behaviour, email network, physical network and email worm propagation modelling to show the effect of modifying DNS response traffic characteristics into email worms epidemics. Our results clearly indicate that rate limiting DNS responses is an effective strategy to slow down email worm propagation and can significantly affect the number of susceptible user machines an email worm can reach at its fast spreading phase.

References

- [1] Arbor networks worldwide isp security report, 2005. www.arbor.net/downloads/Arbor_Worldwide_ISP_Security_Report.pdf.
- [2] Messaging anti-abuse working group: Email metrics report, 2007. <http://www.maaawg.org>.
- [3] Symantec: Internet security threat report trends for januaryjune 07, 2007. <http://www.symantec.com/business/theme.jsp?themeid=threatreport>.
- [4] M. Barthelemy, A. Barrat, R. Pastor-Satorras, and A. Vespignani. Dynamical patterns of epidemic outbreaks in complex heterogeneous networks. *Journal of Theoretical Biology*, 235:275, 2005.
- [5] N. Chatzis. Mass mailingworm detection by means of situation aware dns. In *ISADS '07: Proceedings of the Eighth International Symposium on Autonomous Decentralized Systems*, pages 279–286, Washington, DC, USA, 2007. IEEE Computer Society.
- [6] H. Ebel, L.-I. Mielsch, and S. Bornholdt. Scale-free topology of e-mail networks. *Physical Review E*, 66:035103, 2002.
- [7] R. Fujimoto, K. Perumalla, A. Park, H. Wu, M. Amar, and G. Riley. Large-scale network simulation: how big? how fast? *Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003. MASCOTS 2003. 11th IEEE/ACM International Symposium on*, pages 116–123, 2003.
- [8] A. Gupta and R. Sekar. An approach for detecting self-propagating email using anomaly detection. In *Proc. of the International Symposium on Recent Advances in Intrusion Detection RAID, LNCS*, pages 55–72. Springer, 2003.
- [9] K. Ishibashi, T. Toyono, K. Toyama, M. Ishino, H. Ohshima, and I. Mizukoshi. Detecting mass-mailing worm infected hosts by mining dns traffic data. In *MineNet '05: Proceeding of the 2005 ACM SIGCOMM workshop on Mining network data*, pages 159–164, New York, NY, USA, 2005. ACM Press.
- [10] R. Matsuba, Y. Musashi, and K. Sugitani. Detection of mass mailing worm-infected ip address by analysis of syslog for dns server. *IPSIJ SIG*, pages 67–72, 2004.
- [11] Y. Moreno, R. Pastor-Satorras, and A. Vespignani. Epidemic outbreaks in complex heterogeneous networks. *The European Physical Journal B*, 26:521, 2002.
- [12] Y. Musashi, R. Matsuba, and K. Sugitani. Indirect detection of mass mailing worms-infected pc terminals for learners. In *3rd International Conference on Emerging Telecommunications Technologies and Applications*, pages 233–237, 2004.
- [13] Y. Musashi and K. Rannenber. Detection of mass mailing worm-infected pc terminals by observing dns query access. *IPSIJ SIG Notes*, pages 39–44, 2004.
- [14] M. Newman, S. Forrest, and J. Balthrop. Email networks and the spread of computer viruses. *Physical Review E*, 66(3):035101, 2002.
- [15] R. Pastor-Satorras and A. Vespignani. Immunization of complex networks. *Physical Review E*, 65(3):036104, 2002.
- [16] V. Paxson. Bro: a system for detecting network intruders in real-time. *Comput. Networks*, 31(23-24):2435–2463, 1999.
- [17] G. Riley. Large-scale network simulations with gtnets. *Simulation Conference, 2003. Proceedings of the 2003 Winter*, 1:676–684, 2003.
- [18] M. Roesch. Snort - lightweight intrusion detection for networks. In *LISA '99: Proc. of the 13th USENIX conference on System administration*, pages 229–238, Berkeley, CA, USA, 1999. USENIX.
- [19] S. Singh, C. Estan, G. Varghese, and S. Savage. The earlybird system for real-time detection of unknown worms. Technical Report CS2003-0761, University of California, San Diego, 2003.
- [20] D. Whyte, P. van Oorschot, and E. Kranakis. Addressing malicious smtp-based mass-mailing activity within an enterprise network. Technical Report TR-05-06, Carleton University, School of Computer Science, 2005.
- [21] C. Wong, S. Bielski, J. McCune, and C. Wang. A study of mass-mailing worms. In *WORM '04: Proc. of the 2004 ACM workshop on Rapid malware*, pages 1–10, New York, NY, USA, 2004. ACM Press.
- [22] C. Wong, S. Bielski, A. Studer, and C. Wang. Empirical analysis of rate limiting mechanisms. In *RAID*, volume 3858 of *Lecture Notes in Computer Science*, pages 22–42. Springer, 2005.
- [23] J. Xiong. Act: attachment chain tracing scheme for email virus detection and control. In *WORM '04: Proceedings of the 2004 ACM workshop on Rapid malware*, pages 11–22, New York, NY, USA, 2004. ACM Press.
- [24] C. Zou, D. Towsley, and W. Gong. Email worm modeling and defense. *Computer Communications and Networks, 2004. ICCCN 2004. Proceedings. 13th International Conference on*, pages 409–414, 2004.
- [25] C. Zou, D. Towsley, and W. Gong. Modeling and simulation study of the propagation and defense of internet e-mail worms. *IEEE Trans. Dependable Secur. Comput.*, 4(2):105–118, 2007.