Jestr Journal of Engineering Science and Technology Review 2 (1) (2009) 43-47

JOURNAL OF
Engineering Science and Technology Review

Research Article

www.jestr.org

Genetic algorithm based Internet worm propagation strategy modeling under pressure of countermeasures

N. Goranin* and A. Cenys

Information Security Laboratory, Department of Information System, Faculty of Fundamental Sciences, Vilnius Gediminas Technical University Sauletekio al. 11, SRL-I-415, LT-10223, Vilnius, Lithuania

Received 30 March 2009; Accepted 18 May 2009

Abstract

Internet worms remain one of the major threats to the Internet infrastructure. Modeling allows forecasting the malware propagation consequences and evolution trends, planning countermeasures and many other tasks that cannot be investigated without harm to production systems in the wild. Existing malware propagation models mainly concentrate on malware epidemic consequences modeling, i.e. forecasting the number of infected computers, simulating malware behavior or economic propagation aspects and are based only on current malware propagation strategies. Significant research has been done in the world during the last years to fight the Internet worms. In this article we propose the extension to our genetic algorithm based model, which aims at Internet worm propagation strategies modeling under pressure of countermeasures. Genetic algorithm is selected as a modeling tool taking into consideration the efficiency of this method while solving optimization and modeling problems with large solution space. The main application of the proposed model is a countermeasures planning in advance and computer network design optimization

Keywords: Internet worm, genetic algorithm, model, evolution, countermeasures.

1. Introduction

The number of malware in the wild is constantly increasing [1]. Despite the significant shift in motivation for malicious activity that has taken place over the past several years: from vandalism and recognition in the hacker community, to attacks and intrusions for financial gain which has been marked by a growing sophistication in the tools and methods used to conduct attacks, thereby escalating the network security arms race [2] and leading to domination of botnets in the current malware landscape [3] internet worms remain among the most significant threats to the Internet infrastructure. According to [4] they cover approximately 14% of the current malware landscape and [1] notices that the most widely reported new malicious code family in 2008 was the Invadesys worm, and 4 other worms took their places in Top10 of new malicious code families. The recent outbreak of the Conficker worm [5] shows that the worm problem remains relevant and requires further analysis.

Worms are network viruses, primarily replicating on networks. Usually a worm will execute itself automatically on a remote machine without any extra help from a user. However, there are worms, such as mailer or mass-mailer worms, that will not always automatically execute themselves without the help of a user [6]. In this article we analyze and model Internet worm propagation strategies, since their replication mechanisms differ significantly from mailer and mass-mailer worms [7]. Propagation strategy is one of the most descriptive malware characteristics [8]. Propagation of most worms is rapid (compared with classical computer viruses) and aggressive. Worms such as CodeRed and Nimda have been persistent for longer than 8 months since their introduction date. As worms spread through nearly all networks, they find nearly all of the weakest hosts accessible and begin their lifecycle anew on these systems. This then gives worms a broad base of installation from which to act [9]. The main issues faced in worm evaluation include the scale and propagation of the infections [9]. Modeling allows Internet worm researchers to predict damage for a new worm threat [10], understand the behavior of malware, including spreading characteristics [11], understand the factors affecting the malware spread, determine the required effectiveness of countermeasures in order to control the spread and facilitate network designs that are resilient to malware attacks [12], predict the failures of the global network infrastructure [13]. Since significant research has been done in the world during the last years to fight the Internet worms the worm evolution has a tendency to changes. Our proposed model [7] extension allows modeling the Internet worms' propagation strategies evolution under the pressure of countermeasures. Genetic algorithm [14] was selected as a modeling tool since it simulates natural selection by means of repeatedly evolving population of solutions (malware propagation strategies in our case) and therefore may be used for predicting and modeling possible future propagation strategies. Genetic algorithm modeling has been proved to be effective in many areas such as business decision making, bioinformatics and other [15-18].

* E-mail address: Hngrnn@fmf.vgtu.ltH

ISSN: 1791-2377 © 2009 Kavala Institute of Technology. All rights reserved.

2. Current worm propagation strategies

We define the Internet worms propagation strategy as a combination of methods and techniques, used by the worm to achieve tasks assigned to it by the worm creator. So the strategy suitable to achieve one specific task (e.g., creating the botnet) may be not useful for another (e.g., disrupting Internet functioning). Modern worms are usually created on a modular basis and may contain all or some of the following parts [9]: a reconnaissance module, that scans the Internet for vulnerable hosts; an attack module, that may exploit from one to many known vulnerabilities at potentially vulnerable host; a communication module that allows worms to communicate between themselves or to transfer information to the worm management center; a command module, that allows to accept commands; and an intelligence module, that insures functioning of the communication module, since it contains information how to find a neighbor worm for communication. Specific methods used in each of the modules are called patterns and a strategy can be also defined as a combination of patterns. A strategy is also dependent on worm introduction techniques, i.e. method used to release worm to the wild, connection protocol used (e.g. Transmission Control Protocol (TCP) or User Datagram Protocol (UDP)), etc. Since the number of existing and historic worms is high, we will describe only two propagation strategies used by CodeRed and Ramen, since they represent two different attitudes in complexity, vulnerable platform and functionality and can provide an understanding of strategies used in the wild.

On June 18th 2001 a serious Windows IIS vulnerability was discovered. On July 13th 2001 Code Red worm version 1 that exploited this single vulnerability was released. Due to a code error in its random number generator, it did not propagate well. 10:00 UTC of July 19th Code Red version 2 was released with the corrected random generator. It generated 100 threads. Each of the first 99 threads randomly chose one IP address and tried to set up connection on port 80 with the target machine (if the system was an English Windows 2000 system, the 100th worm thread would deface the infected system's web site, otherwise the thread was used to infect other systems, too) [10]. The worm was programmed to scan hosts in /8 with a 50% probability, /16 – with 37.5% probability and with 12.5% probability it would scan a totally random network [9]. Sub-networks 127.0.0.0/8, loopback, 224.0.0.0/8, multicast were excluded [13]. If the connection was successful, the worm would send a copy of itself to the victim web server to compromise it and continue to find another web server. If the victim was not a web server or the connection could not be setup, the worm thread would randomly gene-rate another IP address to probe. The timeout of the Code Red connection request was programmed to be 21 seconds. Netcraft web server survey showed that there were about 6 million Windows IIS web servers at the end of June 2001 [10]. More than 350.000 of them were infected in several hours [19].

The Ramen worm appeared in January 2001. Ramen attacked RedHat Linux 6.0, 6.1, 6.2, and 7.0 installations, taking advantage of the default installation and three known vulnerabilities: FTPd string format exploits against wu-ftpd 2.6.0, RPC.statd Linux unformatted strings exploits, and LPR string format attacks. This vulnerable software could be installed on any Linux system, meaning the Ramen worm can affect other Linux systems, as well. The worm acted in the following way: defaced any Web sites it found; disabled anonymous FTP access to the system; disabled and removed the vulnerable rpc.statd and lpd daemons, and ensured the worm would be unable to attack the host again; installed a Web server on TCP port 27374, used to pass the worm payload to the child infections; removed any host access restrictions and ensured that the worm software would start at boot time; notified the owner (worm creator) of two e-mail accounts of the presence of the worm infection. Worm then began scanning for new victim hosts by generating random class B (/16) address blocks (scans were restricted from 128/8 to 224/8, the most heavily used section of the Internet). Web server acted as a small command interface with a very limited set of possible actions. The mailboxes served as the intelligence database, containing information about the nodes on the network. This allowed the owners of the database to be able to contact infected systems and operate them as needed [9].

3. Prior and related work

3.1 Epidemiological models

The first epidemiological model of computer virus propagation was proposed by [20]. Epidemiological models abstract from the individuals, and consider them units of a population. Each unit can only belong to a limited number of states. A SIR model assumes the Susceptible-Infected-Recovered state chain and SIS model – the Susceptible-Infected-Susceptible chain. Sheila et al. in [21] use the epidemiological model as a basis for botnet modeling. The model is modified from the general model based upon the type of infection, transfer modality, and potential for re-infection and can be represented as a **M-S-E-I-R** chain, where **M** is the class of computers (hardware or software) who are not infected with malware that can be exploited to enable bot infestation; **S** is used to represent the class of computers that are infected during manufacture with malware that can be exploited to enable bot infestation. **E** is the set of computers that have been infected, are not transmitting the infection, and in whom the infection has not been detected; **I** is the set of computers that have been infected, are transmitting the infection, and in whom the infection has not been detected; **R** is the set of computers that have been infected, whose infection has been detected, and that have had their bot removed.

In a technical report [22] Zou et al. described a model of e-mail worm propagation. The authors model the Internet e-mail service as an undirected graph of relationship between people. In order to build a simulation of this graph, they assume that each node degree is distributed on a power-law probability function.

3.2 Economic models

Lelarge in [23] introduces an economic approach to malware epidemic modeling (including botnets). He states that users and computers on the network face epidemic risks. Epidemic risks (propagating viruses and worms in this case) are risks that depend on the behavior of other entities (externalities) in the network. The model based on graph theory quantifies the impact of such externalities on the investment in security features in a network. Each agent (user) can decide whether or not to invest some amount to self-protect and deploy security solutions that decrease the probability of contagion. When an agent self-protects, it benefits not only to those

who are protected but also to the whole network. If all agents invest in self-protection, then the general security level of the network is very high since the probability of loss is zero. But a self-interested agent would not continue to pay for self-protection since it incurs a cost c for preventing only direct losses that have very low probabilities. When the general security level of the network is high, there is no incentive for investing in self-protection. This results in an under-protected network.

Li et al. [24] model botnet-related cyber-crimes as a result of profit-maximizing decision-making from the perspectives of both botnet masters and renters/attackers. From this economic model, they derive the effective rental size and the optimal botnet size. Fultz in [25] describes distributed attacks organized with the help of botnets as economic security games.

3.3 Internet worm-oriented models

The Random Constant Spread (RCS) model [19] was developed by Staniford et al. using empirical data derived from the outbreak of the CodeRed worm. It assumes that the worm has a good random number generator that is properly seeded. The model assumes that a machine cannot be compromised multiple times and operates several variables: *K* is the constant average compromise rate, which is dependant on worm processor speed, network bandwidth and location of the infected host; *a(t)* is the proportion of vulnerable machines which have been compromised at the instant *t*, *Na(t)* is the number of infected hosts, each of which scans other vulnerable machines at a rate *K* per unit of time. But since a portion *a(t)* of the vulnerable machines is already infected, only *K(1-a(t))* new infections will be generated by each infected host, per unit of time. The number *n* of machines that will be compromised in the interval of time *dt* (in which *a* is assumed to be constant) is thus given by:

$$n = (Na) \cdot K(1-a)dt. \quad (1)$$

N is assumed to be a large constant address space so the chance that worm would hit the already infected host is negligible. From this hypothesis:

$$n = d(Na) = Nda \quad (2)$$

It is also possible to write

$$Nda = (Na) \cdot K(1-a)dt. \quad (3)$$

From this

$$\frac{da}{dt} = Ka(1-a) \quad (4)$$

where

$$a = \frac{e^{K(t-T)}}{1+e^{K(t-T)}}. \quad (5)$$

So the model can predict the number of infected hosts at time *t* if *K* is known. The higher is *K*, the quicker the satiation phase will be achieved by worm. As [9] states, that although more complicated models can be derived, most network worms will follow this trend.

Other authors [26] propose the discrete time model (AAWP), in the hope to better capture the discrete time behavior of a worm. However, according to [13] continuous model is appropriate for large-scale models, and the epidemiological literature is clear in this direction. The assumptions on which the AAWP model is based are not completely correct, but it is enough to note that the benefits of using a discrete time model seem to be very limited.

On the other hand Zanero et al in [13] propose a sophisticated compartment based model, which treats Internet as the interconnection of autonomous systems, i.e. sub-networks. Interconnections are a so-called "bottlenecks". The model assumes, that inside a single autonomous system (or inside a densely connected region of an AS) the worm propagates unhindered, following the RCS model. The authors motivate the necessity of their model via the fact that the network limited worm Saphire which was using UDP protocol for propagation was following the RCS model till the "bottlenecks" were flooded by its scans.

Zou et al in [27] propose a two-factor propagation model, which is more precise in modeling the satiation phase taking into attention the human countermeasures and the decreased scan and infection rate due to the large amount of scan-traffic. The same authors have also published an article on modeling worm propagation under dynamic quarantine defense [28] and evaluated the effectiveness of several existing and perspective worm propagation strategies [29].

3.4 Other malware-oriented models

Malware propagation in Gnutella type Peer-to-Peer (P2P) networks was described in [12] by Ramachandran et al. The study revealed that the existing bound on the spectral radius governing the possibility of an epidemic outbreak needs to be revised in the context of a P2P network. An analytical model that emulates the mechanics of a decentralized Gnutella type of peer network was formulated and the study of malware spread on such networks was performed.

Ruitenbeek in [30] simulates virus propagation using parameterized stochastic models of a network of mobile phones, created with the help of Mobius tool and provides insight into the relative effectiveness of each response mechanism. Two models of the propagation of mobile phone viruses were designed to study the impact of viruses on the dependability and security of mobile phones: the first model quantifies the propagation of multimedia messaging system (MMS) viruses and the second - of Bluetooth viruses.

In their presentation Zou et al. [31] suggest using botnet propagation model via vulnerability exploitation and notice some similarities of bot and worm propagation.We can not aggree with this statement since botnets use more propagation vectors than worms do. Botnet propagation modeling using time zones was proposed by Dagon et al. [32]. The model uses diurnal shaping functions to capture regional variations in online vulnerable populations.

Authors of [33] have developed a stochastic model of P2P botnet formation to provide insight on possible defense tactics and examine how different factors impact the growth of the botnet.

3.5 GA based models

In [7] we proposed the genetic algorithm based model, which was dedicated to evaluating existing as well as modeling other potentially dangerous Internet worms' propagation strategies at initial propagation phase. The efficiency of strategies was evaluated by applying the proposed fitness function. The proposed model was tested on existing worms' propagation strategies with known infection probabilities. The tests have proved the effectiveness of the model in evaluating propagation rates and have shown the tendencies of worm evolution. We have also proposed the genetic algorithm (GA) based propagation rate estimation model [8] which evaluated the negative (decrease) change of population size after satiation phase of a newly appearing worms by generating a decision tree based on a statistical data of known worms.

4. Extensions to the GA based Internet worm propagation strategy modeling framework

4.1 General model description

Since the full model description would require too much space and is available in [7] here we provide only the general model description. The model consists of a propagation strategy representation structure (each strategy is represented as a chromosome), GA acting under specified conditions and a fitness function, which evaluates the strategy's infection rate at the initial propagation phase, leaning on probability and time consumption estimations of strategy's used methods. We have chosen to model strategies for a theoretical Internet worm, which aims infecting the largest amount of hosts during a fixed relatively short period of time. Model is based on the adopted GA: during the initialization stage initial population of strategies is generated. At selection stage strategies are selected through a fitness-based process and in case termination condition is not met evolutionary mechanisms are started. In case termination condition is reached, algorithm execution is ended.

Initial population is generated on a random basis, i.e. each individual, representing separate worm propagation strategy is combined of random genes' values. Population size N is equal to 50. Population size remains constant after each new generation. The combined termination condition was selected. The algorithm would stop producing new generations in two cases: either the number of generations has reached 100, or the fitness evaluation of the fittest individual in a population remains constant for 10 consecutive generations. The crossover point for each pair of parents is selected randomly and defines the gene, after which the crossover operation is performed. The mutation operator defines the gene of a newly generated individual that should change value from current to any other random value from the range of possible gene values. Mutation operator is activated to each newly generated individual with a 0.005 probability. Fitness proportionate selection was used. The sample generated strategy may look like:

Si=(IP_GEN="Random, excluding 127.0.0.0/8, loopback, 224.0.0.0/8, multicast"; OS_PLATF="Apple OS"; TRANSF="Connection oriented"; EXPL_1=" CVE-2007-3876"; EN_EXPL_2="False"; EN_EXPL_3="False"; EN_EXPL_4="False"; EN_EXPL_5="True"; EN_EXPL_6="False"; EN_EXPL_7="False"; EN_EXPL_8="False"; EXPL_2="-"; EXPL_3="-"; EXPL_4="-"; EXPL_5=" CVE-2004-0485"**; EXPL_6="-"; EXPL_7="-"; EXPL_8="-"; EN_MEM="False"; MEM="-"; EN_HIER="True"; HIER="Autonomous"; EN_COM="False"; COM="-"; EN_EXEC="True"; EXEC="Update functionality"; EN_ADD="True"; ADD="Write to MBR to remain after reboot"; EN_EVOL="False"; EVOL="-").*

The proposed model provides a general framework for evaluating different worms' propagation strategy parameters. The proposed model was tested on existing worms' propagation strategies with known infection probabilities and was used for forecasting Internet worm propagation strategy evolution in case no countermeasures are taken.

4.2 Model extensions

In order to evaluate countermeasures efficiency on worm propagation it is necessary to classify them. In this article we use the countermeasures taxonomy proposed by Brumley et al. in [34]. The fitness function used in our previous model [7] which does not evaluated the efficiency of countermeasures was written as

$$F_S = k \cdot p_1 \cdot p_2 \cdot p_3 \sum_{i=4}^{30} p_i \quad (6)$$

where: S – evaluated strategy; p_1 – probability that the generated IP address exists and alive, p_2 – probability that host is running the OS platform that the worm supports, p_3 – probability that worm will be successfully transferred to the potential victim, p_i – probability that the i^{th} gene will result in an infected host, i=4..30; k – the number of cycles the worm, using the evaluated strategy, can perform in one second time interval.

The taxonomy [34] contains several countermeasure types: Reactive Antibody Defense (signatures, patching after worm break-out); Reactive Address Blacklisting (blocking the the connections from known infected hosts); Proactive Protection (universal system hardening based on worm disorientation); Local Containment ("good neighbor" blocking the outgoing worm scans if infected). We do not evaluate the technical problems related with the deployment of each countermeasure type, but it is obvious that their deployment is time dependant, since it takes time to prepare signatures, disseminate and constantly update blacklist, etc. It is also unarguable that worm spread becomes time dependant and the rate will decrease not when satiation phase is reached but much earlier. In that case Eq.6 can be rewritten as

$$F_S(t) = k \cdot p_1(t) \cdot p_2 \cdot p_3(t) \sum_{i=4}^{30} p_i(t) \quad (7)$$

Variable p_2 is not time-dependant since we assume that the number of computers running the OS is constant (negligible percent of users will change OS for example from Windows to Linux or vice versa in case a new worm appears) and the disorientation measures will effect in exploit efficiency. Each $p(t)$ can be described as a curve

which shows the decrease of probability. In real life all countermeasures would be used in combination. Due to that fact the function *p(t)* representing the probability decrease in time would be an approximation of the real statistical data. Currently no such data is available and systematic data collection is needed in order to create such curves.

Eq.7 could be used to draw the curve of a specific worm propagation strategy that would be decreasing in time. In order to compare efficiency of different strategies under pressure of countermeasures we propose the new fitness function:

$$F_{SC} = \frac{dF_s(t)}{dt} \tag{8}$$

which is equal to time derivative of F_S. Derivative shows the strategy's efficiency decrease rate. The lower is decrease the more efficient the strategy is.

All other model assumptions and limitations do not change. We could not check the efficiency of the proposed model extension due to the lack of statistical data but the framework proposed allows modeling of Internet worm evolution under pressure of countermeasures. It is also important to note that different countermeasure proportions may lead to different probability curves and worm strategy evolution. The future work should be concentrated on the collection of statistical data and its modeling.

5. Conclusions

In this article we have proposed the extension to our genetic algorithm based model, which aims at Internet worm propagation strategies modeling under pressure of countermeasures. Extension is based on assumption that probability of infection is time-dependant and is decreasing over time when countermeasures are being deployed. The proposed fitness function selects the evolving strategies by evaluating the decrease rate of their efficiency. Due to the lack of statistical data we can not forecast what combination of countermeasures would be the most effective in each case and future work should be concentrated on the collection of statistical data and its modeling.

The proposed model can be used as a framework in computer network design optimization. Genetic algorithm is selected as a modeling tool taking into consideration the efficiency of this method while solving optimization and modeling problems with large solution space.

References

1. D.Turner, Symantec Global Internet Security Threat Report, Symantec Corporation (2008).
2. P.Barford, and V.Yegneswaran, Proc. Special Workshop on Malware Detection, Arlington VA, USA, (2005).
3. Z.Li, A.Goyal, Y.Chen, and V.Paxson, Proc. ASIACCS'09, Sydney NSW, Australia, pp.53-56 (2009).
4. IBM, X-Force 2008 Trend & Risk Report (2009).
5. http://vil.nai.com/vil/content/v_153464.htm
6. P.Szor, The Art of Computer Virus Research and Defense, Addison Wesley Professional (2005).
7. N.Goranin, and A.Cenys, Information Technology And Control (Kaunas) 37, 133 (2008).
8. N.Goranin, and A.Cenys, Electronics and Electrical Engineering (Kaunas) 86, 23 (2008).
9. J.Nazario, Defense and Detection Strategies against Internet Worms, Artech House, Inc. (2004).
10. C.C.Zou, W.Gong, and D.Towsley, Proc. 9th ACM conference on Computer and communications security, Washington DC, USA, pp.138-147 (2002).
11. M. Garetto, W. Gong, and D. Towsley, Proc. INFOCOM, San Francisco CA, USA, pp. 1869-1879 (2003).
12. K. Ramachandran, and B. Sikdar, Proc. IPDPS, Rhodos Island, Greece, pp. 8 (2006).
13. G.Serazzi, and S.Zanero, Proc. 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems, pp. 26-50 (2006).
14. J. Holland, Adoption in natural and artificial systems, The MIT press (1975).
15. Chittur, Master Thesis, Ossining High School (2001).
16. C.Birchenhall, N.Kastrinos, and S.Metcalfe, Journal of Evolutionary Economics 7, 375 (1997).
17. J.Stender, E.Hillebrand, and J. Kingdon, Genetic Algorithms in Optimization, Simulation and Modeling, IOS Press (1994).
18. R.R.Hill, G.A.McIntyre, and S. Narayanan, Proc. SimTechT 2001, Canberra, Australia (2001).
19. S.Staniford, V.Paxson, and N.Weaver, Proc. 11th USENIX Security Symposium, San Francisco CA,USA, pp. 149-167 (2002).
20. O.J.Kephart, and S.R.White, Proc. 1991 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland CA, USA, pp. 343-359 (1991).
21. S.B.Banks, and M.R.Stytz, Challenges Of Modeling BotNets For Military And Security Simulations, Proc. SimTecT 2008, Melbourne, Australia (2008).
22. C.C.Zou, D.Towsley, and W.Gong, Technical rep. TRCSE-03-04 (2004).
23. M.Lelarge, Proc. Fifth bi-annual Conference on The Economics of the Software and Internet Industries, Toulouse, France (2009).
24. Z.Li, Q.Liao, and A.Striegel, Proc. of The Workshop on Economics of Information Security 2008, Hanover, New Hampshire, USA (2008).
25. N.Fultz, Master thesis, US Berkley School of Information (2008).
26. Zesheng Chen, Lixin Gao, Kevin Kwiat. Modeling the Spread of Active Worms // Proceedings of IEEE INFOCOM 2003. – IEEE, 2003.
27. C.C.Zou, W.Gong, and D.Towsley, Proc. CCS'02, Washington DC, USA (2002).
28. C.C.Zou, W.Gong, and D.Towsley, Proc. WORM'03, Washington DC, USA (2003).
29. C.C.Zou, W.Gong, and D.Towsley, Performance Evaluation, 63, 700 (2005).
30. E.Van Ruitenbeek, Master thesis, University of Illinois (2007).
31. C.Zou, D.Dagon, and W. Lee, Proc. ARO-DARPA-DHS Special Workshop on Botnets, Arlington VA,USA (2006).
32. D.Dagon, C.Zou, and W.Lee, 13th Network and Distributed System Security Symposium, San Diego CA, USA (2006).
33. E.Van Ruitenbeek, E.Sanders, W.H., Modeling Peer-to-Peer Botnets, Quantitative Evaluation of Systems, 2008. QEST '08. Fifth International Conference, ISBN: 978-0-7695-3360-5, Publication Date: 14-17 Sept. 2008,
34. D.Brumley, L.Liu, P.Poosankam, and D. Song, Technical Report "Taxonomy and effectiveness of worm defense strategies", Carnegie Mellon University (2005).