MOKSLAS – LIETUVOS ATEITIS
SCIENCE – FUTURE OF LITHUANIA

ISSN 2029-2341 print / ISSN 2029-2252 online
2012 4(1): 13–19 doi:10.3846/mla.2012.04

Sistemų inžinerija, kompiuterių technologija T 120
System Engineering, Computer Technology T 120

Elektronika ir elektrotechnika
Electronics and Electrical Engineering

MALICIOUS BOTNET SURVIVABILITY MECHANISM EVOLUTION FORECASTING BY MEANS OF A GENETIC ALGORITHM

Nikolaj Goranin[1], Antanas Čenys[2], Jonas Juknius[3]

[1, 2]Vilnius Gediminas Technical University
[3]The Communications Regulatory Authority of the Republic of Lithuania, Vilnius Gediminas Technical University
E-mails: [1]nikolaj.goranin@vgtu.lt; [2]antanas.cenys@vgtu.lt; [3]jjj@cert.lt

Abstract. Botnets are considered to be among the most dangerous modern malware types and the biggest current threats to global IT infrastructure. Botnets are rapidly evolving, and therefore forecasting their survivability strategies is important for the development of countermeasure techniques. The article propose the botnet-oriented genetic algorithm based model framework, which aimed at forecasting botnet survivability mechanisms. The model may be used as a framework for forecasting the evolution of other characteristics. The efficiency of different survivability mechanisms is evaluated by applying the proposed fitness function. The model application area also covers scientific botnet research and modelling tasks.

Keywords: botnet, genetic, algorithm, forecasting, survivability, evolution, model.

Introduction

The term “bot” describes a remote control program loaded on a computer usually after a successful invasion (Brand *et al*. 2009) that is used for malicious purposes (Provos, Holz 2007) often against the intentions of computer owners and without their knowledge (Lee *et al*. 2007). A botnet is a network of computers on which a bot has been installed and is usually managed remotely from a Command & Control (C&C) server. The main purpose of botnets is to use infected computers for fraudulent activities (Barroso 2007): information stealing (Barford, Yegneswaran 2005), spam distribution, performing DDoS attacks (Banks, Stytz 2008; Lee *et al*. 2007) which due to botnet development have developed from theoretical to real informational weapons (Fultz 2008), click fraud, key cracking, phishing, new malware distribution to the wild (Lee *et al*. 2007), pirated media distribution and other tasks (Karasaridis *et al*. 2007). Botnets are managed by a criminal or a group of criminals (Barroso 2007) called botmasters or botherders (Banks, Stytz 2008).

It is widely accepted that botnets pose one of the most significant and steadily increasing threats to the Internet with devastating consequences (Barroso 2007; Rajab *et al*. 2007). Bot technology has accelerated in its development in the last few years (Banks, Stytz 2008). The reason for this change is a significant shift in motivation for malicious activity: from vandalism and recognition in the hacker community to attacks and intrusions for financial gain. This shift has been marked by the growing sophistication of the tools and methods used for conducting attacks (Barford, Yegneswaran 2005; Juknius, Cenys 2009). Bot armies are effective for two reasons: they can execute multiple overt actions against targets and, alternatively, provide multiple coordinated and covert listening points within targeted networks and computer systems (Banks, Stytz 2008).

It is difficult to measure the extent of damage caused on the Internet by botnets, but the damage done is significant (Grizzard *et al*. 2007). (Rajab *et al*. 2007) find that a major contributor of unwanted Internet traffic – 27% of all malicious connection attempts – can be directly attributed to botnet-related spreading activity. The estimation of the botnet size vary considerably, especially due to the fact that hackers frequently attack large numbers of easy-to-compromise home computers (Wash 2008), the number of which cannot be measured accurately (Zhuge *et al*. 2007). For example, between 1 July and 31 December 2007, Symantec observed an average of 61,940 active bot-infected computers per day, a 17% increase from the previous reporting period (Symantec… 2008); (Zhuge *et al*. 2007) state that they tracked 3 290 IRC-based botnets during the measurement period between June 2006 and June 2007 and in total observed about 700 700 distinct IP addresses. The biggest observed botnet they tracked controlled more than 50 000 hosts. According to ENISA report (Barroso 2007), each botnet has an average of 20 000 compromised computers (bots): some C&C servers manage only a few infected computers while the large ones manage thousands of bots. Our measurements performed at the Communications

Regulatory Authority of the Republic of Lithuania have shown the following dynamics: in 2007, the total number of unique bots in Lithuania was equal to 3 917, in 2008 – 1 502 (due to effective measures taken against centralized botnets) and 10 690 for only three first months of 2009. The trend toward economic motivation is likely to catalyze the development of new capabilities in the botnet code making the task of securing networks against this threat much more difficult (Barford, Yegneswaran 2005). According to (Barroso 2007), the objective of criminals using botnets will be twofold: to increase the number of infected computers and a growth in stability and survivability applying different methods. An important point is to outline and predict botnet mechanisms for the future since this allows researchers to develop countermeasures against these kinds of botnets before they appear in the wild (Starnberger *et al*. 2008). The article proposes the genetic algorithm (GA) based malware strategy forecasting model for predicting techniques the future botnets will use for improving their survivability mechanisms.

Botnet Analysis

Botnets possess the characteristics of a virus, a worm and a Trojan (Banks, Stytz 2008). They are different from traditional discrete infections as they act as a coordinated attacking group (Dagon *et al*. 2005). The overall architecture and implementation of botnets is complex and is evolving toward the use of common software engineering techniques such as modularity (Barford, Yegneswaran 2005).

Botnets usually do not rely on a single method of propagation but make use of a combined approach. The methods include scanning for vulnerable hosts (Li *et al*. 2007), network shares, spam or unsolicited e-mail, P2P (Peer-to-peer), (Barroso 2007), net news, web blogs, other WEB resources (Dagon *et al*. 2005), social engineering via an enticement 'lure' e-mail, browser exploit and malicious file download, (Barroso 2007), via instant messenger (Dittrich, D., Dittrich, S. 2002), etc. Separate botnet parts can use different propagation methods. In case botnet uses scanning for the search of vulnerable hosts (Li *et al*. 2007), three types of scanning can be separated: localized scanning (each bot chooses scanning range based on its own IP prefixes), targeted scanning (bot master specifies a particular IP prefix for bots to scan) and uniform scanning (scanning the whole Internet). Botnet creators are forced to use different obfuscation and deception methods for protecting botnets. The examples of the advanced modern botnet rely on a wide range of complex methods such as extremely resilient random topologies, traffic anonymization (Dagon *et al*. 2005), load balancing, reverse proxies for C&C servers, fast-flux services, Rock Phish (Barroso 2007), the encrypted/obfuscated control channel (Wang *et al*. 2007) and many others (Dagon *et al*. 2005; Wang *et al*. 2007). The recent trend is toward smaller botnets with only several hundred to several thousand zombies since big botnets are bad from the standpoint of manageability and the wider availability of broadband access makes smaller botnets as capable as the larger botnets earlier (Vogt *et al*. 2007).

Most botnets that appeared prior to 2005 (Canavan 2005) had common centralized architecture, i.e. the bots in the botnet were connected directly to C&C servers (Wang *et al*. 2007) and were based on IRC due to its ability to scale to thousands of clients easily (Karasaridis *et al*. 2007).

According to ENISA report (Barroso 2007), IRC is still being used by some botnets, but HTTP is now more widespread, since it is even easier to implement and can be hidden in normal user web navigation traffic. One of the most typical and widespread IRC botnets is the Agobot family, which is among the most widespread in the number of variants created according to (Gordon 2004). The (Gordon 2004) paper presents a detailed overview of the history of the Agobot family and changes in the functionality and description of them. Agobot is created on a modular basis and mainly affects computers running MS Windows platform. Bot development kit is distributed under GNU GPL license (Gordon 2004). Agobot can exploit many well-known OS vulnerabilities and back doors left by other viruses (Barford, Yegneswaran 2005). Exploits and delivery functions are separated. Once the first step exploits succeed, it opens a shell on the remote host to download bot binary encoded to avoid network-based signature detection. The bot has the module of testing for debuggers and VMWare once it is installed. In case it detects VMWare, it stops running (Zhaosheng *et al*. 2002).

Considering the above mentioned weaknesses inherent to the centralized architecture of C&C botnets, it is a natural strategy for botmasters to design a peer-to-peer (P2P) control mechanism into their botnets (Wang *et al*. 2007). In P2P architecture, there is no centralized point for C&C (Grizzard *et al*. 2007). Nodes in the P2P network act as both clients and servers in a way that there is no centralized coordination point that can be incapacitated. If the nodes in the network are taken offline, the gaps in the network are closed and the network continues to operate under control of the botmaster. One more problem posed by P2P botnets to security specialists is difficulty in estimating the size of the P2P botnet (Dittrich, D., Dittrich, S. 2002). The best known decentralized botnet is the STORM botnet. SANS Institute (Sans… 2008) has named the STORM botnet as the biggest security issue of the year 2008. There are no significant differences from IRC botnets in malicious functionality; however, it differs in resistance and self-protection mechanisms. The P2P-based botnet STORM based on the Kademlia (eDonkey) algorithm so that it can contact its peer list if they are online (Mukamurenzi 2008) is very hard to trace and shut down, because the botnet has robust network connectivity, uses encryption and controls traffic dispersion. Each bot influences only a small part of the botnet, and upgrade/recovery is accomplished easily by its botmaster (Zhang 2008). It has aggressive defences (attacks anyone (DDoS) who tries to analyse it or reverse engineer it) and uses clique architecture where each clique has its own 40-bit encryption key. There are no file exchanges between infected hosts that make it difficult to track. STORM uses fast-flux service networks. The DNS records of the website changes every few minutes (Barroso 2007; Mukamurenzi 2008).

Prior and Related Work

The first epidemiological model for computer virus propagation was proposed by (Kephart, White 1991). Epidemiological models abstract from the individuals and consider them the units of a population. Each unit can only belong to a limited number of states (Susceptible-Infected-Recovered state chain in the SIR model and Susceptible-Infected-Susceptible in SIS).

The Random Constant Spread (RCS) model (Staniford *et al*. 2002) was developed using empirical data derived from the outbreak of the CodeRed worm. As (Nazario 2004) states, although more complicated models can be derived, most network worms will follow this trend. (Chen *et al*. 2003) propose the AAWP discrete time model in the hope to better capture the discrete time behaviour of a worm. However, according to (Serazzi, Zanero 2004), a continuous model is appropriate for large scale models and the benefits of using a discrete time model seem to be very limited. On the other hand, (Serazzi, Zanero 2004) propose a sophisticated compartment based model treating the Internet as the interconnection of autonomous systems, i. e. subnetworks. Interconnections are called "bottlenecks". The model assumes that inside a single autonomous system the worm propagates unhindered, following the RCS model. (Zou *et al*. 2002) propose a two-factor propagation model that is more precise in modelling the satiation phase taking into consideration human countermeasures, decreased scan and infection rate due to a large amount of scan-traffic. The same authors have also published an article on modelling worm propagation under dynamic quarantine defence (Zou *et al*. 2003) and evaluated the effectiveness of several existing and perspective worm propagation strategies (Zou *et al*. 2005).

In a technical report (Zou *et al*. 2004), the authors described a model of e-mail worm propagation. Malware propagation in Gnutella type P2P networks was described in (Ramachandran, Sikdar 2006). An analytical model that emulates the mechanics of the decentralized Gnutella type network was formulated and the study of malware spread on such networks was performed. The authors of (Ruitenbeek *et al*. 2007) simulate virus propagation using parameterized stochastic models for a network of mobile phones created with the help of Möbius tool.

In (Goranin, Cenys 2008), we have proposed the GA based model dedicated to forecasting the evolution of propagation techniques used by the Internet worms at the initial propagation phase. The effect of countermeasures on the evolution of the Internet worm was discussed in (Goranin, Cenys 2009). A rather similar concept was proposed almost one year later in (Noreen *et al*. 2009).

Several botnet-oriented models were put forward; nevertheless, they all concentrate on the tasks other than forecasting botnet evolution. (Lelarge 2002) introduces an economic approach to malware epidemic modelling (including botnets). (Li *et al*. 2009) model botnet-related cybercrimes as a result of profit-maximizing decision-making from the perspectives of both botnet masters and renters/attackers. From this economic model, they derive an effective rental size and an optimal botnet size. (Fultz 2008) describes distributed attacks organized with the help of botnets as economic security games. (Banks, Stytz 2008) use the epidemiological model as a basis for botnet modelling. In (Zou *et al*. 2008), the authors suggest using the botnet propagation model via vulnerability exploitation and notice some similarities of bot and worm propagation. Botnet propagation modelling using time zones was proposed in (Dagon *et al*. 2006). The authors of (Ruitenbeek, Sanders 2008) developed a stochastic model of P2P botnet formation to provide insight on possible defence tactics and examine how different factors impact the growth of the botnet.

Botnet Strategy Evolution Forecasting Model

As stated in (Banks, Stytz 2008), the development of botnet simulation and modelling capabilities requires advances in improving the understanding of botnet technologies and the development of standards that support the simulation of bot army operations. However, these tasks are complex for a variety of reasons such as a wide variety of botnets and their manner of propagation, challenge posed by modelling the amount of time and the patterns of their infestation. Thus, the GA approach to modelling botnets is extremely efficient due to its ability to solve complex problems with large solution space. GA (Holland 1975) simulates natural selection by means of the repeatedly evolving population of solutions (botnet survivability strategies in our case), and therefore may be used for predicting and modelling possible future survivability strategies (combination of methods and techniques used by malware). While creating the GA based model, there are two main tasks to be solved: first, it is the correct selection of chromosomes and genes representing the proposed solution; second, the creation of the fitness equation (or fitness evaluation criteria such as statistical evaluation etc.) that evaluates the fitness of the solution (single chromosome) generated during evolution and selects the most appropriate solutions according to the selection strategy. It is obvious that solution representation is highly dependent on the analyzed problem and its complexity, whereas the fitness function – on the evaluated criteria.

GA consists of initialization, selection and evolution stages. During the initialization stage, the initial population of strategies is generated. Each strategy is represented as a chromosome. The initial population of strategies is generated on a random basis. At the selection stage, the strategies are selected through the fitness-based process. In case termination condition is not met, evolutionary mechanisms are started. In case termination condition is reached (number of generations or evolution stagnation), algorithm execution is ended. Otherwise – evolutionary mechanisms are activated. 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 the value from current to any other random value from the range of possible gene values. Fitness proportionate selection is used for parent selection.

Strategy Representation

The behaviouristic characteristics of botnets (propagation vectors, communication channels and hierarchy, functionality, defence techniques) can be described as a botnet strategy. In GA modelling, each strategy is represented as a chromosome. Botnets vary from other malware types in complexity. Many of them change behaviour at different periods of their existence. Thus, the creation of a universal representational chromosome is a complex task. The representation of the proposed strategy via chromosomes is provided in Table.

We suggest using a reference system for gene activation and method definition (selection from maximum 9 methods or gene deactivation). In Table, the gene (fixed-length number – 10 positions) in the “Gene code” column activates one or several methods from the reference database (samples are provided in the “Reference database (or sample)” column). “0” marks references to the methods that do not exist, whereas other digits (1–9) point to the references to the database. The number of non-zero digits shows the number of the methods activated. No the same non-zero digits in one gene can appear (e.g. 4510000700 is “OK”, but 4550000700 is not). This check is performed during the initial population generation phase. The non-zero digit order is not important (in case the gene is 4510000700, methods 4, 5, 1 and 7 are active). The 0000000000 gene means that it is not active and no methods are used. If the gene is compulsory for the botnet but is not active, such an individual (strategy) will be simply eliminated by the evolutionary selection process. This check is performed at the initial population generation phase.

Table. Chromosome structure

Lentelė. Chromosomos struktūra

Gene No./ code/descr.	Reference database (or sample)	Comments
1 / TARGET_SEARCH/ Defines the methods used by the botnet for potential victim search	1) Scan – Random; 2) Scan – Random, excluding 127.0.0.0/8, loopback, 224.0.0.0/8, multicast, NAT 3) Scan – Random addresses from the networks reserved for home user networks 4) Non-automatic infection 5) E-mail spam 6) Instant messaging 7) Infected site 8) Removable media 9) P2P	Each method may use from 1 to 9 exploits. The necessary number of exploits with referencing exploiting probability is selected from the list of exploits. Limitations: exploits used for one method should run on a single platform
2 / TRANSF / Defines the mechanism for exploit body transfer	1) Connectionless (also called “Fire and forget”) 2) Connection oriented	The connectionless mechanism uses UDP protocol for exploiting body transfer, connection oriented ~ TCP. Important in case of scan. “Connection oriented” in all other cases
3 / EXPL_PLATF / Defines the OS platform used for each method	1) DOS 2) *nix (Unix, Solaris, Linux) 3) Win9x 4) Win NT (NT, 2000, XP) 5) Mobile OS 6) Apple OS 7) Multi platform 8) Other OS 9) WEB application exploit	All exploits used should run on a single platform. Different methods may run on different OS. WEB based exploits are not OS dependent

End of Table

Gene No./ code/descr.	Reference database (or sample)	Comments
4 / EXPL_NUM / Defines the number of exploits	[0..9] (random number) Defines the number of exploits used by each method	A random number of exploits used by each method, activated in TARGET_SEARCH
5 / HIERARCHY / Defines the hierarchy used for communication	1) Central – 1 management host (further MH) 2) Central – 1 MH – botnet is split 3) Central – several (2–9) MH – independent 4) Central – several (2–9) MH – load balancing 5) Central – several (2–9) MH hosts – fast-flux protection 6) Central – several (2–9) MH – fast-flux protection – load balancing 7) Belongs to several botnets (2–9) with central MH 8) Decentralized – P2P 9) Decentralized – P2P and fast-lux technology	In this gene, only the first number out of 10 has sense and defines the hierarchy used for management. Other are only "0"
6 / FUNCTIONALITY / Defines functionality that the botnet provides for botnet owners.	1) Information collection 2) Backdoor opening 3) Botnet owner notification about the Compromise 4) Packet sniffing 5) DDoS functionality 6) Spam sending 7) Remote update and deinstallation of the installed bot 8) Botnet rental tools 9) Botnet ease-of-use	This list is not complete since it is limited to the number of 9, which is selected for the reason of simplicity
7 / SELF_PROTECT / Defines the methods that a single bot can use for protecting itself/	1) Blocking Firewall / Antivirus processes 2) Blocking Antivirus Updates 3) Blocking OS updates 4) Deinstallation imitation, if detected by Antivirus 5) Imitation of usefulness 6) Period of inactivity 7) Low activity 8) De-installation if "honeypot" is suspected 9) Social engineering	This list is not complete since it is limited to the number of 9, which is selected for the reason of simplicity

Definition of Botnet Stability and Evolution Trends

For botmasters, it is necessary to insure that the botnet will be stable (functional, manageable and of a relatively fixed size) for time period *T* necessary to fulfil botmasters tasks. *T* is task dependant survivability mechanisms activated in the botnet to insure the necessary stability level in each case will be different.

Botnet stability may be discussed in two aspects: hierarchy stability insuring the overall functionality and stability of the botnets (e.g. if C&C is blocked, the bots become useless even if not detected by antivirus programs) and the stability of botnet nodes – bots, i.e. probability that the bot will not be removed from the botnet network by different countermeasures. To insure stability, botnet creators implement different survivability (or protection) mechanisms.

In case we want to evaluate botnet evolution only in of survivability, we can say that hierarchy stability is directly proportional to the number of C&C used and corresponding protection mechanisms and can be calculated according to Equation 1. GA modelling is not needed since the evolution trend is clear and the number of possible trends is very limited.

$$N = \frac{T}{t_{\text{CC\_block}}(hierarchy\_Nr)}, \tag{1}$$

where *N* – the number of C&C; *T* – time interval necessary for the botnet to remain stable; $t_{\text{CC\_block}}$ – the average time needed for botnet fighters to block the C&C of a specific hierarchy (*hierarchy_Nr*) using or not using some self-protection measures. In reality, the decision of the botnet creator would be based on an economic evaluation of hierarchy implementation or realization complexity.

Another botnet stability part – bot or node stability - has much more possible survivability mechanisms combinations, dependent on a variety of functions the bot has to perform and evolution trends are not so clear. That is why we propose applying the GA model and evaluation of survivability mechanism via the fitness-based process.

Fitness Function and Model Limitations

Considering the definition, the fitness function is a particular type of an objective function that quantifies the optimality of a solution so that a particular individual may be ranked against the other individuals. In our case, we propose the fitness function that allows the evaluation of survivability mechanism used by the botnet nodes.

We state that bot survivability can be evaluated by the use of probabilistic and time consumption parameters for the methods activated by genes for each of the strategies:

$$F(S) = k \cdot \prod_{i=1}^{9} \left(1 - p_i\left(self\_protect\right)\right), \tag{2}$$

where *F* – the fitness function; *S* – botnet strategy being evaluated; p_i – probability that the i^{th} method (*self protect*) will protect the botnet node bot from detection and removal; k – bot activity level (e.g. sniffing, spam sending, DDoS performing, etc.) that directly influences the efficiency of self-protection measures employed by the bot since the higher is the bot activity level, the higher is probability that it is noticed and removed. k is calculated:

$$k = \frac{t(S)}{T}, \tag{3}$$

where

$$t = \sum_{j=1}^{7} \sum_{i=1}^{9} t_i \cdot \frac{CPU\_LOAD_i}{100\%} \tag{4}$$

is the summary of time consumption of strategy S in the evaluated time period T; t_i – time consumption of a specific method (j – the gene's index; i – the method's index of the j^{th} gene); CPU_LOAD_i – average method's load on CPU of the infected computer during time t_i. CPU load is selected as one of the most descriptive computer process activity parameters. Model limitation is introduced so that all computers included in the botnet run almost similar CPUs by the processing power.

Conclusions

The botnet-oriented genetic algorithm based malware strategy evolution forecasting model was proposed, which aims at forecasting botnet survivability mechanism evolution forecasting and may be used as a framework for forecasting the evolution of other botnet characteristics. The model consists of the structure representing propagation strategy, the genetic algorithm acting under specified conditions and a fitness function for botnet survivability mechanism strategy evaluation. Due to a lack of statistical data, the model is provided as a proof-of-concept with no real data tests. The model can be extended or adapted for other malware types or characteristic evolution modelling.

References

Banks, B. S.; Stytz, R. M. 2008. Challenges of modeling botnets for military and security, in *Proceedings of SimTecT 2008*.

Barford, P.; Yegneswaran, V. 2005. An inside look at botnets, in *Proceedings of Special Workshop on Malware Detection. Advances in Information Security*, Vol. 27. Springer US, 171–191.

Barroso, D. 2007. *Botnets – the Silent Threat*. ENISA Position Paper, 3, 1–9.

Brand, M.; Champion, A.; Chan, D. 2009. *Combating the Botnet Scourge* [online]. Available from Internet: http://www.cse.ohio-state.edu/~champion/research/Combating_the_Botnet_Scourge.pdf

Canavan, J. 2005. The evolution of malicious IRC bots, in *Proceedings of Virus Bulletin Conference 2005*, 104–114.

Chen, Z.; Gao, L.; Kwiat, K. 2003. Modeling the spread of active worms, in *Proceedings of IEEE INFOCOM 2003*, 1890–1900.

Dagon, D.; Gu, G.; Zou, C.; Grizzard, J.; Dwivedi, S.; Lee, W.; Lipton, R. 2005. A taxonomy of botnets, in *Proceedings of CAIDA DNS-OARC Workshop*. 16 p.

Dagon, D.; Zou, C.; Lee, W. 2006. Modeling botnet propagation using time zones, in *Proceedings of the 13 th Network and Distributed System Security Symposium*. 15 p.

Dittrich, D.; Dittrich, S. 2002. P2P as botnet command and control: a deeper insight, in *Proceedings of the 2008 3rd International Conference on Malicious and Unwanted Software*, 41–48.

Fultz, N. 2008. *Distributed attacks as security games*. US Berkley School of Information, Berkley. 8 p.

Goranin, N.; Cenys, A. 2008. Genetic algorithm based internet worm propagation strategy modeling, *Information Technology and Control* 37(2): 133–140.

Goranin, N.; Cenys, A. 2009. Genetic algorithm based Internet worm propagation strategy modeling under pressure of countermeasures, *Journal of Engineering Science and Technology Review* 2(1): 43–47.

Gordon, J. 2004. *Agobot and the the "Kit" chen Sink* [online]. Available from Internet: http://www.infectionvectors.com/vectors/Agobot_&_the_Kit-chen_Sink.pdf

Grizzard, B. J.; Sharma, V.; Nunnery, C.; Kang, B. B. H.; Dagon, D. 2007. Peer-to-peer botnets: overview and case study, in *Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets*.

Holland, J. 1975. *Adoption in Natural And Artificial Systems*. The MIT press. 211 p.

Juknius, J.; Cenys, A. 2009. Intelligent botnet attacks in modern Information warfare, in *Proceedings of 15th International Conference on Information and Software Technologies*, 39–42.

Karasaridis, A.; Rexroad, B.; Hoeflin, D. 2007. Wide-scale botnet detection and characterization, in *Proceedings of USENIX Workshop on Hot Topics in Understanding Botnets 2007*.

Kephart, O. J.; White, R. S. 1991. Directed-graph epidemiological models of computer viruses, in *Proceedings of the 1991 IEEE Computer Society Symposium on Research in Security and Privacy*, 342–359. http://dx.doi.org/10.1109/RISP.1991.130801

Lee, W.; Wang, C.; Dagon, D. 2007. *Botnet Detection: Countering the Largest Security Threat*. Springer. 168 p.

Lelarge, M. 2002. The economics of malware: epidemic risks model, network externalities and incentives, in *Proceedings of Fifth bi-annual Conference on the Economics of the Software and Internet Industries*, *Toulouse, 2009.*

Li, Z.; Goyal, A.; Chen, Y. 2007. Honeynet-based botnet scan traffic analysis, in *Advances in Information Security*, Vol. 36. Springer US, 25–44.

Li, Z.; Liao, Q. C.; Striegel, A. 2009. Botnet economics: uncertainty matters, in *Managing Information Risk and the Economics of Security*. Springer US, 1–23.

Mukamurenzi, N. M. 2008. *Storm Worm: A P2P Botnet*. Norwegian University of Science and Technology, Department of Telematics.

Nazario, J. 2004. *Defense and Detection Strategies Against Internet Worms*. Artech House, Inc. 322 p.

Noreen, S.; Murtaza, S.; Shafio, M. Z.; Faroo, M. 2009. Evolvable malware, in *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*. ACM, 1569–1576.

Provos, N.; Holz, T. 2007. *Virtual Honeypots: from Botnet Tracking to Intrusion Detection*. Addison-Wesley Professional. 440 p.

Rajab, M. A.; Zarfoss, J.; Monrose, F.; Terzis, A. 2007. My botnet is bigger than yours (maybe, better than yours): why size estimates remain challenging, in *Proceedings of the First Conference on Hot Topics in Understanding Botnets'07*, 5–5.

Ramachandran, K.; Sikdar, B. 2006. Modeling malware propagation in Gnutella type peer-to-peer networks, in *Parallel and Distributed Processing Symposium* 20: 8.

Ruitenbeek, V. E.; Courtney, T.; Sanders, H. W.; Stevens, F. 2007. Quantifying the effectiveness of mobile phone virus response mechanisms, in *Proceedings of Dependable Systems and Networks, 2007 // 37th Annual IEEE/IFIP International Conference, 2007*, 790–800.

Ruitenbeek, V. E.; Sanders, H. W. 2008. Modeling Peer-to-Peer Botnets, in *Proceedings of Quantitative Evaluation of Systems, 2008 // Fifth International Conference*, 307–316.

SANS Institute. 2008. *The Sans Institutes Top Ten Cyber Security Menaces for 2008* [online]. Available from Internet: http://www.itu.int/

Serazzi, G.; Zanero, S. 2004. Computer Virus Propagation Models, in *Lecture Notes in Computer Science*, Vol. 2965/2004. Springer-Verlag, 26–50.

Staniford, S.; Paxson, V.; Weaver, N. 2002. How to own the internet in your spare time, in *Proceedings of the 11th USENIX Security Symposium*, 149–167.

Starnberger, G.; Kruegel, C.; Kirda, E. 2008. Overbot: a botnet protocol based on Kademlia, in *Proceedings of the 4th International Conference on Security and Privacy in Communication Networks*, 1–9.

Symantec Corp. 2008. *Symantec Global Internet Security Threat Report Trends for July–December 07* [online]. Available from Internet: http://www.symantec.com

Vogt, R.; Aycock, J.; Jacobson, M. J. Jr. 2007. Army of botnets, in *Proceedings of Network and Distributed System Security Symposium*, 111–123.

Wang, P.; Sparks, S.; Zou, C. 2007. An advanced hybrid peer-to-peer botnet, in *Proceedings of USENIX Workshop on Hot Topics in Understanding Botnets*, 39–42.

Wash, R. 2008. *Mental Models of Home Computer Security* [online]. Available from Internet: http://cups.cs.cmu.edu

Zhang, J. 2008. *Storm Worm & Botnet Analysis* [online]. Available from Internet: http://securitylabs.websense.com

Zhaosheng, Z.; Guohan, L.; Yan, F. C.; Zhi, J. R.; Han, P. 2002. Botnet research survey, in *Proceedings of the 2008 32nd Annual IEEE International Computer Software and Applications Conference*, 967–972.

Zhuge, J.; Holz, T.; Han, X.; Guo, J.; Zou, W. 2007. *Characterizing the IRC-based Botnet Phenomenon.* Peking University & University of Mannheim Technical Report.

Zou, C. C.; Dagon, D.; Lee, W. 2008. *Modeling and Measuring Botnets* [online]. Available from Internet: http://www.gtisc.gatech.edu

Zou, C. C.; Gong, W.; Towsley, D. 2002. Code red worm propagation modeling and analysis, in *Proceedings of CCS'02*, 138–147.

Zou, C. C.; Gong, W.; Towsley, D. 2003. Worm propagation modeling and analysis under dynamic quarantine defense, in *Proceedings of WORM'03*, 51–60.

Zou, C. C.; Gong, W.; Towsley, D. 2005. On the performance of Internet worm scanning strategies, *Performance Evaluation* 63(7): 700–723. http://dx.doi.org/10.1016/j.peva.2005.07.032

Zou, C. C.; Towsley, D.; Gong, W. 2004. Email virus propagation modeling and analysis, *Technical report TRCSE- 03–04*. University of Massachussets.

KENKĖJIŠKŲ *BOTNET* TINKLŲ IŠGYVENAMUMO MECHANIZMŲ EVOLIUCIJOS PROGNOZAVIMAS GENETINIO ALGORITMO PRIEMONĖMIS

N. Goranin, A. Čenys, J. Juknius

Santrauka

Botnet tinklai pripažįstami kaip vieni pavojingiausių šiuolaikinių kenksmingų programų ir vertinami kaip viena iš didžiausių grėsmių tarptautinei IT infrastruktūrai. *Botnet* tinklai greitai evoliucionuoja, todėl jų savisaugos mechanizmų evoliucijos prognozavimas yra svarbus planuojant ir kuriant kontrpriemones. Šiame straipsnyje pateikiamas genetiniu algoritmu pagrįstas modelis, skirtas *Botnet* tinklų savisaugos mechanizmų evoliucijai prognozuoti, kuris taip pat gali būti naudojamas kaip pagrindas kitų *Botnet* tinklų savybių evoliucijai modeliuoti. Skirtingi savisaugos mechanizmai vertinami taikant siūlomą tinkamumo funkciją.

Reikšminiai žodžiai: *Botnet*, genetinis algoritmas, prognozė, savisauga, evoliucija, modelis.