EXPERIMENTS WITH SIMULATION OF BOTNETS AND DEFENSE AGENT TEAMS

Igor Kotenko
Laboratory of Computer Security Problems
St. Petersburg Institute for Informatics and Automation
39, 14th Liniya, St. Petersburg, 199178, Russia
E-mail: ivkote@comsec.spb.ru

KEYWORDS
Agent-based Simulation, Network Simulation, Network attacks and defense, Botnets, DDoS.

ABSTRACT

Botnets allow malefactors manage millions of infected computers simultaneously and provide large-scale successful attacks. The paper suggests an approach for multi-agent simulation of botnets and botnet protection mechanisms. The main contribution of the paper is an improved simulation environment for agent based simulation of botnets and experimentation with this environment for analysis of different botnets and protection mechanisms. Experiments demonstrate the capabilities of the simulation environment for investigating various stages of the botnet lifecycle and the efficiency of different protection mechanisms.

1. INTRODUCTION

Botnets allow malefactors manage millions of infected computers simultaneously and provide large-scale successful attacks. One of the promising approaches for investigation of botnets and protection mechanisms is agent-based simulation.

The paper is devoted to the study of botnets propagating using network computer worms and used to execute Distributed Denial of Service (DDoS) attacks. The paper is based on earlier publications of the authors on agent-based simulation (Kotenko and Ulanov 2007; Kotenko 2009; Kotenko 2010; Kotenko et al. 2012-1; Kotenko et al. 2012-2). In the paper we try to elaborate the agent-oriented approach suggested in these works for simulation of botnets and botnet protection mechanisms in the Internet. The used agent-oriented approach supposes that the network counteraction is represented as the interaction of different teams of software agents (Kotenko and Ulanov 2007; Kotenko 2009; Taveter et al. 2010), and the aggregated system behavior appears by means of the local interactions of particular agents in a dynamic environment that is defined by the model of the Internet.

In distinction from those papers, the main contribution of the paper is the improved simulation environment for agent based simulation of botnets and experimentation with this environment for analysis of different botnets and protection mechanisms. Many new experiments were added, their representation and analysis methods were improved. This enabled us to compare the capabilities of the defense methods against botnets in the course of their life cycle. The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 presents the simulation framework and the architecture of the simulation environment developed. Section 4 describes a set of experiments fulfilled. Concluding remarks are given in Section 5.

2. RELATED WORK

Research on agent-based simulation is based on a variety of methods and approaches. The classical frameworks and architectures for multi-agent simulation of distributed complex systems are shared plans theory (Grosz and Kraus 1996), joint intentions theory (Cohen and Levesque 1991) and hybrid approach (Tambe 1997). The techniques based on belief-desire-intention, distributed constraint optimization, distributed Partially observable Markov decision process, and game-theoretic (Tambe et al. 2005) are emphasized. Different mechanisms for collaborative agent team maintenance are used (Kaminka et al. 2007; Stone et al. 2010; Agmon et al. 2011).

In the studies on the analysis of botnets, the definition of their lifecycle is given (Govil and Jivika 2007; Feily et al. 2009; Naseem et al. 2010). It consists of the phases (or stages) of primary infection, propagation, management and control, and attack. The roles of the participants of botnets are considered (Feily et al. 2009), features of the botnets with centralized (Govil and Jivika 2007; Naseem et al. 2010) and decentralized (Bailey et al. 2009; Grizzard et al. 2007; Feily et al. 2009; Wang et al. 2007) architecture are analyzed, and various types of attack executed using botnets are described. In (Dagon et al. 2007) the efficiency criteria of botnet operation are discussed.

Studies on simulation for investigation of botnets and computer networks mainly rely on methods of discrete-event simulation of processes being executed in network structures (Simmonds et al. 2000; Wehrle et al. 2010), as well as on trace-driven models initiated by trace data taken from actual networks (Owezarski and

Proceedings 27th European Conference on Modelling and Simulation ©ECMS Webjørn Rekdalsbakken, Robin T. Bye, Houxiang Zhang (Editors)
ISBN: 978-0-9564944-6-7 / ISBN: 978-0-9564944-7-4 (CD)

Larrieu 2004). G.Riley et al. (Riley et al. 2004) implement a network worm propagation model. A.Suvatne (Suvatne 2010) suggests a model of "Slammer" worm propagation by using "Wormulator" (Krishnaswamy 2009) simulation environment. M.Schuchard (Schuchard et al. 2010) presents simulation tool which allows to simulate a large scale botnet. Gamer and Mayer (Gamer and Mayer 2009) consider a DDoS simulation tool, called Distack using OMNeT++. Li et al. (Li at al. 2002) propose own simulation environment to estimate the quality of implementation of botnet protection mechanisms.

This paper describes the approach which combines agent-based and discrete-event packet-level simulation of network protocols. Initially this approach was suggested for DDoS attack and defense simulation (Kotenko and Ulanov 2007). In the present paper, as against other works of authors, the various methods of botnet attacks and counteraction against botnets are explored by implementing comprehensive libraries of attack and defense components.

3. SIMULATION FRAMEWORK AND ENVIRONMENT

The proposed simulation environment realizes simulation models which implement the processes for operation of botnet agent teams and defense agent teams. Main components of the simulation framework which is implemented in the simulation environment are as follows:

- Ontology of application domain containing application notions and relations between them;
- Protocols of teamwork and taskwork maintenance for the agents of different teams;
- Models of scenario behavior of agents for team, group and individual levels;
- Libraries of agent basic functions;
- Communication platform and components for agent message exchange;
- Models of functioning environment, including topological, functional and other components;
- Models that provide the interaction of teams (antagonistic and non-antagonistic competing, cooperation, adaptation).

We distinguish at least three types of agent teams: the botnet teams, the defense teams, and the teams of usual users and servers.

Botnet teams include the following types of agents: (bot)master, command centre (C&C), zombies (bots). Botmaster, by sending different commands, sets goals for the botnet and controls the behaviour of the network at the highest level. C&C carries out the delivery of commands received from the botmaster to bots. Bots, receiving the commands from the C&C, immediately carry out actions under the orders of the botmaster.

According to the types of communication channels between the agents, four kinds of structures of teams are used (Figure 1-3): (1) centralized; (2) simple distributed; (3) multilevel distributed; (4) peer to peer (P2P). In P2P team (Figure 3) each node can be represented as (bot)master, C&C or zombies (bots).

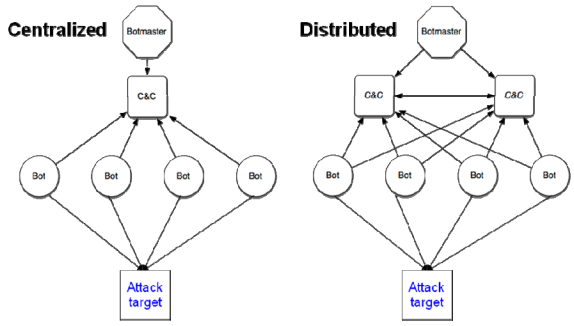

Figure 1: Centralized and simple distributed Botnet Teams

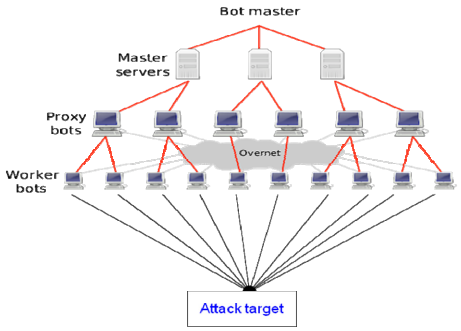

Figure 2: Multilevel distributed Botnet Team

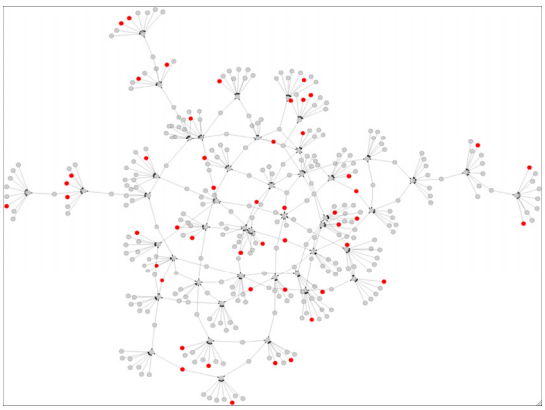

Figure 3: Peer to peer (P2P) Botnet Team

Defense teams are represented by the following common classes of agents (Figure 4): information processing (sampler); attack detection (detector); filtering (filter); investigation (investigator); rate limitation (limiter). Samplers collect and process network data for anomaly and misuse detection. Detector coordinates the team, correlates data from samplers, and detects attacks. Filters are responsible for traffic filtering using the rules provided by detector.

Investigator tries to defeat attack agents. Limiter is for implementation of cooperative defence. Its local goal is to limit the traffic according to the team goal. It lowers the traffic to the attack target and allows other agents to counteract the attack more efficiently.

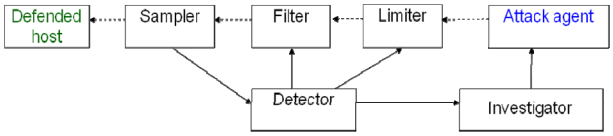

Figure 4: Defense team

To generate a *legitimate traffic*, "user" and "server" agents are determined. These agents generate the traffic statistically similar to traffic of standard user computers and web servers.

These models are implemented in the simulation environment as a sequence of internal abstraction layers (Figure 5): (1) discrete event simulation on network structures, (2) computational network with packet switching, (3) network services, (4) botnet and defense agent teams. Specification of every subsequent layer is an extended specification of the previous one.

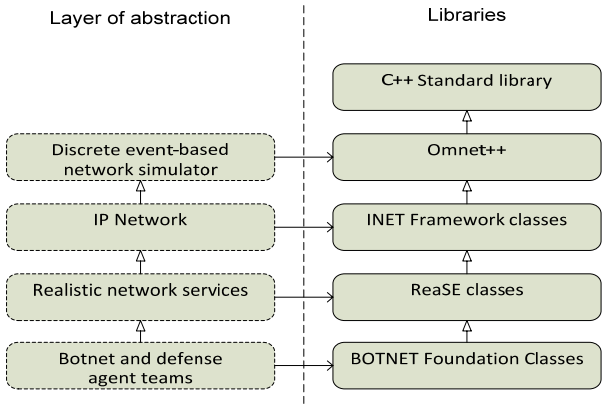

Figure 5: Simulation environment (tool) architecture

The simulation environment relies on the implemented libraries and the third party libraries. Functional purpose of each library matches to the appropriate layer of abstraction. All components of the simulation environment are implemented in C++ programming language with standard runtime libraries. Each particular library provides a set of modules and components which are implementations of entities of appropriate semantic layer. Any given library can rely on the components exported by the libraries of the previous layer and can be used as a provider of components needed for the subsequent layer implementation. The first layer of abstraction is implemented by use of the discrete event simulation environment OMNET++ (Varga 2010). The OMNET++ provides the tools for simulation of network structures of different kinds and processes of message propagation in them. The library INET Framework (INET Framework 2013) is used for simulation of packet-switching networks. This library provides components implemented as OMNET++ modules and contains large variety of models of network devices and network protocols. Simulation of realistic computer networks is carried out by using the library ReaSE (ReaSE 2013).

An example of the user interface of the simulation environment is shown in Figure 6. In the upper left corner you can see the main window that displays the components included in the model and control elements that allow users to interact with them. In the lower left corner the network configuration window is outlined. The main window also includes controls for managing the model time (e.g., one can perform the simulation step-by-step or express mode). There are also controls for searching the entity of interest for editing its state. The structure of a network node is in the lower right corner. In the upper right corner (Figure 6) the window of parameters is depicted. There are the following specification elements to define the investigated network models, attack and defence mechanisms:

- Network topology: quantity and types of hosts, channels between them and their types.
- Botnet team: quantity of bots; botmaster's address and port used for interactions; bot's ports used to send attack packets; victim's address and port; time of attack; attack intensity; address spoofing technique.
- Attack: victim type (application, host or network); type of attack (brute force (UDP/ICMP flood, smurf/fraggle, etc.); attack rate dynamics (can be constant or variable); etc.
- Defense team: address of defended host; detector's address and port for interactions; server's reply size and delay time; adaptation scheme depending on attack severity, etc.
- Defense: deployment location; the stages the defence method can implement (attack prevention, attack detection, tracing the attack source, attack counteraction); attack detection technique (misuse and anomaly detection; etc.
- User team: quantity of users; server's address and port; time to start; quantity of requests to server, interval between them and their size; interval between connections.
- Simulation: simulation duration; quantity of experiments; initialization of random number generator.

4. EXPERIMENTS

On the top level the network topology is simulated on the level of the autonomous systems (AS) where the technique of positive-feedback preference (PFP) (Zhou et al. 2006) is used to model the network topology.

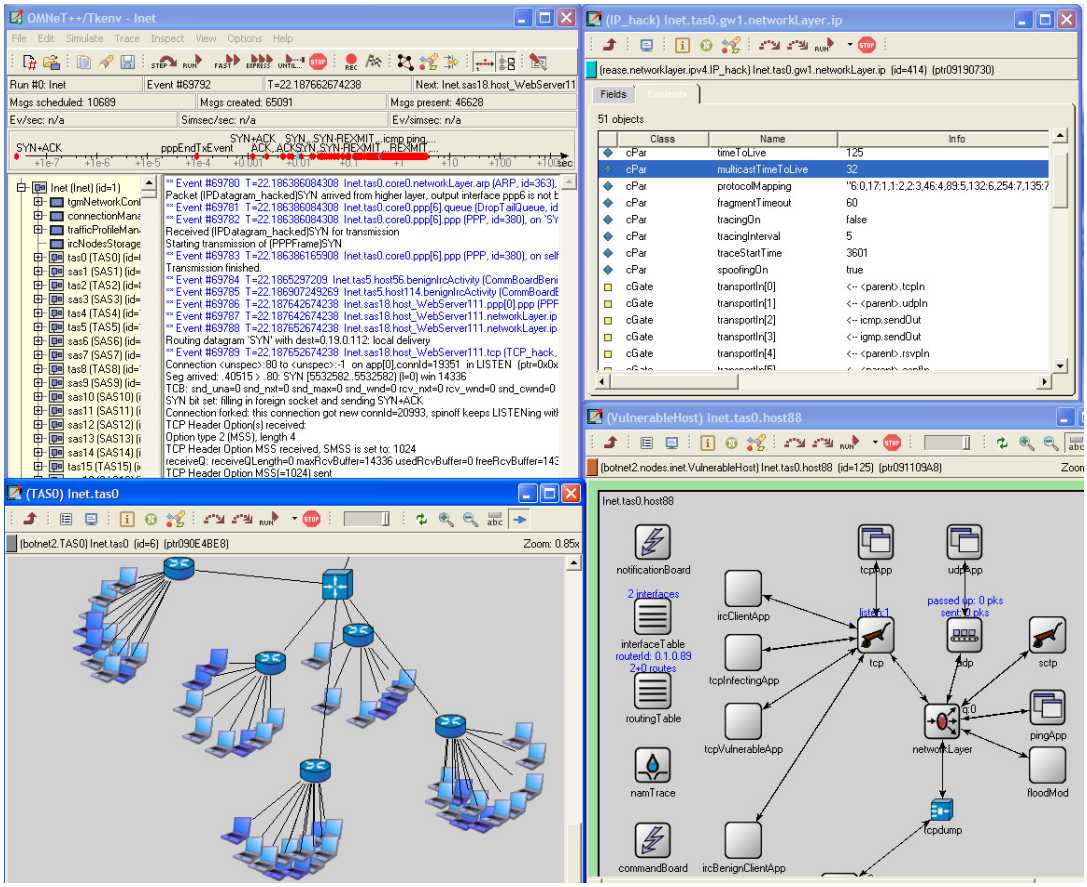

Figure 6: User interface of the simulation environment

On the lower level for each AS the router-level topology is simulated by using the HOT-model (Heuristically Optimal Topology) (Li et al. 2004).

The experiments with the agent-based simulation environment demonstrated the operability of the developed simulation environment and main characteristics of botnets and defense mechanisms investigated. Experiments investigated botnet actions and defense mechanisms on stages of botnet propagation, botnet management and control (reconfiguration and preparation to attacks), and attack execution.

We analysed several techniques, including Virus Throttling (Williamson 2002) and Failed Connection (Chen and Tang 2004), to protect from botnet on the propagation stage. Botnet propagation was performed via network worm spreading. We researched techniques of IRC-oriented botnet detection to counteract botnets on the management and control stage. We also analyzed techniques which work on the different stages of defense against DDoS attacks. These techniques include SAVE (Source Address Validity Enforcement Protocol) (Li et al. 2002), SIM (Source IP Address Monitoring) (Peng et al. 2004) and Hop-count filtering (Jin et al. 2003).

Let us consider only examples of experiments on the stage of botnet management and protection against botnet on this stage. The example of the user interface of the simulation environment during these experiments is shown in Figure 7 which depicts different fragments of the network, and bots are darker.

Let us describe the usage only one of the protection technique which was proposed by M. Akiyama et al. (Akiyama et al. 2007). This technique involves monitoring of IRC-traffic, passing through the observer node, and subsequent calculation of the metrics “Relationship”, “Response” and “Synchronization”, based on the content of network packets. The metric “Relationship” characterizes the distribution of clients in IRC-channel. Too high value of this metric is considered as abnormal. The metric “Response” is calculated as the distribution of response time to the broadcasting request. The metric “Synchronization” characterizes the synchronism in the behaviour of IRC clients.

The IRC traffic was monitored by using sampler agents, installed on the core routers of large network segments. Information about IRC channel and its clients is defined by analysis of IRC packets. Then, based on data obtained, the relationship metrics of observed channels were calculated in real time. It is assumed that the data,

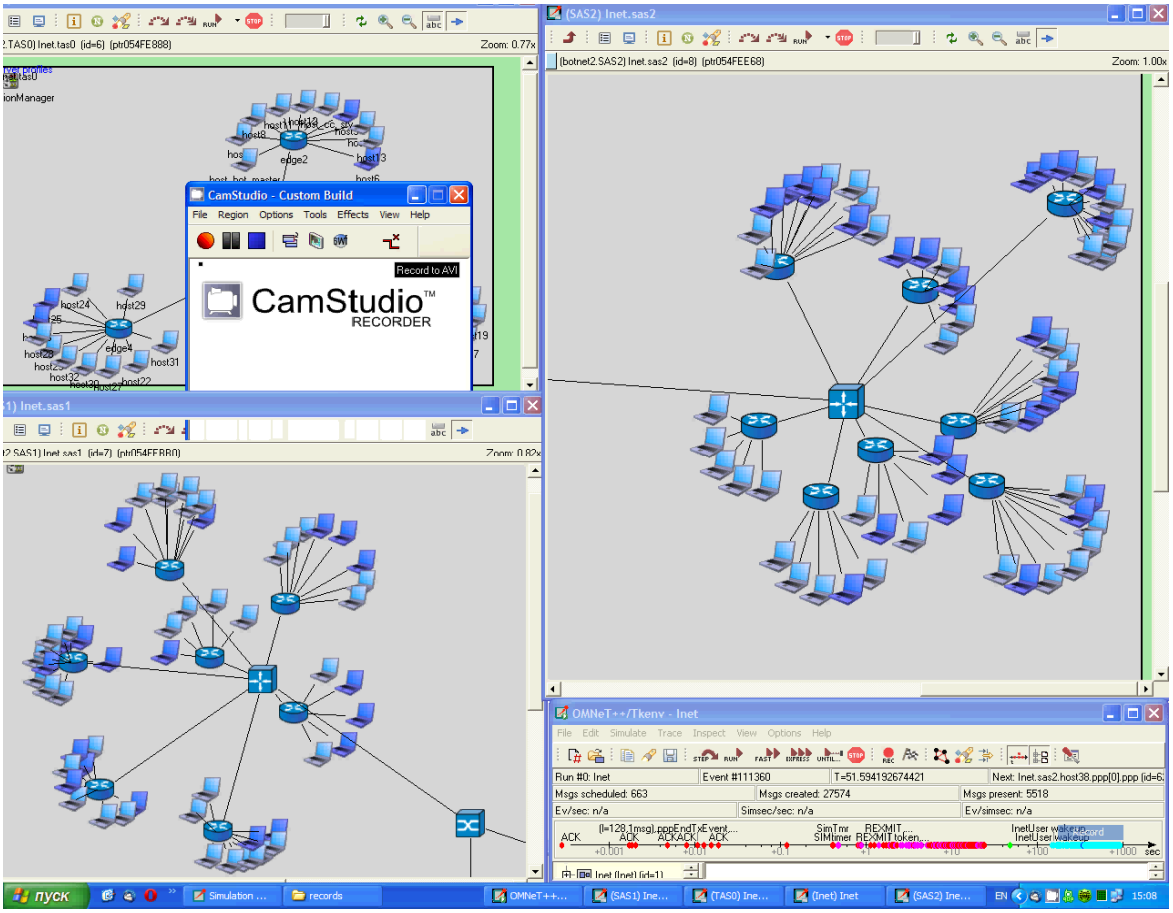

Figure 7: User interface of the simulation environment experiments on the stage of botnet management

obtained from sampler agents, will strongly depend on the location of sampler agents in relation to main IRC flows, merging near the network segment that contains the IRC server.

Table 1 shows a part of observed relationship metrics for IRC channels in various network locations. There are data for the botnet control channel (Irc-bot) and the channel for legitimate IRC communication (Irc-1). The number of clients in the Irc-1 channel is ten. For the legitimate channels, either all the participants are detected or none of the participants are detected. This is because the legitimate IRC communication is performed by exchanging broadcast messages; therefore, if an observer resides on the way of the IRC traffic, it detects all the clients of the corresponding channel. For the control channel Irc-bot, there is significant differentiation of the observed metric depending on the location of sampler agents. This is due to the features of botnet client communication in the IRC channels. Rather than broadcasting messages to all the channel participants, the botnet nodes exchange information only with a small number of nodes belonging to the set of botmaster nodes. It is seen from Table 1 that there are two routers on which the botnet control channel was detected almost completely. The analysis of the network topology showed that the IRC server was located in the segment sas17, while the segment tas0, which was in the close vicinity of sas17, operates as a transit segment between the IRC server and the greater part of the bot clients.

Table 1: Relationship Metrics of IRC Channels

#Sensor	#Irc-bot	#Irc-1
sensor_sas17	97,91%	100,00%
sensor_tas0	95,82%	100,00%
sensor_tas4	26,82%	100,00%
sensor_sas18	7,27%	0,00%
sensor_sas26	5,45%	100,00%
sensor_sas11	5,45%	0,00%
sensor_tas5	5,27%	0,00%
sensor_sas20	5,09%	100,00%
sensor_sas13	5,00%	0,00%

Analysis of the topology of the simulated network shows that the segment sas17 (sensor_sas17) has an IRC server node. The segment tas0, located in proximity to the segment sas17, is a transit for the traffic between the IRC server and the most of IRC bots.

Thus, we can suppose that a defense agent, located on a small number of routers which are transit for the main IRC traffic, can be as effective as the defense agent installed in more number of routers. We can also assume that a defense agent, having a small covering of the protected network, generally will not be efficient,

because only a small part of the IRC control traffic passes the vast majority of routers.

In the experiments the IRC traffic was monitored in different network locations. Based on monitoring results, the synchronization metrics were calculated. Let us consider the synchronization metrics determined by monitoring the traffic on the core router of network segment tas0 (Figure 8).

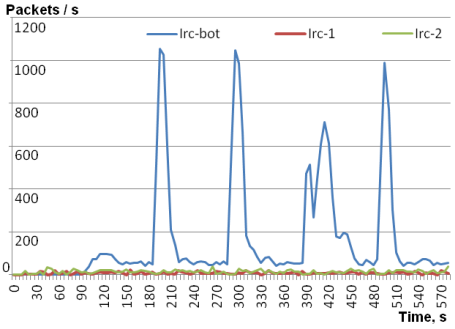

Figure 8: Synchronization Metrics for tas0

From 200 seconds of simulation time, every 100 seconds we can observe sharp spikes of traffic volume related to the botnet control IRC channel. These bursts are caused by response messages from bots on a request from the botmaster.

The network segment tas0 is located in proximity from the network segment which includes the IRC server. Thus, a significant part of the IRC control traffic is transmitted through the router of the network segment tas0. For this reason, the bursts of control channel traffic are expressed against the traffic of legitimate communication.

The traffic on the network segment router sas13 was measured (Figure 9) to evaluate the impact of the proximity of the sampler agent from the IRC server on the severity of bursts of control traffic (and thus on the discernibility of synchronization metric). Traffic measurements show a general decrease of traffic level in the observation point sas13, as well as a good visibility of traffic spikes on the core router of this network segment. Thus, the results of experiments demonstrate the applicability of synchronization metric to detect the IRC control traffic.

The filtering method, based on the relationship metric, used in agents-filters, uses an assumption that the IRC channels with a very large number of clients are anomalous. We carried out a series of experiments where the relationship metric was used for different configurations of filtering components and different critical levels of relationship. It was shown that the efficiency of the IRC traffic detection and filtering, based on the relationship metric, increases sharply when the routers, which are transit for the IRC control traffic, are fully covered by filters.

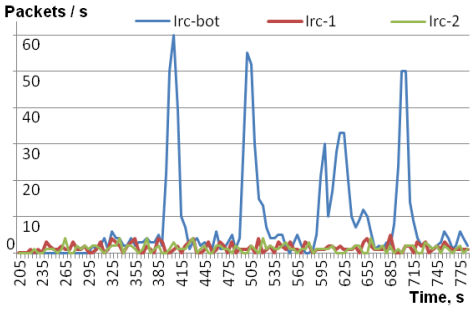

Figure 9: Synchronization Metric for sas13

The filtering method, based on the synchronization metric, uses an assumption that the short synchronous messaging in a single IRC channel is anomalous. The observed synchronization metric is calculated as the number of IRC packets, passing through the samplers for a fixed period of time. In the experiments fulfilled, the filtering criterion is a fivefold increase in traffic for 20 seconds followed by a return to its original value. The results of experiments allow concluding about low quality of the method in the current configuration, since false positive rate has a rather high value.

5. CONCLUSION

In the paper, an agent-based simulation approach for investigating botnets and defense mechanisms against them was proposed. The general architecture of the agent-based environment for the simulation of botnets and defense mechanisms against them was presented. This architecture was implemented on the basis of the discrete events simulation system OMNeT++, and several specialized libraries implementing the models of botnets and defense agent teams. Experiments were performed that demonstrate the behavior of botnets and defense agent teams at the phases of propagation, management and control, and attack. The paper describes only the experiments at the phases of management and control. The experiments confirmed the practical usefulness of the proposed approach for the simulation of complex botnets and the analysis of security of network segments. In future research we are planning to fulfill a comprehensive analysis of effectiveness of botnets and defense agent teams operation, and futher improvement of the implemented simulation environment.

ACKNOWLEDGEMENT

This research is being supported by grant # 13-01-00843 of the Russian Foundation of Basic Research, Program of fundamental research of the Department for Nanotechnologies and Informational Technologies of the Russian Academy of Sciences (contract #2.2), State contract #11.519.11.4008 and the EU as part of the SecFutur and MASSIF projects.

REFERENCES

Agmon, N.; S. Kraus; and G. A. Kaminka. 2011. "Multi-Robot Patrol in Strong Adversarial Environments." *Journal of Artificial Intelligence Research*, 42.

Akiyama, M.; T. Kawamoto; M. Shimamura; T. Yokoyama; Y. Kadobayashi; S. Yamaguchi. 2007. "A proposal of metrics for botnet detection based on its cooperative behavior." In *Proceedings of SAINT Workshop.*

Bailey, M.; E. Cooke; F. Jahanian; et al. 2009. "A Survey of Botnet Technology and Defenses," In *Proceedings of the Cybersecurity Applications Technology Conf. for Homeland Security*.

Chen, S. and Tang Y. 2004. "Slowing Down Internet Worms." In *Proceedings of the 24th International Conference on Distributed Computing Systems*.

Cohen, P. and H.J. Levesque. 1991. "Teamwork." *Nous*, 35.

Dagon, D.; G. Gu; C.P. Lee; et al. 2007. "A Taxonomy of Botnet Structures." In *Proceedings of ACSAC'07*.

Feily, M.; A. Shahrestani; S. Ramadass. 2009. "A Survey of Botnet and Botnet Detection". In *Proceedings of Third Int. Conf. on Emerging Security Information Systems and Technologies*.

Gamer, T. and C. Mayer. 2009. "Large-scale Evaluation of Distributed Attack Detection." In *Proceedings of the 2nd International Workshop on OMNeT++*.

Govil, J. and G. Jivika. 2007. "Criminology of Botnets and Their Detection and Defense Methods," In *Proceedings of of IEEE Int. Conf. on Electro-Information Technology*.

Grizzard, J.B.; V. Sharma; C. Nunnery; et al. 2007. "Peer-to-Peer Botnets: Overview and Case Study." In *Proceedings of the Workshop on Hot Topics in Understanding Botnets*.

Grosz, B. and S. Kraus. 1996. "Collaborative Plans for Complex Group Actions." *Artificial Intelligence*. 86, 2.

INET Framework. 2013. http://inet.omnetpp.org/

Kaminka, G.A.; A. Yakir; D. Erusalimchik; and N. Cohen. 2007. "Towards Collaborative Task and Team Maintenance." In *Proceedings of AAMAS-07*.

Jin, C.; H. Wang; K.G. Shin. 2003."Hop-count filtering: An effective defence against spoofed DDoS traffic." In *Proceedings of ACM Conference on Computer and Communications Security*.

Kotenko, I. and A. Ulanov. 2007. "Agent-based Simulation Environment and Experiments for Investigation of Internet Attacks and Defense Mechanisms." In *Proceedings of ECMS 2007*.

Kotenko, I. 2009. "Simulation of Agent Teams: the Application of Domain-Independent Framework to Computer Network Security." In *Proceedings of* ECMS 2009.

Kotenko, I. 2010. "Agent-Based Modelling and Simulation of Network Cyber-Attacks and Cooperative Defence Mechanisms." *Discrete Event Simulations*. InTech.

Kotenko I.; A. Konovalov; A. Shorov. 2012-1. "Agent-based simulation of cooperative defence against botnets." *Concurrency and Computation: Practice and Experience*, Vol. 24, Issue 6.

Kotenko I.; A. Konovalov; A. Shorov. 2012-2. "Discrete-Event Simulation of Botnet Protection Mechanisms." *Discrete Event Simulations - Development and Applications*. InTech.

Krishnaswamy, J. 2009. *Wormulator: Simulator for Rapidly Spreading Malware*, Master's Projects.

Li, J.; J. Mirkovic; M. Wang; P. Reither; L. Zhang. 2002. "Save: Source address validity enforcement protocol." In *Proceedings of IEEE INFOCOM*.

Li, L.; D. Alderson; W. Willinger; J. Doyle. 2004. "A first-principles approach to understanding the internet router-level topology." *ACM SIGCOMM Computer Communication Review*.

Owezarski, P. and N. Larrieu. 2004. "A trace based method for realistic simulation." In *Proceedings of 2004 IEEE International Conference on Communications*.

Naseem, F.; M. Shafqat; U. Sabir; et al. 2010. "A Survey of Botnet Technology and Detection." *Intern. Journal of Video & Image Processing and Network Security*, 10, 1.

Peng, T.; C. Leckie; K. Ramamohanarao. 2004. "Proactively Detecting Distributed Denial of Service Attacks Using Source IP Address Monitoring." *Lecture Notes in Computer Science*, 3042.

ReaSE. 2013. https://i72projekte.tm.uka.de/trac/ReaSE

Riley, G.; M. Sharif; W. Lee. 2004. "Simulating internet worms." In *Proceedings of the 12th International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems.*

Schuchard, M.; A. Mohaisen, D. Kune; N. Hopper; Y. Kim, E. Vasserman. 2010. "Losing control of the internet: using the data plane to attack the control plane." In *Proceedings of the 17th ACM Conference on Computer and communications security*.

Simmonds, R.; R. Bradford; B. Unger. 2000. "Applying parallel discrete event simulation to network emulation." In *Proceedings of the fourteenth workshop on Parallel and distributed simulation*.

Stone, P.; G. A. Kaminka; S. Kraus; and J. S. Rosenschein. 2010. "Ad hoc autonomous agent teams: Collaboration without pre-coordination." In *Proceedings of the* AAAI'10.

Suvatne, A. 2010. *Improved Worm Simulator and Simulations*. Master's Projects.

Tambe, M.; E. Bowring; H. Jung; et al. 2005. "Conflicts in teamwork: Hybrids to the rescue." In *Proceedings of AAMAS-05*.

Taveter, K.; M. Parmak; and M. Meriste. 2010. "Agent-oriented modelling for simulation of complex environments." In *Proceedings of IMCSIT 2010*.

Varga, A. 2010. "OMNeT++." Chapter in the book *Modeling and Tools for Network Simulation*. Springer Verlag.

Wang, P.; S. Sparks; C.C. Zou. 2007. "An Advanced Hybrid Peer-to-Peer Botnet." In *Proceedings of the First Workshop on Hot Topics in Understanding Botnets*.

Wehrle, K.; M. Gunes; J. Gross; 2010. *Modeling and Tools for Network Simulation*. Springer-Verlag.

Williamson, M. 2002. "Throttling Viruses: Restricting propagation to defeat malicious mobile code." In *Proceedings of ACSAC Security Conference*.

Zhou, S.; G. Zhang; G. Zhang; Z. Zhuge. 2006. "Towards a Precise and Complete Internet Topology Generator." In *Proceedings of Intern. Conference on Communications.*

AUTHOR BIOGRAPHY

IGOR KOTENKO is Professor of computer science and a head of the Laboratory of Computer Security Problems in St. Petersburg Institute for Informatics and Automation of the Russian Academy of Science. His research interests include simulation, multi-agent systems and computer network security.