Attacking and Defending with Intelligent Botnets

Moises Danziger and Marco Aurelio Amaral Henriques

***Abstract*— Machine learning (ML) has been seen as a great ally of security. All his potential to automate actions with some level of intelligence has called the attention of industry which is using it on security systems. However, attackers have also noted all ML potential. In a first moment, attackers have tried to fight ML-based security tools through the study and exploitation of weak points in ML techniques. It is named as adversarial machine learning. Besides this first application, someone could apply a ML-based tool directly against a security system. It is the case of intelligent botnets - a different type of botnet made of relatively intelligent bots which can take decisions by their own during the attack. So, in this work we are making a reflection on the future of botnets within the context of ML and showing that this kind of botnet could break the current detection approaches. We also point out to the need for creating new approaches to combat bots with some kind of intelligence. Moreover, we propose a theoretical model of intelligent bots, their possible impacts and combat strategies.**

***Keywords*— Botnets; Machine Learning; Autonomous systems, Intelligent Agents**

I. Introduction

Botnets are one of the most feared virtual threats after the Internet advent. It brings several dangerous problems for users and systems. Nowadays most botnet attacks are successful due to their evasion abilities and, to improve defense techniques, researchers and the security industry presented a large number of detection methods as can be seen in [1]. Most of these works have presented some good results and some of them might be used in the real world. However, fighting botnets is a complex task and many approaches have been tried, but restricted to laboratory experiments [2].

At the same time, works as those of Bijalwan et al. [3] and Karim et al. [4] presented some challenges, trends and possible applications for botnets in the future. Currently we already can see botnets over IoT (Internet of Things) [5], smartphones (mobile botnets) [6] and cloud environments (botclouds) [7]. These approaches showed us some technological trends in botnet actions. Looking at such trends we can see that the botnet's complexity has been growing as new technologies are appearing in the world. If we pay attention to the technology advances as, for example, cognitive computing [8] and deep learning [9], we can note that there are new possibilities in the botnet area. In this scenario, a new botnet characteristic can be the autonomy of bots. It means that bots could decide by themselves what must be done according to the context, targets and mission. It could be done through strong abilities to do reconnaissance on the environment and through an embedded intelligent process that helps the decision taking.

Moises Danziger and Marco Aurelio Amaral Henriques Faculty of Electrical Engineering and Computation, University of Campinas (UNICAMP), Campinas-SP, Brazil, E-mails: (danziger,marco)@dca.fee.unicamp.br

The first consequence of adding intelligence to botnets is the new type of operation that becomes possible. Now the bots can limit or break their communication with the botmaster or command centers all the time or, in the simplest scenario, they just communicate when the mission is accomplished. This can be disastrous for botnet detection because many detection approaches were created based on the comand & control (C&C) communication process between the bots and the command centers or botmasters. If this process is eliminated or limited, the detection efficiency will probably decrease. But this new model is not so straightforward. It needs a deep change in present botnet concepts and the first actor to look at in this case is the botmaster. He will need new knowledge and skills.

In this work we present a discussion about the impact of attacks applying methods based on ML to botnets with bots that can take decisions on-the-fly. We also present a theoretical model based on multi-agent systems (MAS). Our objective is to antecipate the discussion about the impact of this new kind of botnet and the different modus operandi over the current detection systems.

We call attention to the fact that we are not discussing about the adversarial machine learning as showed by Huang et. al. [10]. Some discussions are showing the high impact of sophisticated adversaries that frequently attempt to break the training processes by, for example, crafiting input data that has similar feature properties to normal data as seen in [11] and [12]. In these cases, the attackers are trying to evade ML-based classifiers systems by exploiting weak points. They are not applying ML to the attack, as we are discussing here.

Navigating through this paper we can see, in the second section, the current botnet research efforts and some results from the application of ML techniques in this area. Following, in section three we present a theoretical model of intelligent bots. Section four presents a discussion about the impacts of the model and possible ways to combat the new menace it imposes. Finally, in sections five and six we draw some conclusions and discuss possible future works.

II. State of Art

Chen and Ji [13] presented an approach of a self-learning worm using importance scanning. It is well known that to spread worms, the developers employ distinct scanning methods as, for example, topological, random, localized, hitlist, routable and the importance scanning. The authors contend that in the Internet it may not be easy for attackers to collect information on vulnerable hosts. So, they pointed to future worms that can become more intelligent and potentially learn a certain knowledge about the vulnerable hosts while propagating. According to the article, the key capability of the worm

is to learn an underlying vulnerable host group distribution. So, the worm can collect information and estimate the group distribution. The results showed that the self-learning worm can spread far faster than a random-scanning, permutation-scanning and a Class A routing worm.

Castiglione et. al. [14] applied a swarm-based approach to improve botnet-based C&C infrastructures. They presented a methodology based on stigmergic communication models. A stigmergic system is a natural method in MAS that is based on collective agent's behavior according to a distributed intelligence paradigm [15]. Working as a set of agents that act together with their operating environment, they are described at every instant by a set of state variables. Stigmergic (from stigmergy) is a form of self-organization that can produce complex and seemingly intelligent structures without prior planning, control or even direct communication between the agents [16]. This is a characteristic of social agents in an ant colony that was applied by Castiglione et. al. in the building of a P2P-based C&C infrastructure. With it, the authors facilitate the interconnection among bots spread over the network. The results showed a C&C architecture more robust and scalable, a better collaboration and coordination environment for botmaster, improved fault tolerance and dynamic adaptation to varying network conditions. Their work gives an idea about the power of ML if applied to develop more efficient botnets. Yet, because of the stigmergy, the agents do not need to possess any previous memory, intelligence or coordination. Stigmergic cooperation significantly improves the infrastructure's ability to dynamically adapt to changes in network conditions, greatly enhancing fault tolerance, robustness, scalability and survivability.

Gaudesi et. al. [17] developed a new obfuscation mechanism based on evolutionary algorithms. Attackers use obfuscation to avoid reverse engineering in their malware/botnet code. They embedded an evolutionary core in the malware to generate a different, optimized hiding strategy for every single infection. It means that the malware program is able to evolve its own packer, creating a brand new encoding routine in each infection. The ML technique applied here is the genetic algorithm which is an approach to search and define the best values for the variables. Note that this kind of malware hiding technique can be a challenge for the security community.

III. A Theoretical Model of Intelligent Botnet

Imagine a group of intelligent agents in a MAS [18] that were created with a mission. They have autonomy to decide how they will perform their actions to reach the target and accomplish their mission. This means that they do not need the C&Cs or botmasters monitoring their activities. The botmaster has a target and prepares a group of agents to spread on the internet, and their mission is to infect and steal sensitive information from a specific target. Bots do not have contact with the C&C after deployment.

After the agents are spread on the network/hosts, a learning process is started and they will use it to take decisions according to the challenges they face in their way. It means that agents/bots have the ability to learn from the events that they experience in the environment and can decide the better way to accomplish their mission. The whole decision process is done without the botmaster knowledge. For example, imagine an agent (bot) in front of a firewall. It can create a plan to bypass the firewall using a strategy based on team game from Game Theory [19]. In this case, the agent could create new agents to do specific tasks and create a set of agents to be sacrificed for the sake of a goal. Now we describe the main components of this model: The Agents, The Operation Model and The Learning Process.

A. The Agents

There are four kinds of agents in our proposal.

1) Super Agent (SA): It is the main agent that can be seen as a local botmaster. SA is responsible for the intelligent part of the bot actions. He must create and maintain the environment map and the surface attack. To avoid databases, he can use simple structures to store the data used to help in the decision process. He can also create other agents to help him to perform his mission. In addition, he can also create fake agents to distract a defending system or some specific agent to attack an enemy botnet. The communication that does not have a coordinator agent is his best choice to avoid him be discovered. For the decision process, he needs to have more ability to work on-the-fly. Note that the agent's ability depends on the ability of the botnet creator in defining the best techniques to be embedded in the agent. For example, an agent can employ the bio-inspired optimization techniques, as for example, that implemented by [14], to discover the best way to do a coordinated attack with a set of agents. More details about the learning process of SA is given in the subsection about learning process.

2) Recon Agent (RA): This agent has two modules: communication and scanner. The first is similar to SA. The second means that this agent can do a lot of things as detecting services, operating system, open ports, defense systems and so on. It can also collect network information, location and other monitoring data. It is very important that this agent is able to do its activities secretly to avoid data exfiltration techniques. One of its most important tasks is updating the environment map.

3) Defense Agent (DA): During the actions, the botnet needs to act in a stealthy way to avoid being detected. It means that the botnet needs to follow some strategy to defend its perimeter. So, the DA is created to prepare the environment against attacks from other botnets and protects the perimeter to avoid defense systems. It can act alone or in group and can also help to aggregate information about the attack surface. It has two modules: communication and defender. The second module is used to detect any attack against the intelligent bots done, for example, by other botnets and/or worms. It is also responsible for the detection of any attempt of defending tools to capture de bot core codes.

4) Attack Agent (AA): It is responsible for exploiting vulnerabilities, and opening the way to the botnet movement. Sometimes, depending on the situation, the attack agents could be created simply to fool the enemy.

B. The Operation Model

Figure 1 shows a macro vision of the proposal. This figure represents only one intelligent bot. Note that the attacker can spread the malware and infect a lot of devices and each one will be a new intelligent bot. In this first version the bots do not have any external communication. But, each bot is controled by the SA agent and it has an id that is used when it meets another intelligent bot during its lifetime (generaly in another already infected device). Once identified as a friend they can exchange knowledge. This is done using messages in a specific languages used by intelligent agents.

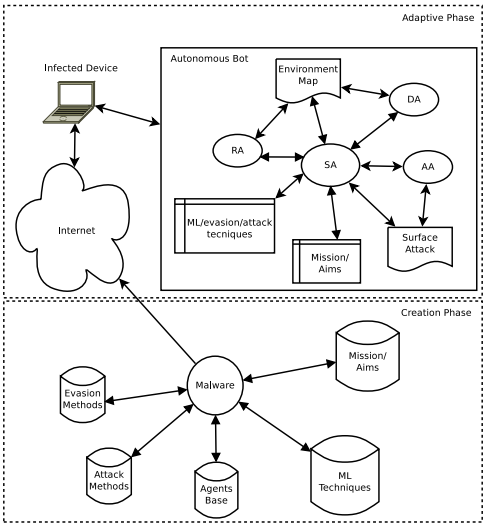

Fig. 1. The creation and adaptive phases in the prototype botnet.

In Figure 1, the first phase is named Creation. There, the botmaster chooses a set of information to create the agents with their rules and roles. They are part of the malware and it will be used during the infection phase. The botmaster also needs to choose the ML techniques that will be used by the SA in the decision module and a set of pre-configured data to be added to the information module. Defense and attack methods are also added into this module. These methods will be used by the SA on-the-fly. According to the map and the attack surface, the SA can decide the best way to go ahead with its mission. During the SA creation, the botmaster uses a set of information with possible scenarios that the agent might experience during its mission. Certainly, the chosen ML technique could impact the efficiency of SA and botnet and the training can impose a high cost to intelligent bots. So, the botmaster needs to be carefull.

After a successful infection, the agents are awakened. There are two maps that the SA will create: environment map and attack surface map. These two maps will be fed and used by SA, DA and AA. For example, the RA collects data from the environment as network and system features, DA inspects the environment looking for defense systems, other botnets or/and intelligent bots (enemy). Besides attacking, in a version with less restriction for agent communications, the AA could also give feedback on his attack result (success or failure). In this case, all information are exchanged using the direct message model of intelligent agents. There is no coordinator in this process. As the brain of the intelligent bot, the SA can also be seen as a local botmaster and implements the concept of mini-botnets as showed in the work of Al-Bataineh and White [20]. Yet, he can create other agents (RA, DA, AA) to help during the attack or defence phases.

The pseudocode below shows the intelligent bot operation algorithm. There is an initial infection process (the kind of infection is not important for our analysis). After that, the bot is created and the agents are awakened. So, the SA starts the botnet control in a loop that is interrupted only when the mission is accomplished or in a auto-destruction process. The other agents are also awakened and begin their activities as defined by their roles. There is also a propagation module that spreads agents into the network.

Here, we are also introducing the concept of bot agents that are not strongly related with devices. The bot agent allows the botnets to avoid the dependency on devices and, so, they can be in any place or device. It will help the idea of mobile bots that can move among devices that were infected. Mobile bots allow the botnet to be more efficient and make progress towards the mission target. It also allows to have groups of mobile bots working together as a group of normal bots in the current botnets. It is different of worm replication since intelligent bots use the past experiences to take decisions.

C. The Learning Process

Learning is essential for an intelligent agent and ML techniques using reinforcement learning (RL) seems to be suitable to the proposed model [21]. Differently from supervised and unsupervised learning, RL has an online learning process. It means that it learns and acts simultaneously and it seems to be a good choice in an intelligent bot because he can experience new scenarios and get new kinds of data that need retraining. In this case, we need to decide the basic attributes of the RL adopted to give bots ability to take decisions. Behind RL there is the well-known Markov Decision Process (MDP), which is a tool for modeling sequential decision-making problems where a decision maker interacts with a system in a sequential fashion [22]. We need to define the MDP attributes: (i) Non-empty set of states, (ii) non-empty set of actions, (iii) the transition probability kernel and (iv) a reward function. For example, the states can be the position in the network infected by the agent. The set of actions can move the bot from the current device to another in the neighborhood, the reward can be based on the number of captured devices in a period of time or the time spent to infect a device. Anyway, the learning module is the most important for intelligent bots.

IV. Impacts of Intelligent Botnets

We can follow the work of Stinson and Mitchell [23] and look at some botnet characteristics upon which automated detection methods rely to think about some impacts on detection systems. We can see that our proposed botnet model can impact detection methods that rely on traffic statistics from network communication between the C&C and the bots. The agents (bots) could calculate the statistics and act to

```
Data: mission,violated = 0
Function InitialInfect()
    start infect;
    agents ← extractAgents;
    CreateBot(agents);
Function CreateBot (agents)
    wakeup agents.SA, agents.RA, agents;DA, agents.AA;
    ControlBotnet(SA,mission);
Function ControlBotnet(SA,mission)
    if violated = 0 then
        continue;
    else
        Destroy(bot);
    end
    while mission not reached do
        env ← ScanEnvironment(RA);
        attackSurface ← Learning(env);
        Propagate(AA,attackSurface);
    end
Function ScanEnvironment(RA)
    scan(Interfaces,Hosts,Apps,Defences);
Function Learning(env)
    evaluate(env,risks);
    create/update attackSurface;
Function Propagate(AA,attackSurface,hostChange)
    attackResult ← attackSurface.aims;
    if attackSurface.aims = attackSurface.mission and
     attackResult = true then
        mission ← reached;
        hostChange ← 0;
    else
        SendAgent(agent);
    end
Function SendAgent(agent)
    sendAgent ← agent;
Function Destroy(bot)
    destroy bot;
```

Algorithm 1: Intelligent Bot Operation Algorithm.

avoid anomalies. Also, methods that rely on flow-charts can be impacted because there is no C&C in our proposal and the communication among agents can be done using a local-based network structure or through the ports 80 or 443 using HTTP. Methods that rely on syntax can also be impacted when cryptography is used in the agent's communication. Our proposal could also impact the methods that rely on observing traffic since the agents can act alone without any communication. For example, it can be difficult to do information flow tracking in communication flow because there is no RP (rendezvous point). Methods that rely on automated, network-based detection of botnet attacks (such scanning) can also be impacted because the agents can choose which subset of attacks will be performed. Furthermore, those that rely on cross-host clustering can also be impacted because the bots can participate in different attacks.

Certainly, the communication can be the weak point of our model because a system could discover the agents. To mitigate it we can apply evasion techniques to avoid the defence system. For example, we can apply DGA to stablish rendezvous points (it can be considered heavy for a simple agent but not for an expanded botnet with several intelligent bots working in a collaborative mode). So, for example, the communication model among agents can be done with P2P-based network techniques.

The concept of a self-learning worm from Chen and Ji [13] is very interesting for our approach. In fact, the idea of an intelligent worm is similar to an intelligent bot. However, the main difference between our approach and that is the ability of an agent to work in a collaborative way. In addition, the concepts of mobile bots and the super agent as a local botmaster are other important differences.

The intelligence behind ML techniques has a number of attractive features for military systems and they can call the attention of many governments. We also point out that this kind of botnet can have military focus because it could be an army of virtual soldiers. It can also be used for cyber espionage where the agents would be responsible for discovering sensitive information about, for example, countries, industry and arms.

Going to the bottom, we could extrapolate the concept of intelligent bots and think about botnets with autonomy. This is the next step of our research and brings more challenges. Yet, we can also think in hybrid botnets where there are intelligent bots and common bots. In this case the intelligent bots could be implemented to defend those common bots from being captured or stopped. Certainly, all of these conjectures are very interesting and we are planning simulations to show the autonomy concepts applied to botnets in several scenarios.

V. Combating Intelligent Bots

Probably, the most efficient way to combat an automous botnet is an approach that also uses some kind of intelligent agents. There are some MAS-based detection approaches as those presented by Pomorova et al. [24] and Savenko et al. [25]. In this way, we could create a MAS-based approach to show a possible combat method against intelligent botnets. Our work is different from that proposed by Kotenko et. al. in [26] because they do not consider intelligent bots nor detection models for them.

Another approach could be that from Salloum and Wolthusen [27] where the authors proposed a semi-intelligent link layer vulnerability discovery to operate in networks. They developed an agent-based (vulnerability) detection mechanism using semi-intelligent propagation strategies as the self-replication from worms. The idea is to reconstruct the topology information found through the link layer discovery protocol to detect neighboring nodes and propagate gradually until total coverage of an enterprise network is reached. No ML technique was implemented in that work and a possible proposal is adding it to improve the ability of those agents and force them to work without any kind of human interation. This idea could be implemented to help agents to defend network perimeters.

It is possible to apply the nematodes idea from Dave Aitel [28]. Nematodes, good, beneficial or benevolent are names for a kind of worm created to combat other worms. Although the concept was used before by rival groups (attackers), it is a very controversial topic as showed by Salloum [29]. In [30] Salloum showed the Seawave, the first compute worm that utilize the second layer of the OSI model (Data Link Layer) as its main propagation way. This worm is a controlled interactive, self-replicating, self-propagating, and self-contained vulnerability mitigation mechanism. The author defined the taxonomy of viruses, worms and botnets. For each one, he point to a defensive model, so, we can have defensive viruses, worms and botnets. Maurushat [31] also did a benevolent worm design and a long discussion about the ethical and legal analysis of this approach to combat worms.

We noted that no learning process was added in those works with nematodes, so, we could apply intelligence on the nematode worms and give them more autonomy to fight against the intelligent bots. In this way, the intelligent bots would have a enemy in the same level of abilities.

VI. Conclusions

The botnet model discussed here is completely different from previous botnets because the C&C communication channel is eliminated or limited. In fact, because the intelligent bots have abilities to decide their next steps, the botmaster could not have online control over the botnet operation. As a result, the current botnet detection approaches that are based only on network analysis should be reviewed. We presented a theoretical model based on MAS that is used as a basic model to typify botnets with intelligent bots. According to our analysis, we can point to a dangerous evolution on botnets which can cause high impact on security systems. On the other hand, we call attention to the fact that for combating this new kind of botnet new tools will be necessary and intelligent techniques, as ML, can be a valuable asset on their development. Finally, we strongly recommend security researchers to review their security systems and applications and apply a threat modeling to evaluate the impact of attacks by tools and methods using ML. In the future, ML can be the best ally to fight more dangerous treats as intelligent botnets and we have to be prepared to better understand and use it.

References

[1] S. S. C. Silva, R. M. P. Silva, R. C. G. Pinto, and R. M. Salles, "Botnets: A Survey," *Comput. Netw.*, vol. 57, no. 2, pp. 378–403, Feb. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2012.07.021

[2] S. Soltani, S. A. H. Seno, M. Nezhadkamali, and R. Budiarto, "A Survey On Real World Botnets And Detection Mechanisms," *International Journal of Information and Network Security*, vol. 3, no. 2, pp. 116–127, 2014.

[3] A. Bijalwan, M. Thapaliyal, E. S. Piili, and R. C. Joshi, "Survey and Research Challenges of Botnet Forensics," *International Journal of Computer Applications*, vol. 75, no. 7, pp. 43–50, August 2013.

[4] A. Karim, R. B. Salleh, M. Shiraz, S. A. A. Shah, I. Awan, and N. B. Anuar, "Botnet detection techniques: review, future trends, and issues," *Journal of Zhejiang University SCIENCE C*, vol. 15, no. 11, pp. 943–983, 2014.

[5] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[6] A. Karim, S. A. A. Shah, and R. Salleh, "Mobile botnet attacks: a thematic taxonomy," in *New Perspectives in Information Systems and Technologies, Volume 2*. Springer, 2014, pp. 153–164.

[7] M. Torkashvan and H. Haghighi, "cbc2: a cloud-based botnet command and control," *Indian Journal of Science and Technology*, vol. 8, no. 22, 2015.

[8] D. S. Modha, R. Ananthanarayanan, S. K. Esser, A. Ndirango, A. J. Sherbondy, and R. Singh, "Cognitive computing," *Communications of the ACM*, vol. 54, no. 8, pp. 62–71, 2011.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[10] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. Tygar, "Adversarial machine learning," in *Proceedings of the 4th ACM workshop on Security and artificial intelligence*. ACM, 2011, pp. 43–58.

[11] B. Biggio, G. Fumera, and F. Roli, "Security evaluation of pattern classifiers under attack," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, pp. 984–996, 04/2014 2014.

[12] B. Nelson, B. Biggio, and P. Laskov, "Understanding the Risk Factors of Learning in Adversarial Environments," in *4th ACM Workshop on Artificial Intelligence and Security (AISec 2011)*, Chicago, IL, USA, October 2011, p. 87–92.

[13] Z. Chen and C. Ji, "A self-learning worm using importance scanning," in *Proceedings of the 2005 ACM workshop on Rapid malcode*. ACM, 2005, pp. 22–29.

[14] A. Castiglione, R. De Prisco, A. De Santis, U. Fiore, and F. Palmieri, "A Botnet-based Command and Control Approach Relying on Swarm Intelligence," *J. Netw. Comput. Appl.*, vol. 38, pp. 22–33, Feb. 2014. [Online]. Available: http://dx.doi.org/10.1016/j.jnca.2013.05.002

[15] E. Bonabeau, "Editor's Introduction: Stigmergy," *Artificial Life on Stigmergy*, vol. 5, no. 2, pp. 95–96, 1999.

[16] L. Marsh and C. Onof, "Stigmergic epistemology, stigmergic cognition," *Cognitive Systems Research*, vol. 9, no. 1–2, pp. 136–149, 2008, perspectives on Social Cognition. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389041707000290

[17] M. Gaudesi, A. Marcelli, E. Sanchez, G. Squillero, and A. Tonda, "Malware Obfuscation through Evolutionary Packers," in *Proceedings of the Companion Publication of the 2015 on Genetic and Evolutionary Computation Conference*. ACM, 2015, pp. 757–758.

[18] S. Russell, P. Norvig, J. Canny, and I. Bratko, *Artificial Intelligence: A Modern Approach*. Pearson Education, Limited, 2005. [Online]. Available: https://books.google.com.br/books?id=52fMAQAACAAJ

[19] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Basar, and J.-P. Hubaux, "Game Theory Meets Network Security and Privacy," Tech. Rep., 2010.

[20] A. Al-Bataineh and G. White, "Information Loss in Enterprise Networks: mini-botnets," *ISSA Journal*, vol. 9, no. 2, pp. 36–40, Feb. 2011.

[21] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[22] M. L. Puterman, "Chapter 8 Markov decision processes," in *Stochastic Models*, ser. Handbooks in Operations Research and Management Science. Elsevier, 1990, vol. 2, pp. 331–434. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0927050705801720

[23] E. Stinson and J. C. Mitchell, "Towards Systematic Evaluation of the Evadability of Bot/Botnet Detection Methods." *WOOT*, vol. 8, pp. 1–9, 2008.

[24] O. Pomorova, O. Savenko, S. Lysenko, and A. Kryshchuk, "Multi-agent Based Approach for Botnet Detection in a Corporate Area Network Using Fuzzy Logic," in *Computer Networks*. Springer, 2013, pp. 146–156.

[25] O. Savenko, S. Lysenko, and A. Kryschuk, "Multi-agent based approach of botnet detection in computer systems," in *Computer Networks*. Springer, 2012, pp. 171–180.

[26] I. Kotenko, A. Konovalov, and A. Shorov, "Agent-based modeling and simulation of botnets and botnet defense," in *Conference on Cyber Conflict. CCD COE Publications. Tallinn, Estonia*, 2010, pp. 21–44.

[27] Z. Al-Salloum and S. Wolthusen, "Semi-autonomous Link Layer Vulnerability Discovery and Mitigation Dissemination," in *IT Security Incident Management and IT Forensics, 2009. IMF '09. Fifth International Conference on*, Sept 2009, pp. 41–53.

[28] D. Aitel, "Nematodes–beneficial worms," *Black Hat Federal*, vol. 33, pp. 39–44, 2006.

[29] Z. S. Al-Salloum, "Defensive computer worms: an overview," *International Journal of Security and Networks*, vol. 7, no. 1, pp. 59–70, 2012.

[30] Z. S. Al-salloum, "Topology-Aware Vulnerability Mitigation Worms: Defensive Worms," 2012.

[31] J. Aycock and A. Maurushat, ""Good" Worms and Human Rights," *SIGCAS Comput. Soc.*, vol. 38, no. 1, pp. 28–39, Mar. 2008.