© Milov O., Yevseiev S. Methodology of modeling the behavior processes of antagonistic agents in security systems // Ukrainian Scientific Journal of Information Security, 2019, vol. 25, issue 3, pp. 150-161.

DOI: 10.18372/2225-5036.25.14460

METHODOLOGY OF MODELING THE BEHAVIOR PROCESSES OF ANTAGONISTIC AGENTS IN SECURITY SYSTEMS

Oleksandr Milov, Serhii Yevseiev

Simon Kuznets Kharkiv National University of Economics

MILOV Oleksandr, PhD, Professor

Year and place of birth: 1955, Zaporizhzhia, Ukraine.
Education: Moscow Energy Institute, 1978.
Position: Associate Professor of Cyber Security and Information Technology Department.
Scientific interests: decision theory, coordination in distributed systems, cryptography, agent-based modeling.
Publications: monographs −17, scientific papers – 85.
E-mail: Oleksandr.Milov@hneu.net.
Orcid ID: 0000-0001-6135-2120.

YEVSEIEV Serhii, Doctor of Technical Science, Senior Researcher

Year and place of birth: 1969, Kharcyzsk, Donetsk reg., Ukraine.
Education: Kharkov Military University, 2002.
Position: Head of Cyber Security and Information Technology Department.
Scientific interests: banking information security, cryptography of cryptographic codes.
Publications: monographs–12, scientific papers–115.
E-mail: serhii.yevseiev@hneu.net.
Orcid ID: 0000-0003-1647-6444.

Abstract. *The problem is formulated and the need for developing a methodology for modeling the behavior of antagonistic agents in security systems is shown. The presented concept is implemented at three levels, namely: the level of the security system as a whole, the level of individual agents and the level of the group of agents. Five stages of the concept implementation are presented. At the first stage, it is proposed to analyze protected business processes and threats to these processes. An ontological model is proposed as a basic model of this stage as a carrier of knowledge about the studied prelet region. An approach to the automation of ontology construction is presented, focused on the intellectual analysis of texts in natural languages, namely, texts of articles published in scientific journals. At the second and third stages of constructing the methodology, models of individual and group behavior of agents of cybersecurity systems are proposed. The presented models reflect the reflective properties of agents that affect the decision-making and learning processes. The developed models made it possible to form a model basis for the self-organization of the security system. A practical application of the described models is an algorithm for determining the implementation of the most probable threat, based on the cost indicators of threats and the probabilities of their implementation. This can ensure the efficient distribution of limited financial investment in cybersecurity.*

Keywords: *cybersecurity, antagonistic agents, modeling methodology, reflective agent, multi-agent systems, business process loop.*

Introduction

Processes for ensuring the security of business processes in the context of an increase in the number, variety and complexity of cyber attacks are mainly human and warring. Their features are determined by the interactions of the attacker, defender and user. Modeling the features and behavior of individuals included in the cybersecurity system is of particular importance for considering the characteristics of this subject area [1, 2].

The challenges of managing cybersecurity systems are initially multidisciplinary. Solutions at various levels of the control loop of such systems are closely interconnected. Thus, investment planning in the development of countermeasures (the level of strategic management) is closely related to the prediction of cyber threats and the operational planning of protective measures (the level of tactical and operational management) [3].

In the mathematical modeling of cybersecurity systems, it should be borne in mind that there are many models, each of which is able to answer a very specific range of specific questions about the behavior of both the attacker and the defender. Each of these models has its own goal and mathematical structure.

The use of any one modeling method and, accordingly, one class of models in solving complex, intercon-

nected management problems, as a rule, leads to inconsistent model fragments and far-from-reality problem statements that do not allow obtaining the required support for decision-making in managing cybersecurity systems.

The use of various concepts, tools and decision support models in solving real problems of ensuring the required level of protection of critical infrastructure facilities is due to the following features: firstly, the complexity of the tasks of managing a cyber defense system, and secondly, the simultaneous solution of control problems on various structures of a cybersecurity system (technological organizational, functional, informational, software, technical, financial), and thirdly, by changing management tasks, truktury and completeness of the source and output data in dynamics in the conditions of existence of hybrid threats

The conditions of uncertainty in which cybersecurity systems operate are characterized by a lack of information necessary to formalize the processes occurring in them. Uncertainty is caused, on the one hand, by the insufficiency or complete absence of methods and means for determining the state of the parties to the conflict, and, on the other hand, by ignorance of the laws governing the processes because of their complexity and insufficient. These factors make it impossible to analytically describe and build formal models that take into account the specifics of cybersecurity systems, which, in turn, significantly reduces the effectiveness of managing such systems under hybrid threats.

In the case when traditional management methods and mathematical descriptions do not give the desired results, the role of the decision maker (DM) sharply increases. DM, based on the ideas and knowledge of experts in this field and their own experience and intuition, are obliged to find solutions to the problem with a certain level of efficiency.

A significant contribution to the decision made by the decision maker is made by the subjective factor, which in cybersecurity systems affects not only the adoption, but also the result of the impact of managerial decisions. This is due to the fact that a significant part of these effects is directed at a person who is an integral part of these systems. In this regard, when formalizing the processes of confrontation under conditions of cyber conflict, it becomes necessary to take into account the features caused by human behavior. Therefore, when constructing a formal model, it is advisable to use methods based on modeling the intellectual activity of decision makers. This allows you to reduce the degree of subjectivity of decisions and, as a result, increase the efficiency of managing the security system.

All this leads to the need to develop a methodology for modeling not only the processes of ensuring cybersecurity of critical infrastructure objects, but also, first of all, the behavior in the process of interaction of participants in cyber conflict.

Research results

The analysis showed that today there is still no scientifically based method for assessing the most likely threats to information security, based on economic estimates of the cost of an attack and the damage done. Such estimates can be obtained based on an analysis of the behavior of criminals and advocates of information resources at any level.

The behavior of cybercriminals and the defenders opposing them is determined by many cyberattacks, the description and classification of which are given in different classifiers of threats [4-6]. At the same time, in the classifiers of information security threats there are not only the probabilities of the realization of a particular threat, but also the cost estimates of both the implementation and the losses that may be incurred when the threat is realized.

A radical review of the current methodological foundations for modeling the behavior of security system agents is required.

On the one hand, the theory lacks a holistic, scientifically based methodology for modeling the behavior of interacting agents in security systems, due to the complexity of the modeling object and the lack of appropriate methods and tools for modeling such complex processes as behavior in conflict conditions.

On the other hand, practice requires the theory to search for new approaches to providing protection against threats in all aspects of security: information security, cybersecurity, information security in a hybrid and synergistic environment.

The absence of a scientifically based methodology for modeling the processes of agent interaction in security systems, the use of which provides an economic justification for the level of security of the business process circuit in the context of modern hybrid threats, inhibits the process of efficient distribution of funds to counter attacks, due to the lack of methods for predicting the most likely attacks and assessing their value.

Modeling the behavior of objects of counteraction of security systems is not traditional for security systems. The complexity of the simulation object, its stochasticity and the lack of appropriate methods and tools for modeling such complex processes explain the lack of a holistic, scientifically based methodology for modeling the behavior of interacting agents in security systems [7-8].

To build the methodology, the concept of modeling the behavior of security system agents is proposed, which is implemented at three levels, the basic level of the security system, the level of individual agents, and the level of the group of agents (Fig. 1). The concept is aimed at guaranteed security of the organization's business processes, it allows you to create a circuit of business processes of the security system.

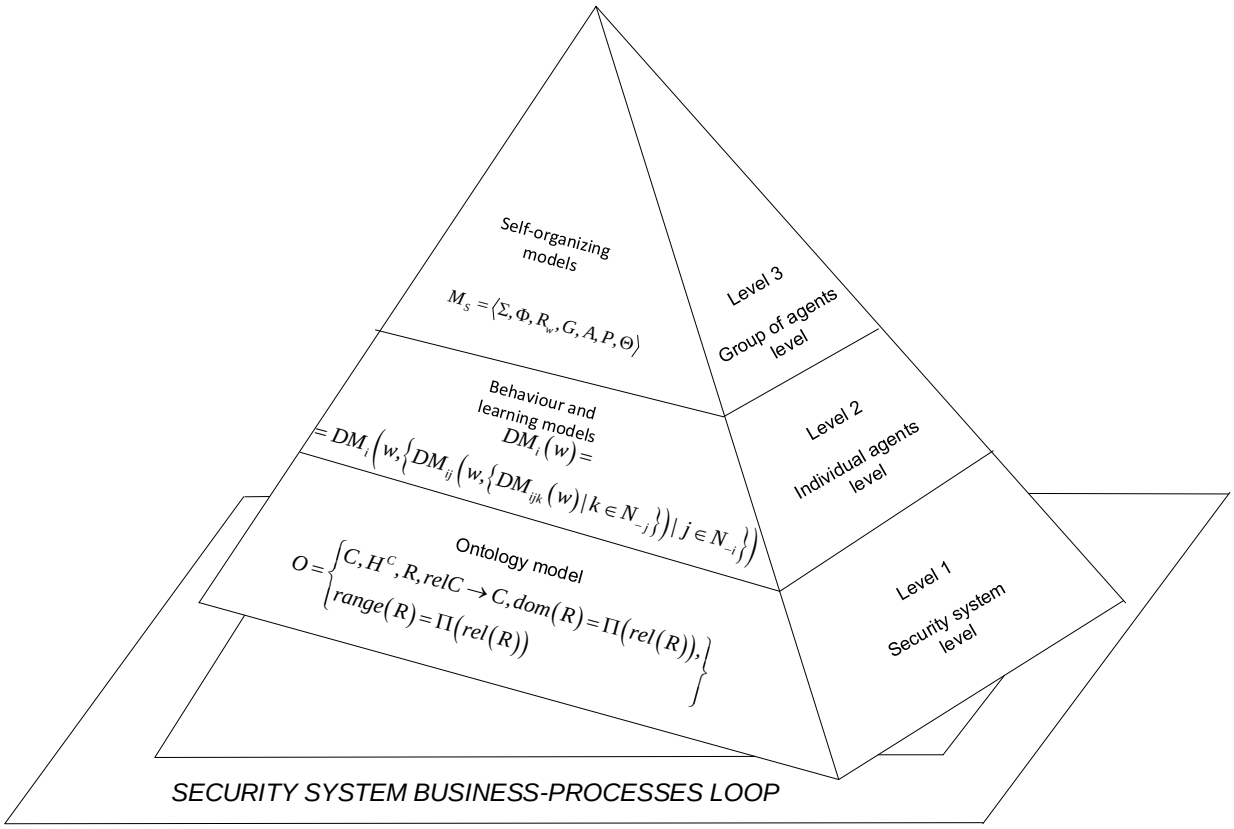


Fig. 1. Implementation of models basis of methodology

Stage 1. Analysis of the business processes loop and possible threats

The outline of the organization's business processes should be considered as the main object of cyber attacks. An organization's business process loop (BP) is a set of business processes and their implementation of information resources, the implementation of which in a given sequence leads to the achievement of the organization's goals, which can be described as follows:

$$S^{BP} = \left\{\left\langle S^{BP_1}, IR^{BP_1}, T^{BP_1}\right\rangle, ..., \left\langle S^{BP_n}, IR^{BP_n}, T^{BP_n}\right\rangle\right\},$$

where S^{BP} - is the loop of business processes as a set of BPs, each of which represents: S^{Bpi} - is the i-*th* business process. defined by the structure of relationships of individual business operations performed in a certain sequence; IR^{BPi} - a set of information resources of the i-*th* business process; T^{BPi} - a set of threats to the i-*th* business process.

Ensuring the protection of the organization's business processes can be represented similar to the BP contour, but the security system. The security system business process circuit is a set of business processes and the resources necessary for them, the implementation of which ensures the normal functioning of the organization's business process circuit. This BP loop can be represented similarly, namely:

$$S^{BS} = \left\{\left\langle S^{BS_1}, Rs^{BS_1}, T^{BS_1}\right\rangle, ..., \left\langle S^{BS_m}, Rs^{BS_m}, T^{BS_m}\right\rangle\right\}$$

where S^{BP} is the circuit of business processes of the security system as a set of BPs, each of which represents S^{BSi} - i-th business process defined by the structure of the links of individual business operations that are performed in a specific sequence in the security system; IR^{BSi} - a set of information resources protected by the i-th business process of the security system; T^{BSi} - a set of threats, the i-th business process of the security system provides protection against.

First of all, the ontological model is built at the first level as a carrier of knowledge about conflict-cooperative interactions of security system agents. A formalized ontology model is proposed in the form of:

$$O = \left\{\begin{matrix} C, H^C, R, relC \to C, dom(R) = \Pi(rel(R)), \\ range(R) = \Pi(rel(R)) \end{matrix}\right\}.$$

The basic unit for ontologies is the concept. As a rule, concepts are hierarchically organized in a hierarchy of concepts. We define the set of concepts and the hierarchy of concepts as follows [9-14]:

– The set *C* whose elements are called concepts.

– H^C hierarchy of concepts: H^C is a relation $H^C \subseteq C \times C$ called a concept hierarchy or taxonomy. $H^C(C_1, C_2)$ means that C_1 is a subconcept of C_2.

Concepts and the hierarchy of concepts are further expanded with the help of non-taxonomic relationships between concepts and a set of axioms. We define them as follows:

– The set R, whose elements are called relations, the sets C and R do not intersect.

– A function $rel: R \to C \times C$ that correlates concepts is not taxonomic. The function $dom: R \to C$ with $dom(R) := \prod_1(rel(R))$ sets the subject area *R*, and the range $R \to C$ with $range(R): \prod_2(rel(R))$ gives its range. For $rel(R) = (C_1, C_2)$ we also record $R(C_1, C_2)$.

– A set of axioms of the ontology of A^O, expressed in the corresponding logical language, for example, in the language of first-order logic.

For the operation of the Text-To-Onto components, it is necessary to provide a link in the contents of the document (in particular, individual words) to ontological objects. This mapping is provided by the lexicon (or case). The lexicon for the structure of ontologies $O := \{C, R, H^C, rel, A^O\}$ is a 4-dimensional $L := \{L^C, L^R, F, G\}$, consisting of:

– two sets of L^C and L^R whose elements are called lexical entries for concepts and relations, respectively;

– two relationships $F \subseteq L^C \times C$ and $G \subseteq L^R \times R$, called references for concepts and relationships, respectively. Starting from *F*, we set for $L \in L^C$, $F(L) = \{C \in C \mid (L, C) \in F\}$, and also for $F^{-1}(C) = \{L \in L^C \mid (L, C) \in F\}$, (*G* and *G*-1 are defined similarly).

Formal semantics for ontologies is an indispensable condition, which is implemented in the inference mechanism to ensure this for the above definition. We also additionally formulate axioms of A^O,, specific for the subject area of interaction of antagonistic agents, and a knowledge base consisting of concepts and relations between them.

To build an ontology model, the TextToOnto ontology construction approach can be proposed, which allows you to build an ontology based on texts from various scientific sources: scientific articles, monographs, etc., obtained from various databases of scientific publications, repositories, university sites, and other sources (fig. 2).

As a result of the first stage of building the methodology:

– the components of the business process contour are determined;

– the probabilities of cyber attacks on information resources are estimated;

– the correspondence between the attack, the information resource and the business process that uses it is determined;

– the cost of information resources is determined.

Stage 2. Development of level models of individual security system agents

Creating models of this level, it is assumed that each agent *i* perceives the state of the confrontation medium *w* and performs the action a_i at each step. It is contemplated that the behavior of each agent can be described using a simple mapping of state to action. It is also assumed that the correct behavior exists for each agent. The agent's target behavior consists of all the correct mappings of the state and action of the agent. To determine the target behavior for an agent, as a rule, you need to know for each set of actions that all other agents will perform in this *w*.

For further discussion, we introduce the following notation for representing models of individual and group behavior of agents:

– *N* - the set of all agents, among which there is one particular agent; *W* the set of possible states of the agent's counter environment, where $w \in W$ is one specific state.

– A_i - the set of all actions that the ith agent can take.

– $A_i = DM_i(W)$ - decision function for agent *i*. It says what action Agent *i* will take in every environment state.

– $A_i = G_i(W)$ - objective function for agent *i*. It tells us what action agent *i* should take. It takes into account the actions that other agents will take.

– $e(A_i) = \Pr[DM_i(w) \neq G_i(w) \mid w \in D]$ - Agent *i* error. This is the probability that agent *i* will take the wrong action, given that the worlds w are taken from a fixed probability distribution *D*.

More formally, the behavior of each agent is represented by the decision function defined $A_i = DM_i(W)$ for agent *i*. This function maps each state $w \in W$ to the action $a_i \in A_i$ that agent i takes in that state.

The action that agent i must perform in each state *w* (that is, the correct action for each state *w*) is defined by the objective function $A_i = G_i(W)$, which also maps each state $w \in W$ to the action $a_i \in A_i$. Since the choice of action for agent *i* often depends on the actions of other agents, the objective function i must take these actions into account. That is, in order to generate the objective function for *i*, you need to know $DM_j(w)$ for all $j \in N_{-i}$ and $w \in W$ (the record $j \in N_{-i}$ means that *j* belongs to the set of all agents except the *i*-th one). These functions $DM_j(w)$ tell us about the actions that all other agents will perform in each state *w*. You can use these actions in conjunction with state *w* to determine the best action that *i* should take. An agent usually does not have direct access to its target function, and the target function is not part of the agent.

The measure of the correct behavior of the agent is given by the measure of error. Define the error of the decision-making function DM_i of agent *i* as:

$$e(DM_i) = \sum_{w \in W} D(w) \Pr[DM_i(w) \neq G_i(w)] = \\ = \Pr_{w \in W}[DM_i(w) \neq G_i(w)],$$

where *D*(*w*) is the fixed probability distribution of threats in accordance with the classifier containing estimates of the implementation of a particular threat at any time. $e(DM_i)$ gives us the likelihood that agent *i* will take the wrong action. $e(DM_i)$ is the measure we use to evaluate how well agent *i* works. Error 0 means that the agent performs all the actions dictated by its target function. Error 1 means that the agent never takes actions dictated by its target function. All of these designations form our basis for describing MAS.

An agent can be implemented as an instance of a simple $DM_i(w)$ decision function, or an agent can model other agents as using a decision function and use predictions from these functions to determine what action to take, or an agent can use an arbitrarily deep nesting of functions. We call these *k*-level agents, where $k \geq 0$ refers to the level of nesting that the agent uses.

Fig. 2. Ontological model of cooperative-conflict relations between cyber agents

We define a zero-level agent as an agent that is not able to recognize the fact that there are other agents in the world. The only time the presence of other agents affects the agent of the 0^{th} level is when their actions lead to changes in the payment that the agent of the 0th level receives. The zero-level agent *i* is implemented using a procedure that directly creates an instance of a decision function $DM_i(w)$. This function captures all the knowledge that an agent possesses.

Level 1 agent *i* recognizes the fact that there are other agents in the world and that they are taking action, but he does not know anything about them. Given these facts, the strategy of the 1st level agent is to predict the actions of other agents based on their models and use these forecasts when trying to determine their best action. The 1st-level agent assumes that other agents choose their actions using the mapping *W* to *A*. Therefore, the 1st-level agent *i* is implemented using procedures that directly create functions $DM_i(w,\vec{a}_{-i})$ and $DM_{ij}(w)$ for all agents $j \neq i$. Agent behavior can be described by the function

$$DM_i(w) = DM_i(w,\vec{a}_{-i}),$$

where

$$\vec{a}_{-i} = \{DM_{ij}(w) \mid j \in N_{-i}\}.$$

Table 1

The decision functions that various level *k* agents have

Level	Types of knowledge	Behavior
0-level	$DM_i(w)$	
1-level	$DM_{ij}(w)$ $DM_i(w,\vec{a}_{-i})$	$A_i(w) = DM_i(w) = DM_i(w,\{DM_{ij}(w) \mid j \in N_{-i}\})$
2-level	$DM_i(w,\vec{a}_{-i})$ $DM_{ij}(w,\vec{a}_{-j})$ $DM_{ijk}(w)$	$A_i(w) = DM_i(w) =$ $= DM_i(w,\{DM_{ij}(w,\{DM_{ijk}(w)\} \mid k \in N_{-j}) \mid j \in N_{-i}\})$

In other words, the behavior of the 1st level agent can be described using the decision-making function, which is formed by the following composition of the agent's crucial functions:

$$DM_i(w) = DM_i(w,\{DM_{ij}(w) \mid j \in N_{-i}\}).$$

An example of agent *i* level 1, modeling two other agents *j* and *k*, is presented in Fig. 4. Here we see the functions of agent *i*, which include his agent models *j* (DM_{ij}) and *k* (DM_{ik}). For example, DM_{ij} is a function that tells us what *i* thinks *j* will do in each state *w*. This action does not have to be performed by *j*, since model *j* may be incorrect. That is, it is not necessary that $DM_{ij}(w)=DM_j(w)$ for all $w \in W$. When *i* needs to determine what action to take, he first evaluates his models *j* and *k* to determine what actions he will perform, i.e., a_j and a_k in fig. 3. These actions then form a vector. Since we now have values for *w* and $\vec{a}_{-i}$ for, we can evaluate the function $DM_i(w,\vec{a}_{-i})$ for these values to get the action that *i* will take, that is, a_i in fig. 3.

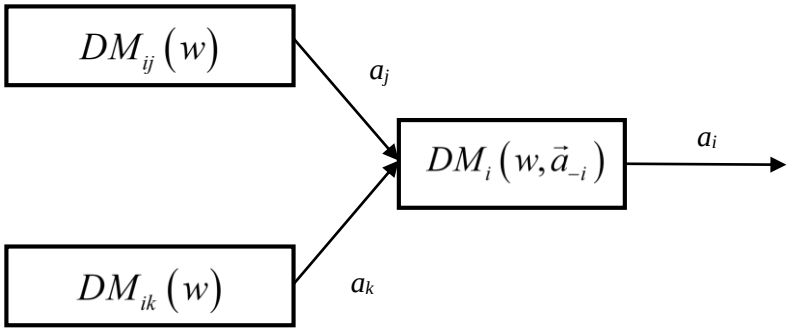


Fig. 3. The agent *i* of the 1st level determines what action to take

Agent *i* level 2 also recognizes other agents in the world and, in addition, has some information about their decision-making processes and previous observations. That is, a level 2 agent has an understanding of the internal procedures of other agents used to select an action. This model of other agents allows a level 2 agent to reject "useless" information when choosing the next action, for example, *i* may know that *j* performs the same action regardless of what, in his opinion, others will do. We can say that a level 2 agent is implemented using procedures that directly implement the three decision-making functions $DM_i(w,\vec{a}_{-i})$, $DM_{ij}(w,\vec{a}_{-j})$ and $DM_{ijk}(w)$. $DM_{ij}(w,\vec{a}_{-j})$ fixes that agent i thinks it will do *j*, given that both of them are in state *w*, and *j* believes that all other agents will perform the actions specified $\vec{a}_{-j}$. $DM_{ijk}(w)$ reflects the fact that *i* thinks *j* thinks that *k* will do in state *w*, where $i \neq j \neq k$. Note that $DM_{ijk}(w)$ it is possible that *i=k*, that is, agent *i* may have a model of itself that was built by another agent. Level 2 Agent Behavior Can Be Described Using Decision Function

$$DM_i(w) =$$

$$= DM_i\left(w, \left\{DM_{ij}\left(w, \left\{DM_{ijk}(w) \mid k \in N_{-j}\right\}\right) \mid j \in N_{-i}\right\}\right).$$

The simple way that a Tier 1 agent can become Tier 2 agents is to assume that "others are like him" and model others using the same decision and observation functions that the agent himself used when he was a Tier agent 1. This type of process, of course, will be effective only when other agents really look like a modeling agent. If so, then this method can be used to "load" the agent to any level of modeling.

So far, we have assumed that agents possessed all the knowledge they needed to choose their actions (i.e. all $DM_i(w)$), and that this knowledge did not change over time. However, it should be considered more realistic that agents use some form of machine learning, which sometimes begins with absolutely no initial knowledge, and sometimes is based on existing knowledge that developers have built into the agent.

We can model these agents, allowing the change in the decision-making functions of $DM_i(w)$ over time $DM_i^t(w)$. The superscript *t* indicates the time at which this decision function is executed by agent *i*. The learning task faced by the agent is to change $DM_i^t(w)$ it so that it matches $G_i^t(w)$. If we represent the space of all possible decision-making functions, then for agent i $DM_i^t(w)$ and $G_i^t(w)$ will be two points in this space, as shown in fig. 4. The problem of training an agent can be reformulated as the problem of moving its crucial function as close as possible to its objective function, where the distance between the two functions is determined by an error $e(DM_i^t)$. This is a traditional machine learning problem, which is shown in fig. 4.

However, as soon as agents begin to change their decision-making functions (that is, change their behavior), the learning problem becomes more complex, since these changes can lead to a change in the target functions of other agents. As a result, we get the function of a moving target, as shown in fig. 5. In these systems, it is not clear whether the error will ever reach 0 or, more generally, what the expected error will be over time. Determining what will happen to an agent error in such a system is what is called the moving target function problem.

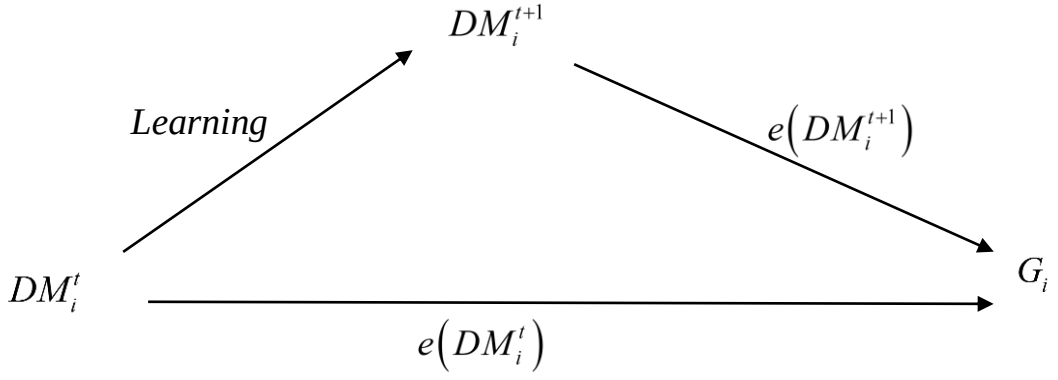


Fig. 4. The traditional issue of learning

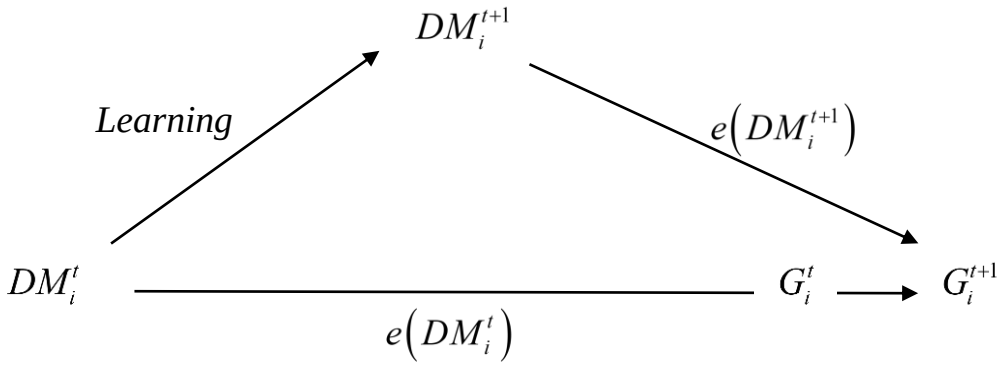


Fig. 5. The problem of learning in multi-agent systems

We assume that the agents in MAS are involved in the discrete action / training cycle shown in fig. 6. The cycle works as follows. At time *t*, agents perceive the world $w^t \in W$, which is taken from a fixed distribution of $D(w)$. Then the agents take actions dictated by their functions DM_i^t, it is assumed that all these actions are performed in parallel. Finally, each of them receives a reward, which their respective learning algorithms use for change DM_i^t in order to better fit G_i^t. By the time *t*+1, the agents receive new functions DM_i^{t+1} and are ready to perceive the world again and repeat the cycle. Note that at time *t*, the target function G_i^t of agent *i* is obtained taking into account DM_j^t all other agents. That is, G_i^t this is the best possible behavior for agent *i* at time *t*, given that all other agents $j \in N_{-i}$ take actions dictated by their actions DM_j^t.

The agent training algorithm is responsible for changing DM_i^t in DM_i^{t+1} so that it better matches G_i^t. Various machine learning algorithms have achieved this correspondence with varying degrees of success.

After agent *i* performs an action and receives some return, he activates his learning algorithm. The learning algorithm is responsible for using this gain to change DM_i^t to DM_i^{t+1}, making DM_i^{t+1} as appropriate as possible. We can expect that for some *w* it would be true $DM_i^t(w) = G_i^t(w)$, while for some others *w* it would not. That is, some of the $w \to a_i$ mappings specified $DM_i^t(w)$ could be incorrect. In general, a learning algorithm can affect both correct and incorrect display. We consider these two cases separately.

Les's start with looking at the wrong mappings and define the rate of change of the agent as the probability that the agent will change one of its incorrect mappings. Formally, we define the rate of change c_i for agent *i* as

$$\forall w\, c_i = \Pr\left[DM_i^{t+1}(w) \neq DM_i^t(w) \mid DM_i^t(w) \neq G_i^t(w)\right].$$

The change rate tells how likely it is that the agent can change the incorrect mapping to something else. This "something else" may be the right action, but it may be another wrong action. The probability that the agent changes the incorrect display to the correct action is called the agent's learning rate, which is defined as l_i where

$$\forall w\, l_i = \Pr\left[DM_i^{t+1}(w) \neq G_i^t(w) \mid DM_i^t(w) \neq G_i^t(w)\right].$$

When determining the l_i value for a specific agent, it is necessary to remember that the operating environment, visible at each time step, is taken from $D(w)$.

There are two limitations that must always be fulfilled in satisfying these two indicators. Since the transition to the correct display implies that the change has been made, the value of l_i must be less than or equal to c_i, that is, it must always be true. In addition, if then c_i=l_i, since only two actions are available, then the erroneous one must be correct. The additional value for the learning speed is 1-l_i and refers to the probability that the incorrect display will not be changed to the correct one. An example of learning speed l_i=0.5 means that if agent i initially had all the mappings incorrect, then after the first iteration it will receive only half of them incorrect.

Now consider the correct agent mappings and define the retention coefficient as the probability that the correct mapping will remain true at the next iteration. The retention coefficient is defined as r_j where

$$\forall w\, r_i = \Pr\left[DM_i^{t+1}(w) = G_i^t(w) \mid DM_i^t(w) \neq G_i^t(w)\right].$$

We suggest that the behavior of a wide range of learning algorithms can be fixed (or at least approximated) using the corresponding values for c_i, l_i and r_i. However, note that these three metrics claim that the changing mappings are independent of the newly acquired w. This independence can be justified by noting that most learning algorithms usually perform some form of generalization. That is, after observing one state of the environment *w* and the related recoil, a typical learning algorithm is able to generalize what he has studied to some other states of the world. This generalization is reflected in the fact that indicators of change, learning, and retention apply to all people. However, a more accurate model takes into account the fact that in some training algorithms, the display of the just seen state of the world will change with a higher probability than the display of any other state of the world.

Speeds are time independent, because we assume that agents use the same learning algorithm throughout their lives. Speeds cover the capabilities of this learning algorithm and, therefore, should not change over time.

Finally, we determine volatility to indicate the probability that the objective function will change from time *t* to time *t*+1. Formally, volatility is defined as

$$\forall w\, v_i = \Pr\left[G_i^{t+1}(w) \neq G_i^t(w)\right].$$

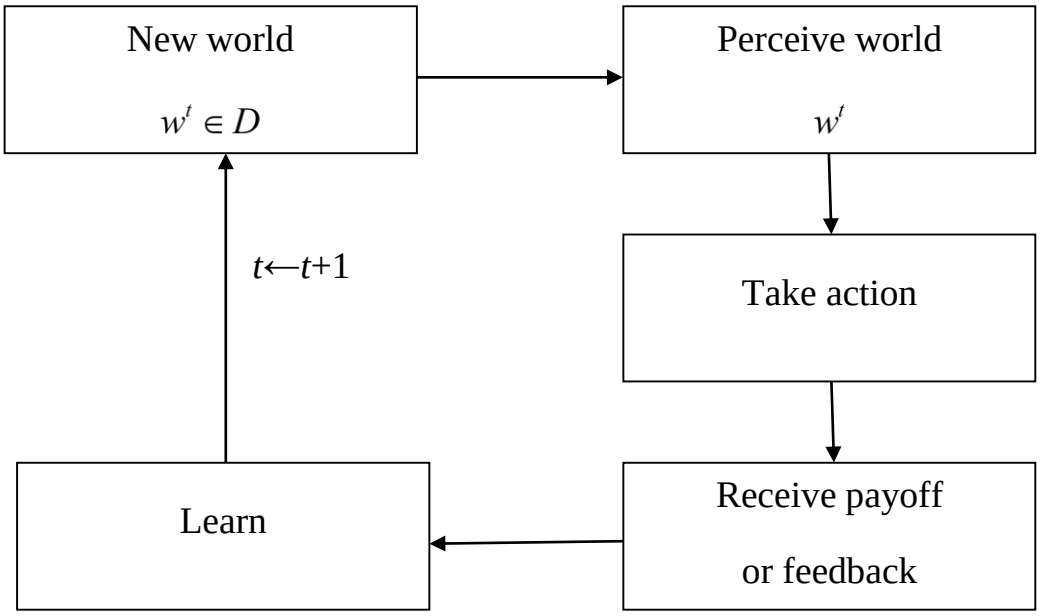


Fig. 6. The cycle of action / training for the agent

Stage 3. Development of level models of a group of security system agents.

For a model representation of the level of a group of security system agents, it is necessary to detail the model of an individual agent, taking into account the fact that the security system is distributed in its structure.

The basic model of this level is a modification of the previously developed model of an individual agent, but is modified to take into account the group features of the models of this level. to take into account the dynamics of processes and interactions of individual agents. This modification allows you to implement the model of the 2nd level of methodology (coordination, adaptation and self-organization). The corresponding sets and functions of an individual agent are shown in Fig. 7.

The key concept of the proposed approach is an individual element of the system, a module whose agent is an expert on a unique subsystem, regarding which he and only he has the most complete knowledge, and for which he is responsible.

Stage 4. Development of system level models.

Cybersecurity systems (CS) belong to the class of complex organizational systems, the main features of which are the number of their constituent elements, the variety of links between the elements, and the limiting uncertainty (a priori or arising during operation).

The approach to solving the problem of managing such complex systems may consist in creating systems capable of self-organization, and possibly making logical decisions, i.e., making decisions in alternatives that are equally likely. When constructing the structure of such systems, the principles of self-organization are used.

A cyber security system as a self-organizing system is characterized by a set of functions performed by it. The functional description consists in defining the functions of the system, given in accordance with the spatial or structural feature. System functions are defined through interconnections with other systems.

The mathematical model of a self-organizing system is constructed in accordance with its definition and properties. The basis for building the model is a structural-functional approach. From the point of view of this approach, the self-organizing SS system can be viewed as a set:

$$SS = \langle \Sigma, \Phi, R_w, G, A, P, \Theta \rangle,$$

where Σ is the structure of the system; Φ − system function; R_w is the emergence ratio; G − many goals; A − the relation of adaptability; P − a set of memory elements; Θ – set of time moments.

System structure. One of the most important characteristics of a self-organizing system is its structure. The structure Σ is considered as a multigraph with certain nodes:

$$\Sigma = \langle S, C, R, R_i \rangle,$$

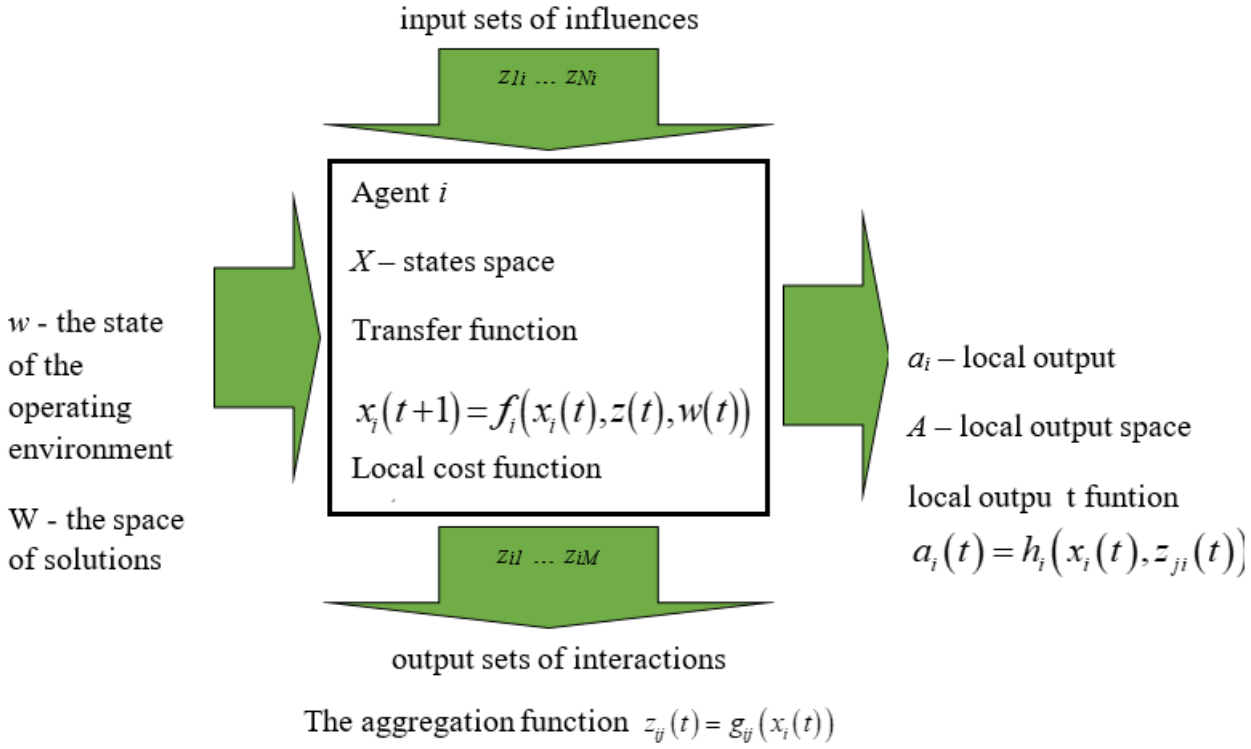
where S is the set of elements of the system (the nodes of the multigraph); C − the set of parameters of elements (the set of statements regarding the properties of the nodes of a multigraph); R − the set of connections between the elements (arcs of a multigraph); R_t − the incidence relation that assigns a pair of nodes to each arc. The structure of self-organizing systems is characterized by integrity, adaptability and development.

The integrity of the structure of the system is determined by the set of elements and the branched connections between them.

The adaptability of the structure of the system, its response to changes in the environment emphasize its dynamic properties - its variability in functioning.

The extreme form of system variability is the development of its structure, which is understood as the complication of the system, its accumulation of information, the transition to a more ordered state.

System function It is an external manifestation of its properties when interacting with the environment. The function of the system is a way to act when the goal is achieved and for a self-organizing system is determined by its goal.



$v_i(x_i,\vec{z}_i,t)$ - local costs associated with the transition (at time T) from state x_i to time t during interaction;

$w_i(z_i,x_i,x_i^+,t)$ - minimum local transition costs from the state x_i in t steps with given input sequences of interactions $z_{1i}, \ldots, z_{Ni}$

Fig. 7. The model of agent with interaction

Set the system function as

$$\Phi = \langle \Pi, F \rangle,$$

where Π – the set of variables that determine the function of the system; $X \subseteq \Pi = W \times M$; X – is the set of input actions that interact with the information inputs W; F – the set of functions on variables that, for each element, determine the dependence of the output variables on the input variables.

A state is a set of essential properties that the system possesses at a given time. The state of the system is determined by the set of states of its elements.

The functions implemented by the self-organizing system can be conditionally divided into three groups: target, basic (basic), auxiliary (additional).

The objective function corresponds to the main functional purpose of the system and is set solely by the purpose of its operation.

The main functions reflect the orientation of the system and are a combination of the macrofunctions implemented by it, necessary for the most optimal achievement of the goal, and are determined by the objective function and the system quality criteria.

By additional functions, we will mean functions aimed at maintaining or improving the quality of the system's functioning within certain limits.

Emergence. The emergence relationship R_w reflects the unity of the structure and function of the system, the relationship between them. It is a parameter forming a system: from the two previous objects – structures and functions – it forms a system. Neither structure nor function form a system. The emergence in some way connects the elements of the structure with functions, maps Φ into Σ: $R_w = |\Phi| \times |\Sigma|$, where | ... | means a set of elements of Φ or Σ; $\times$ - the sign of the Cartesian product of two sets.

Set of goals. The set G corresponds to the set of goals facing the system; G is a multi-grid, i.e., a distributive grid endowed with join, intersection, and composition operations. If $\alpha_1 \le \alpha_2$, then $\alpha_1 \bigcirc \alpha \le \alpha_2 \bigcirc \alpha$ and $\alpha \bigcirc \alpha_1 \le \alpha \bigcirc \alpha_2 \ \forall \alpha, \alpha_1, \alpha_2 \in G$. The order relation on G is interpreted as follows: if $\alpha_1, \ldots, \alpha_n \in G$ and $\alpha_1 \le \alpha_2$, then the solution of the α_2 problem provides for the solution of the α_1 problem; $\alpha_1 \cup \alpha_2$ interpreted as a task consisting in solving problems α_1 and α_2; $\alpha_1 \cap \alpha_2$ - as the largest of those problems whose solution is obtained simultaneously with the solution of problems α_1 and α_2. The operation of composition on G is interpreted as a strictly sequential solution of problems, and the closure of G with respect to composition means that an arbitrary development of a set of targets in time is provided in the

system. The sets Σ and *G* are connected as follows. Let *H*(Σ) be the set of mappings that preserve the order of a partially ordered set Σ into itself. Then H (Σ) is a multigrid with operations $\cup$, $\cap$, and $\bigcirc$, and there is a homomorphism $\gamma : G \to H(\Sigma)$.

The assignment of a mapping $\gamma : G \to H(\Sigma)$ reflecting the restructuring of the system under the influence of the goal α can be taken as a comparison of the goal $\alpha \in G$ and the structure $\sigma \in \Sigma$ of some new structure $\sigma' = \gamma(\alpha)\sigma$.

Adaptability. The relationship of adaptability links the behavior and structure of the system with changes in the effects of the external environment and with internal states. Adaptability in some way reflects the effects of the external environment *X* and the state *M* on the structure Σ and the function Φ. This ratio falls into two: A_1 and A_2. $A_1 < A_2$ and. Here

$$A_1 = |\Xi| \times |\Sigma|; \quad A_2 = |\Xi| \times |\Phi|,$$

where $\Xi = W \times M$, $X \subseteq \Xi$.

Time. Θ is a directed set of moments of time, that is, a set with an operation ≤. In a self-organizing system, a specific situation ς is put in correspondence with each moment of time, that is, there is a map $\beta : \Theta \to \Omega$, where Ω is a set of situations.

Stage 5. Development of an algorithm for determining the set of probable threats and assessing their cost indicators.

The proposed algorithm implements the following actions. Both sides of the attack are determined by the importance (rating) of the attacks that are economically feasible (fig. 8).

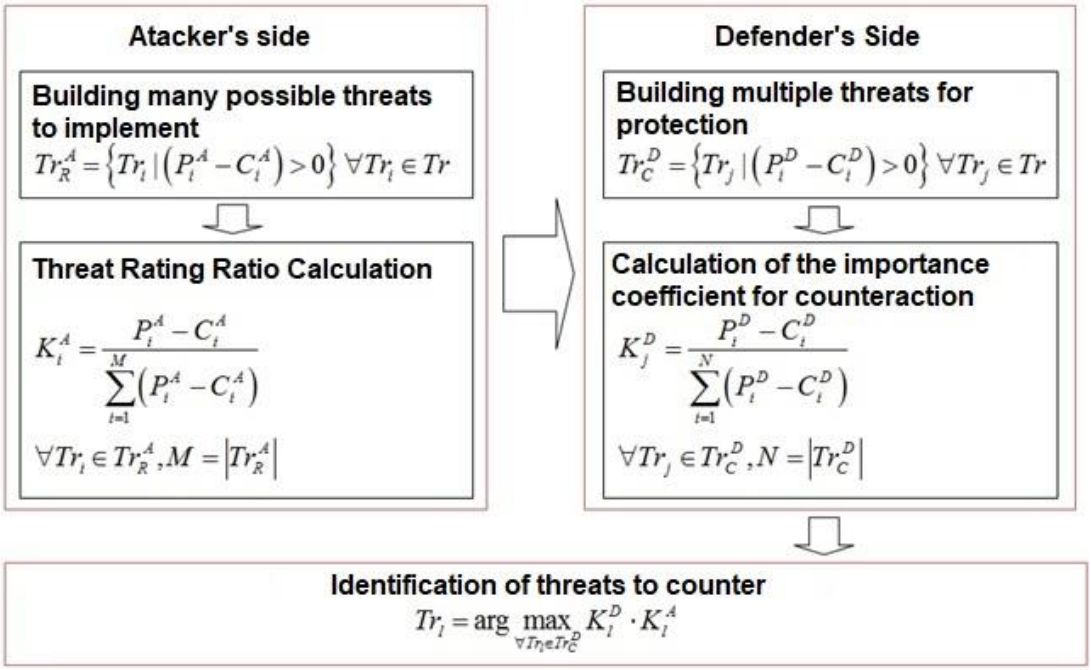


Fig. 8. Determining the most probable threat to implementation

1st step. Those attacks are determined whose effect of the implementation exceeds the costs of their implementation.

$$Tr_R^A = \{Tr_i \mid (P_i^A - C_i^A) > 0\} \; \forall Tr_i \in Tr,$$

where Tr_R^A – many potential threats that are effective for the attacker; Tr_i – threat to the *i*-th information resource; P_i^A – assessment of the cost of success of the attack on the *i*-th resource of the business process by the attacker; C_i^A – the cost of an attack on the *i*-th resource of a business process by an attacker.

2*nd* step. Similarly, the directions of protection are determined that provide an effect higher than the costs of their provision.

$$Tr_C^D = \{Tr_j \mid (P_i^D - C_i^D) > 0\} \; \forall Tr_j \in Tr,$$

where Tr_C^D – many threats against which it is economically feasible to build protection; P_i^D – estimation of the cost of the loss of the *i*-th information resource for the defense side; C_i^D – the cost of protecting the *i*-th information resource for the protection side.

3*rd* step. The importance factors for attackers are defined as the share of the winnings of the total winnings, which can be obtained potentially when implementing the whole complex of threats for attackers:

$$K_i^A = \frac{P_i^A - C_i^A}{\sum_{i=1}^{M}(P_i^A - C_i^A)};$$

$$\forall Tr_i \in Tr_R^A, M = |Tr_R^A|.$$

where K_i^A – rating coefficient (importance) of threat realization to the i-th information resource; *M* – the power of a multitude of selected potentially effective threats to the attacker.

4th step. The importance factors for defenders are defined as the share of the winnings of the total winnings that can be obtained potentially when implementing the entire range of protective measures:

$$K_j^D = \frac{P_i^D - C_i^D}{\sum_{i=1}^{N}\left(P_i^D - C_i^D\right)};$$

$$\forall Tr_j \in Tr_C^D, N = \left|Tr_C^D\right|.$$

where K_j^D - rating coefficient (importance) of building the protection of the *j*-th information resource.

5th step. As the most probable threat that can be realized, one of them is selected for which the product of the importance coefficients of the attacker and the attacker is the maximum:

$$Tr_l = \arg\max_{\forall Tr_l \in Tr_C^D} K_l^D \cdot K_l^A.$$

Conclusion

The synthesis of models for constructing a methodology for modeling the behavior of the opposing agent of the security system is based on: improved models of the decision-making nmodel and the proposed models of training, coordination and self-organization, as well as models for assessing the distribution of funds for information security.

References

[1]. O. Milov, A. Voitko, I. Husarova, O. Domaskin, E. Ivanchenko, I. Ivanchenko, O. Korol, H. Kots, I. Opirskyy, O. Fraze-Frazenko, "Development of methodology for modeling the interaction of antagonistic agents in cybersecurity systems", *Eastern-European Journal of Enterprise Technologies*, vol. 9, no. 2, pp. 56-68, 2019.

[2]. O. Milov, S. Yevseiev, V. Aleksiyev, S. Balakireva, I. Tyshyk, O. Shmatko, "Development of a methodology for building an information security system in the corporate research and education system in the context of university autonomy", *Eastern-Europe Journal of Enterprise Technologies*, 3(9-99), pp. 49-63, 2019.

[3]. O. Milov, S. Milevsky, O. Korol, "Development of basic principles for corporate planning", *Системи обробки інформації*, випуск 1 (156), 2019.

[4]. С. Євсеєв, "Класифікатор кіберзагроз інформаційних ресурсів автоматизованих банківських систем", *Кібербезпека: освіта, наука, техніка*, № 2(2), 2018, С. 47-67.

[5]. О. Юдін, С. Бучик, А. Чунарьова, О. Варченко, "Методологія побудови класифікатора загроз державним інформаційним ресурсам", *Наукоємні технології*, № 2 (22), С. 200-210,2014.

[6]. О. Юдін, С. Бучик, "Класифікація загроз державним інформаційним ресурсам нормативно-правового спрямування. Методологія побудови класифікатора", *Захист інформації*, Том 17 (2), С. 108-116, 2015.

[7]. В. Городецкий, И. Котенко, О. Карсаев, "Многоагентная система защиты информации в компьютерных сетях: механизмы обучения и формирования решений для обнаружения вторжений", *Проблемы информатизации*, № 2, С. 67-73, 2000.

[8]. И. Котенко, О. Корсаев, "Использование многоагентных технологий для комплексной защиты информационных ресурсов в компьютерных сетях", *Известия ТРТУ*, №4, С. 38-50, 2002.

[9]. А. Милов, О. Король, "Разработка онтологии поведения взимодействующих агентов в системах безопасности", *4th International Congress on 3D Printing (Additive Manufacturing) Technologies and Digital Industry 2019 (11-14 April, 2019)*, pp. 832-842.

[10]. A. Maedche, S. Staab, "Discovering conceptual relations from text", *Proceedings of the 14th European Conference on Artificial Intellignece (ECAI'2000),* 2000.

[11]. A. Maedche, S. Staab, "Discovering conceptual relations from text", *In Proceedings of ECAI-2000*, IOS Press, Amsterdam, 2000.

[12]. A. Maedche, S. Staab, "Mining Ontologies from Text", *In Proceedings of EKAW-2000, Springer Lecture Notes in Artificial Intelligence (LNAI-1937), Juan-Les-Pins*, France, 2000. Springer, 2000.

[13]. A. Maedche, S. Staab, "Semi-automatic engineering of ontologies from text", *In Proceedings of the 12th Internal Conference on Software and Knowledge Engineering*. Chicago, USA, July, 5-7, 2000. KSI, 2000.

[14]. A. Maedche, S. Staab, "Semi-automatic engineering of ontologies from text", *In Proceedings of the 12th Internal Conference on Software and Knowledge Engineering*. Chicago, USA, July, 5-7, 2000. KSI, 2000.

УДК 004.946.5.056

Мілов О.В., Євсеєв С.П. Методологія моделювання процесів поведінки антагоністичних агентів в системах безпеки

Анотація. *Пропонується методологія моделювання взаємодії антагоністичних агентів в системах кібербезпеки з використанням методів на основі моделей рефлексивної поведінки антогоністіческіе агентів в умовах сучасних гібридних загроз. Визначені основні концепції, що формують основу інтегрованого моделювання поведінки антагоністичних агентів в системах кібербезпеки. Показано, що у більшості робіт акцент робиться на моделюванні поведінки тільки однієї зі сторін кіберконфлікта. У тому випадку, коли розглядається взаємодія всіх сторін конфлікту, підходи, що використовуються, орієнтовані на рішення часткових завдань, або моделюють спрощену ситуацію. Сформульована проблема і показана необхідність розробки методології моделювання поведінки антагоністичних агентів а системах безпеки. Запропонована концепція, що яка реалізується на трьох рівнях, а саме: рівні системи безпеки в цілому, рівні індивідуальних агентів і рівні групи агентів. Представлені п'ять етапів реалізації концепції. На першому етапі пропонується проводити аналіз бізнес-процесів та загроз цим процесам. В якості базової моделі цього етапу пропонується онтологічна модель як носій знань про досліджувану прлеметной області. Запропонован підхід до автоматизації побудови онтології, орієнтований на інтелектуальний аналіз текстів на природних мовах, а саме, текстів статей, опублікованих в наукових журналах. На другий і третій стадії побудови методології пропонуються моделі індивідуальної та групової поведінки агентів систем кібербезпеки. Представлені моделі відображаються рефлексивні властивості агентів, що впливають на процеси прийняття рішень і навчання. Розроблені моделі дозволяють*

сформувати модельний базис самоорганізації системи безпеки. У запропонованій методології традиційні методи і інструменти моделювання не протиставляються один одному, а розглядаються в сукупності, формуючи тим самим єдину методологічну базу моделювання поведінки антагоністичних агентів. Практичним використанням описаних моделей є алгоритм визначення реалізації найбільш ймовірної загрози, виходячи з вартісних показників загроз і можливостей їх здійснення, що може забезпечити ефективний розподіл обмежених фінансових коштів інвестування в систему кібербезпеки.
Ключові слова: *кібербезпека, антагоністичні агенти, методологія моделювання, рефлексивний агент, мультиагентні системи, контур бізнес-процесів.*

Милов А.В., Евсеев С.П. Методология моделирования процессов поведения антагонистических агентов в системах безопасности

Аннотация. *Сформулирвана проблема и показана необходимость разработки методологии моделирования поведения антагонистических агентов а системах безопасности. Представленная концепция реализуется на трех уровнях, а именно: уровне системы безопасности в целом, уровне индивидуальных агентов и уровне группы агентов. Представлены пять этапов реализации концепции. На первом этапе предлагается проводить анализ защищаемых бизнес-процессов и угроз этим процессам. В качестве базовой модели этого этапа предлагается онтологическая модель как носитель знаний об исследуемой прлеметной области. Представлен подход к автоматизации построения онтологии, ориентированный на интеллектуальный анализ текстов на естественных языках, а именно, текстов статей, опубликованных в научных журналах. На второй и третьей стадии построения методологии предлагаются модели индивидуального и группового поведения агентов систем кибербезопасности. Представленные модели отражаются рефлексивные свойства агентов, оказывающие влияние на процессы принятия решений и обучения. Разработанные модели позволили сформировать модельный базис самоорганизации системы безопасности. Практическим приложеним описанных моделей является алгоритм определения реализации наиболее вероятной угрозы, исходя из стоимостных показателей угроз и вероятностей их осуществления. Это может обеспечить эффективное распределение ограниченных финансовых средств инвестирования в систему кибербезопасности.*
Ключевые слова: *кибербезопасность, антагонистические агенты, методология моделирования, рефлексивный агент, мультиагентные системы, контур бизнес-процессов.*

Отримано *25 листопада 2019 року*, затверджено редколегією 18 *грудня 2019 року*