Knowledge Base System for Risk Analysis of the Multi-step Computer Attacks

Gulnara Yakhyaeva and Aleksey Ershov
Novosibirsk State University, Department of Information Technology, Pirogova st. 2, Novosibirsk, Russian Federation

Keywords: Information Security, Cyber Threats, Multi-step Attack, Knowledge Base, Description Logic, Interpretation, Case-based Model, Fuzzy Model, Generalized Fuzzy Model.

Abstract: This work describes the module of the "RiskPanel" software system, performing risk analysis of multi-step computer attacks. The module is based on statistical analysis of actual computer attack precedents. At the user's request the system calculates objective probability of information security risks, taking into account all possible multi-step attacks (i.e. possible combinations of known attacks). The estimation of probability is presented as an interval because we don't always have a full description of real attacks.
The task of this work is described using the model-theoretic formalism. The first step is to build a knowledge base of computer attacks. The formal description of the knowledge base structure is made with the Description Logic. Formalization of estimated (fuzzy) judgments is made in the language of the Fuzzy Model Theory. The article contains algorithms for calculation of probabilistic risk intervals and describes program implementation of the developed methods.

1 INTRODUCTION

The information risk management is currently one of the most important and dynamically developing directions of strategic and operational management in the field of information security. Computer networks play an important role in many areas. Successful cyber attacks lead to considerable financial, image and other losses and provoke the increase in the number of potential attackers. Rising demands lead to complexity of structure and increasing size of the networks, which in its turn leads to increased complexity of security analysis and delay in application of protective countermeasures.

The main task of computer security specialists is rapid response to changes of the current security status of all the components of the corporate information system. For this purpose it is useful to have a software system that allows quickly determining the type of an attack without any special skills, learning the latest information about possible consequences of the attack and how to prevent them.

Computer attacks can be divided into two classes: single-step and multi-step attacks (Dawkins and Hale, 2004). In a single-step attack, the attacker exploits a vulnerability, commits an attack, and reaches his ultimate goal. Multi-step attacks are more complicated. The attacker can exploit the vulnerability not to immediately achieve the goal, but to open a new vulnerability, which can be used for carrying out another attack. In this case, the ultimate objective is achieved at the end of the chain of multiple attacks.

There are two peculiar properties of multi-step attacks: random relationship between multiple attack steps and uncertainty in the description of the attack steps of a multi-step attack (Zhang, et al., 2014; Zhai and Zhou, 2011). Therefore, it is impossible to manually analyze all possible combinations of the attacks.

One of the possible approaches to the analysis of network security includes attack modeling and impact assessment based on attack graphs (Xinming Ou, 2006). The concept of "attack graph" is quite simple. An attack graph is a graph that represents all possible sequences of attacks that allow the attacker to reach his goals through multi-step exploitation of vulnerabilities.

Attack graphs can be a useful tool in different areas of computer security. A system administrator can use attack graphs for:

- building a list of attacks to which the system is vulnerable in the current configuration;

Yakhyaeva, G. and Ershov, A.
Knowledge Base System for Risk Analysis of the Multi-step Computer Attacks.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 2*, pages 143-150
ISBN: 978-989-758-187-8

– search for ways how an attacker can achieve a specific purpose;
– determination of the countermeasures which should be taken to ensure the security of the system.

This paper examines the precedent approach to building attack graphs. This approach is based on the model-theoretic formalization of domain ontologies (Palchunov, 2006; Palchunov, 2008). The formal description of the structure of the knowledge base is made with the Description Logic (Baader, et al., 2007). Formalization of estimated (fuzzy) judgments is made in the language of the Fuzzy Model Theory (Pal'chunov and Yakhyaeva, 2015). Software implementation of the knowledge base is made with the Neo4j[1] graph DBMS.

2 RELATED WORK

At present there are more than a hundred software systems for information risk management. All of them can be divided into two groups (Alhomidi and Reed, 2014):

• basic software systems for qualitative risk analysis;

• full analysis software systems for quantitative risk analysis.

Basic software systems are commonly used by the companies of the 3rd CMM maturity level (Carcary, 2013).

Software systems for full risk analysis include systems with more advanced tools of risk analysis and management. Such tools are in demand among the organizations of the 4th and 5th CMM maturity levels. The fourth-level organizations address the measurement of parameters, characterizing the information security policy. Technology of information security management remains the same, but the risk analysis phase includes quantitative methods of estimation of the residual risk parameters and effectiveness of various countermeasures in risk management. Different kinds of optimization problems in the field of information security are solved on the fifth level. The most famous software systems of this class are CRAMM[2], RiskWatch[3], Digital Security[4] and OCTAVE[5].

[1] neo4j.com
[2] www.cramm.com
[3] www.riskwatch.com
[4] www.dsec.ru
[5] www.cert.org/octave

There is currently no commonly used definition of the "attack graph" concept, and various research groups often mean different things by this word combination.

One of the approaches (Sheyner, et al., 2002) is to define the attack model, which is a finite state machine with multiple states and possible transitions between them, where each element of the set of states describes the state of the entire computer network at the current stage of the computer attack. The attack graph of the specified model is the set of finite sequences of transitions from the initial state of the system in such states that violate specified security property. This approach has one major drawback. With the growth of the number of nodes in a computer network, the number of possible states of the network increases exponentially, making the creation of such attack graph practically useful only for small networks with a small number of vulnerabilities.

Another approach is based on building a logical attack graph (Wayne and Boyer, 2006). In this approach the graph node is a logical expression. It describes not the state of the network as a whole, but only one aspect of this state. Graph edges represent causal connections between the network configuration and the potential privileges that an attacker can get.

This logical approach has two certain advantages over the "attack scenario" approach. Firstly, logical attack graph clearly defines the causal relationship between the system configuration and potential attacker's privileges. In the case of "scenario" approach it would be necessary to take into account the full description of the system state on one or several of the preceding steps to determine the causes of dangerous situation. While logical graph represents that causal relationship with the edges of the graph. It is possible to restore all possible attack scenarios by performing a simple depth-first traversal of the logical graph. Secondly, the size of the logical graph always depends polynomially on the size of the network, and the size of the scenario graph in the worst case has an exponential dependence on the size of the network.

This paper introduces one more semantics of the "attack graph" concept. In the proposed approach a graph node is a precedent of a computer attack with several properties. The edges of the graph are based on the values of two properties of an attack precedent: necessary preconditions to commit this attack, and the consequences that the attack has for the system. The oriented edge from graph vertex A to vertex B is built if the consequences of

committing attack A open the possibility of carrying out the attack B. The multi-step attack is a path through this oriented graph, each vertex of which is reachable from the previous one by a single step over a directed edge.

3 THE KNOWLEDGE BASE ON COMPUTER SECURITY

3.1 Knowledge Base Structure

Let's consider the domain $\Delta =$ "computer attacks". All atomic concepts of the subject area are divided into six classes:

1) $\mathbb{P}_1$: "Symptoms";
2) $\mathbb{P}_2$: "Threats";
3) $\mathbb{P}_3$: "Vulnerabilities";
4) $\mathbb{P}_4$: "Consequences";
5) $\mathbb{P}_5$: "Losses";
6) $\mathbb{P}_6$: "Countermeasures".

Each of these concept classes is represented in the database as a treelike structure, and each attack case is characterized by concepts from these classes (Yakhyaeva and Yasinskaya, 2014). A set of concepts for each class is composed by parsing the database of computer security precedents, the National Vulnerability Database (the NIST[6] agency). The NVD is a USA state project, a regularly updated database of computer vulnerabilities with the descriptions (or references to the descriptions in the other Internet resources) of vulnerabilities discovered in various software systems and components. The descriptions of these vulnerabilities contain information on which version of the product is affected by this vulnerability, how it could be exploited, what measures can be taken to remedy this vulnerability, and other information.

Table 1 shows some examples of atomic concepts for each of the classes.

Hereafter let's denote the atomic concepts by $p_i^{(j)}$, where the top index means that $p_i^{(j)} \in \mathbb{P}_j$. For formalization of knowledge in the domain Δ the binary relation $R \subseteq \mathbb{P}_4 \times \mathbb{P}_3$ is introduced. The semantics of this relation is the following: $\langle p_i^{(4)};\ p_j^{(3)} \rangle \in R$ iff the consequence (of some attack) $p_i^{(4)}$ opens vulnerability $p_j^{(3)}$.

Table 1: Examples of concepts for each of the above classes.

Concept class	Example of atomic concept
Symptoms	Outgoing traffic increase
Threats	Trojan Win32 Antavmu
Vulnerabilities	Buffer overflow
Consequences	Denial of service
Losses	Loss of availability
Countermeasures	Antivirus software

On the basis of this relation the atomic role r is defined.

Therefore, in the domain Δ we will examine a finite set $\mathbb{P} = \mathbb{P}_1 \cup \mathbb{P}_2 \cup \mathbb{P}_3 \cup \mathbb{P}_4 \cup \mathbb{P}_5 \cup \mathbb{P}_6$ of atomic concepts and one atomic role r. The set of all concepts CON_Δ of the domain Δ is composed according to the standard syntax of the Description Logic. All the concepts from CON_Δ are divided into *role concepts* (i.e. using an atomic role r) and *non-role concepts*.

Terminology of the object domain Δ or $\boldsymbol{TBox_\Delta}$ (i.e. intensional knowledge about the domain) consists of the set of all the specialization axioms, representing the hierarchical structure of concept classes.

The second component of a knowledge base is the *World Description* or $\boldsymbol{ABox_\Delta}$ (i.e. extensional knowledge about the domain). In the $\boldsymbol{ABox_\Delta}$, individuals are firstly introduced by giving them names, and then the asserts properties of these individuals (Baader, et al., 2007).

In the domain Δ lets define individuals as the precedents of computer attacks. The main sources of information about computer attacks for RiskPanel system are corporate databases, NIST and MITRE[7]. Each attack $\boldsymbol{e}$ is characterized by the presence/absence of certain concepts from each class of concepts $\mathbb{P}_j$. Thus, for each attack we build a finite set of concept assertions of the following two kinds:

$$p_i^{(j)}(\boldsymbol{e}) \quad \text{or} \quad \neg p_i^{(j)}(\boldsymbol{e}).$$

The concept assertion of the kind $r(\boldsymbol{e_i}, \boldsymbol{e_j})$ is defined in the following way. There is concept assertion $r(\boldsymbol{e_i}, \boldsymbol{e_j})$ iff there are such concept assertions $p_k^{(4)}(\boldsymbol{e_i})$ and $p_l^{(3)}(\boldsymbol{e_j})$ that $\langle p_k^{(4)}, p_l^{(3)} \rangle \in$

[6] http://www.nist.gov/

[7] http://www.mitre.org/

R^T (where R^T is the transitive closure of the relation R).

Here the composition of the *World Description for object domain Δ* or $\boldsymbol{ABox_\Delta}$ is finished. The pair $\langle \boldsymbol{TBox_\Delta}, \boldsymbol{ABox_\Delta} \rangle$ is the Knowledge Base of the domain Δ. As new concepts of this domain (i.e. New threats, vulnerabilities, etc.) or new precedents of computer attacks appear, the Knowledge base is expanded. However, the structure of the knowledge base remains unchanged.

3.2 Model-theoretic Formalization of Estimated Judgements

To perform statistical analysis of the data we need the ***precedent model*** and the ***fuzzy model*** of the domain (Pulchunov and Yakhyaeva, 2005; Yakhyaeva, 2007).. These models are composed, based on the ***interpretation*** set of the Knowledge Base $\mathbb{B}_\Delta$.

Consider a finite set of the computer attacks

$$\mathbb{E} = \{\boldsymbol{e_1}, \dots, \boldsymbol{e_n}\}$$

which was used for describing $\boldsymbol{ABox_\Delta}$. Let $\mathbb{E}$ be a domain of quantification. Then r is a binary relation over this domain, and $\mathbb{P}$ is a collection of unary relations as well. Denote $\sigma_\Delta = \mathbb{P} \cup \{r\}$. Then

Definition 1. Algebraic system $\mathfrak{A}_\Delta$=$\langle \mathbb{E}, \sigma_\Delta \rangle$ is called ***interpretation*** of $\mathbb{KB}_\Delta$ if $\mathfrak{A}_\Delta \vDash \boldsymbol{ABox_\Delta}$ (i.e. for each concept assertion $\varphi \in \boldsymbol{ABox_\Delta}$ we have $\mathfrak{A}_\Delta \vDash \varphi$).

Definition 2. *An ordered triple* $Case(\mathfrak{A}_\Delta) \leftrightharpoons \langle \{a\}, \sigma_\Delta, \tau \rangle$ *is called a* ***precedent model*** *of the domain Δ, generated by the interpretation* $\mathfrak{A}_\Delta = \langle \mathbb{E}, \sigma_\Delta \rangle$*, provided that for every concept* $p(x) \in CON_\Delta$ *we have*

$$\tau(p(a)) = \{\boldsymbol{e} \in E \mid \mathfrak{A}_\Delta \vDash p(\boldsymbol{e})\}.$$

In the precedent model each concept is associated with the set of computer attack precedents which have this concept as a characteristic. Note that from the model-theoretical point of view, the model $Case(\mathfrak{A}_\Delta)$ is a Boolean-valued model. In this Boolean-valued model each proposition of the signature $\sigma_\Delta \cup \{c_a\}$ is associated with an element of Boolean algebra $\rho(\mathbb{E})$.

Most methods of statistical processing of data use objective and/or subjective probability. The objective probability refers to the relative frequency of occurrence of any event in the set of observations or the ratio of the number of desired events to the total number of observations. The subjective probability refers to the degree of confidence of some expert or group of experts that this event will actually happen.

In the proposed approach the concept of fuzzy model is used to describe objective probabilities.

Definition 3. *An ordered triple* $Fuz(\mathfrak{A}_\Delta) \leftrightharpoons \langle \{a\}, \sigma_\Delta, \mu \rangle$ *is called a* ***fuzzy model*** *of the domain Δ, generated by the interpretation* $\mathfrak{A}_\Delta = \langle \mathbb{E}, \sigma_\Delta \rangle$*, provided that for every concept* $p(x) \in CON_\Delta$ *we have*

$$\mu\big(p(a)\big) = \frac{\|\{\boldsymbol{e} \in \mathbb{E} \mid \mathfrak{A}_\Delta \vDash p(\boldsymbol{e})\}\|}{\|E\|}.$$

In the fuzzy model, the truth values of the statements (concepts) are the numbers from the interval $[0, 1]$, which represent the objective probability of concept belonging to a random attack precedent. A more detailed description of the properties of the precedent and fuzzy models can be found in the works of (Yakhyaeva, 2009; Pal'chunov and Yakhyaeva, 2015).

Note that the information about precedents from the Internet can be not complete. Then (according to the "open-world semantics" paradigm) we have a set of different interpretations of the $\mathbb{KB}_\Delta$. From the perspective of the model theories, it is a class of models generated by the set of sentences $\boldsymbol{ABox_\Delta}$. Denote the class of all the interpretations of $\mathbb{KB}_\Delta$ as

$$I_\Delta = \{\mathfrak{A}_\Delta \mid \mathfrak{A}_\Delta \vDash \boldsymbol{ABox_\Delta}\}.$$

Definition 4. *An ordered triple* $\mathfrak{A}_{GF} \leftrightharpoons \langle \{a\}, \sigma_\Delta, \xi \rangle$ *is called the* ***generalized fuzzy model*** *of the domain Δ, generated by the class of interpretations* I_Δ*, provided that for every concept* $p(x) \in CON_\Delta$ *we have*

$$\xi\big(p(a)\big) = \big\{\mu\big(p(a)\big) \mid Fuz(\mathfrak{A}_\Delta) = \langle \{a\}, \sigma_\Delta, \mu \rangle \; and \; \mathfrak{A}_\Delta \in I_\Delta\big\}.$$

Consider the concept $p(x) \in CON_\Delta$. Let

$$\alpha_p = \|\{\boldsymbol{e} \in \mathbb{E} \mid \forall \mathfrak{A}_\Delta \in I_\Delta \colon \mathfrak{A}_\Delta \vDash p(x)\}\|$$

and

$$b_p = 1 - \|\{\boldsymbol{e} \in \mathbb{E} \mid \forall \mathfrak{A}_\Delta \in I_\Delta \colon \mathfrak{A}_\Delta \nvDash p(x)\}\|.$$

In paper ((Yakhyaeva & Yasinskaya, 2015)) it was proven that

$$\inf \xi\big(p(a)\big) = \frac{\alpha_p}{\|E\|}$$

and

$$\sup \xi\big(p(a)\big) = \frac{\beta_p}{\|E\|}.$$

Note that if $p(x) \in CON_\Delta$ is not a role concept than

$$\xi\big(p(a)\big) = \Big\{\frac{\alpha_p}{\|E\|}, \frac{\alpha_p + 1}{\|E\|}, \dots, \frac{\beta_p}{\|E\|}\Big\}.$$

And if $p(x)$ is a role concept than

$$\xi(p(a)) \subseteq \left\{\frac{\alpha_p}{\|E\|}, \frac{\alpha_p+1}{\|E\|}, \ldots, \frac{\beta_p}{\|E\|}\right\}.$$

4 ALGORITHMS OF RISK ANALYSIS

Risk analysis is the systematic use of available information to determine how often specified events may occur and the magnitude of their consequences[8]. In our case, risk analysis is determining the probability of threats, vulnerabilities, consequences and losses which may happen as a result of a computer attack on a corporate information system. In other words, the problem of risk analysis is finding the infimum and the supremum of truth values of concepts from CON_Δ on the generalized fuzzy model $\mathfrak{A}_{GF}$.

Note that if $p(x) \in CON_\Delta$ is a non-role concept, then it describes the state which the system will have after a single-step computer attack. The article (Yakhyaeva and Yasinskaya, 2015) describes the algorithm `getPDNFVerityOnCase`, computing truth values for such concepts. The algorithm is proven to be correct. The work (Yakhyaeva, et al., 2014) describes the module of the RiskPanel system, implementing risk analysis for single-step attacks in the question-answer mode.

On the other hand, role concepts describe properties of multi-step attacks. This work proposes algorithms for answering the following questions:

(Q1) What is the probability that after some multi-step attack the system will have the state, matching the description of non-role concept $p(x)$?

(Q2) What is the probability that after any multi-step attack the system, sooner or later, will definitely have the state, matching the description of non-role concept $p(x)$?

Questions (Q1) and (Q2) can be formalized with concepts

$$\mathbb{p}_1 = (p \vee \exists r.p) \text{ и } \mathbb{p}_2 = \big(p \vee \forall r.(\exists r.p)\big),$$

where p is any non-role concept.

Our task is to find the infimum (denoted by $\inf(\mathbb{p}_i)$) and the supremum (denoted by $\sup(\mathbb{p}_i)$) of truth values of these concepts on the generalized fuzzy model $\mathfrak{A}_{GF}$. Let us consider the graph $\boldsymbol{G} = \langle \mathbb{E}, r \rangle$ defined on the set of attack precedents. We perform postfix traversal of $\boldsymbol{G}$. As a result, every graph vertex is assigned with one of the three values: TRUE, UNKNOWN or FALSE. Therefore

$$\inf(\mathbb{p}_i) = \frac{n(TRUE)}{N},$$

$$\sup(\mathbb{p}_i) = \frac{1 - n(FALSE)}{N},$$

where $n(TRUE)$ is the number of vertices in the graph $\boldsymbol{G}$ labeled $TRUE$, $n(FALSE)$ is the number of vertices in $\boldsymbol{G}$ labeled $FALSE$ and N – total amount of vertices in $\boldsymbol{G}$.

Here we provide pseudocode of the two algorithms, FuzGLEMP (Fuzzy graph labeling for the extended modality of possibility) and FuzGLEMN (Fuzzy graph labeling for the extended modality of necessity), for labeling graph $\boldsymbol{G}$ with concepts $\mathbb{p}_1$ and $\mathbb{p}_1$ respectively (see Table 2 and Table 3).

Table 2: Algorithm FuzGLEMP.

```
PG; // precedent graph
E; // expression from CONΔ

// queue for labeling
Q = {leafs of PG};

while (Q ≠ Ø) {
  // the first vertice in the Q
  V = Q.take();
  if (V is leaf of PG) {
    // label V with the result of
    // evaluation of E on V
    label(V, eval(E, V));
    Q.add({all immediate ancestors of V that are not in
Q yet});
  } else {
    if (all immediate successors of V are labeled) {
      D = {labels of all immediate successors of V};
      if (D contains TRUE) {
        label(V, TRUE);
      } else if (D contains only UNKNOWN and FALSE)
{
        if (eval(E, V) == TRUE) {
          label(V, TRUE);
        } else {
          label(V, UNKNOWN);
        }
      } else if (D contains only FALSE) {
        label(V, eval(E, V));
      }
    } else {
      Q.add(V);
    }
  }
}
```

[8] http://www.palisade.com/risk/risk_analysis.asp

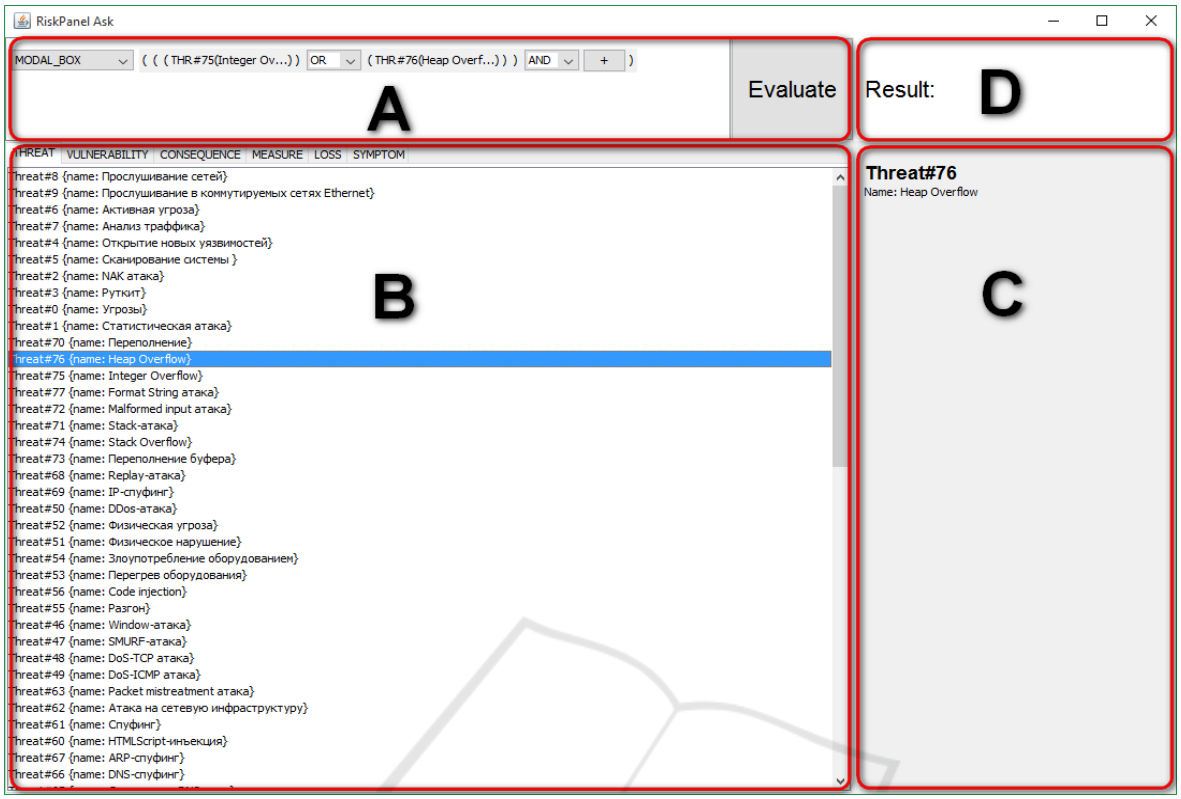

Figure 1: The main window.

Both algorithms are based on the bottom-up graph labeling algorithm from ((Blackburn, et al., 2007), which was modified to work with non-complete knowledge about computer attack precedents.

Table 3: Algorithm FuzGLEMN.

```
PG; // precedent graph
E; // expression from CONΔ

// queue for labeling
Q = {leafs of PG};

while (Q ≠ Ø) {
  // the first vertice in the Q
  V = Q.take();
  if (V is leaf of PG) {
    // label V with the result of
    // evaluation of E on V
    label(V, eval(E, V));
    Q.add({all immediate ancestors of V that are not in
Q yet});
  } else {
    if (all immediate successors of V are labeled) {
      D = {labels of all immediate successors of V};
      if (D contains FALSE) {
        label(V, FALSE);
      } else if (D contains only UNKNOWN and
TRUE) {
        if (eval(E, V) == FALSE) {
          label(V, FALSE);
        } else {
          label(V, UNKNOWN);
        }
      } else if (D contains only TRUE) {
        label(V, eval(E, V));
      }
    } else {
      Q.add(V);
    }
  }
}
```

5 SOFTWARE INTERFACE DESCRIPTION

We developed a software system called RiskPanel asessentially a workplace for experts to ensure the security of corporate information, based on the methodology of generalized fuzzy models (Pulchunov, et al., 2011). RiskPanel has a module for analysis of the various multi-step computer attacks. Consider the module interface.

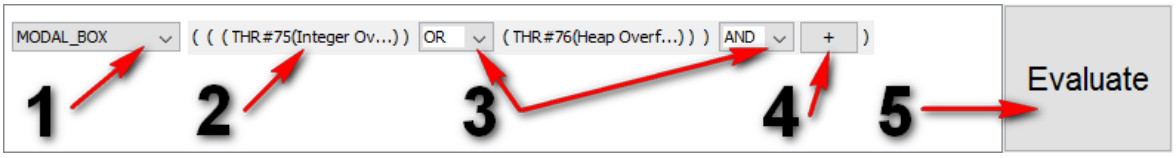

Figure 2: Input area.

The main window (Fig. 1) has four main areas:

- A — the input area
- B — the area for selection of atomic concepts
- C — the area with detailed information about atomic concepts
- D — the output area

Input area A allows user to enter the specific question (Fig. 2).

First of all, the user selects the type of his question (drop-down list (1)). Currently there are only two types: MODAL_BOX and MODAL_DIAMOND (implementing the concepts $\mathbb{p}_1$ and $\mathbb{p}_2$ respectively).

Secondly the user composes non-role concept $p(x)$, which is what the question is related to. Atomic concepts (2), necessary for composition of the concept $p(x)$ can be selected in the main window area **B**. Clicking with the left mouse button on the concept shows all the available information related to this concept in the area **C**. Logical operators, connecting atomic concepts, can be set with drop-down selectors (3). The new operator or an atomic concept can be added by pressing the button (4). If the newly added item is an operator, then its operands will be empty, and buttons like (4) are displayed instead of them.

When the statement is composed, the user can press the button (5) to start computation of the truth value of the statement. If the statement is not complete (e.g. there are still some empty operands left), the user will be warned about this. The truth value cannot be calculated until the user composes the complete statement.

If the computation is successful, the area D will contain the truth probability interval of the composed statement.

6 CONCLUSIONS

This article proposes methods of quantitative risk analysis of multi-step computer attacks. These methods are based on the combination of the precedent approach to knowledge representation and description logic.

Domain knowledge is represented as a finite set of real attack precedents, i.e. a set of semi-structured natural language texts.

Based on these precedents, in TBox and ABox terms of the Discription Logic the Knowledge Base is built. For each interpretation of the Knowledge Base the precedent and the fuzzy model of the domain is generated. The class of fuzzy models, generated by the set of all possible interpretations is a generalized fuzzy model. This model is used for evaluation of truth values of formula descriptions of information risks.

Developed methods are implemented in one of the modules of the “RiskPanel” software system.

ACKNOWLEDGMENT

The reported study was partially supported by RFBR, research project No. 14-07-00903-a.

REFERENCES

Alhomidi, M. & Reed, M., 2014. Attack graph-based risk assessment and optimization approach. *International Journal of Network Security & Its Applications (IJNSA),* 6(3), pp. 31-43.

Baader, F., McGuinness, D., Nardi, D. & Patel-Schneider, P., 2007. *The description logic handbook: Theory, implementation, and applications.* 2-d ред. Cambridge : Cambridge University Press.

Blackburn, P., Van Benthem, J. & Wolter, F., 2007. *Handbook of Modal Logic.* Amsterdam: Elsevier.

Carcary, M., 2013. IT Risk Management: A Capability Maturity Model. *The Electronic Journal Information Systems Evaluation,* 16(1), pp. 3-13.

Dawkins, J. & Hale, J., 2004. *A Systematic Approach to Multi-Stage Network Attack Analysis.* Washington, Proceedings ofthe Second IEEE International Information Assurance Workshop (IWIA '04), IEEE Computer Society.

Palchunov, D., 2006. Simulation of thinking and formalization of reflection: I. Model-theoretic formalization of the ontology and reflection. *Filosofiya nauki,* 31(4), pp. 86-114.

Palchunov, D., 2008. Simulation of thinking and formalization of reflection: II. Ontologies and

formalization of concepts. *Filosofiya nauki,* 37(2), pp. 62-99.

Pal'chunov, D. & Yakhyaeva, G., 2015. Fuzzy logics and fuzzy model theory. *Algebra and Logic,* 54(1), pp. 74-80.

Pulchunov, D. & Yakhyaeva, G., 2005. Interval fuzzy algebraic systems. *Proceedings of the Asian Logic Conference ,* pp. 23-37.

Pulchunov, D., Yakhyaeva, G. & Hamutskya, A., 2011. Software system for information risk manadgement "RiskPanel". *Programmnaya ingeneriya,* Том 7, pp. 29-36.

Sheyner, O. и др., 2002. *Automated Generation and Analysis of Attack Graphs.* Oakland, California, IEEE Symposium on Security and Privacy.

Wayne, F. & Boyer, M. A. M. X. O. A., 2006. *A Scalable Approach to Attack Graph Generation.* New York, CCS '06 Proceedings of the 13th ACM conference on Computer and communications security.

Xinming Ou, W. F. B. M. A. M., 2006. *A Scalable Approach to Attack Graph Generation.* New York, б.н., pp. 336-345.

Yakhyaeva, G., 2007. *Fuzzy model truth values.* Bratislava, Slovak Republic, Proceedings of the 6-th International Conference Aplimat, pp. 423-431.

Yakhyaeva, G., 2009. *Logic of Fuzzifications.* Tumkur, Proceedings of the 4th Indian International Conference on Artificial Intelligence (IICAI-09), pp. 222-239.

Yakhyaeva, G. & Yasinskaya, O., 2014. Application of Case-based Methodology for Early Diagnosis of Computer Attacks. *Journal of Computing and Information Technology,* 22(3), p. 145–150.

Yakhyaeva, G. & Yasinskaya, O., 2015. *An Algorithm to Compare Computer-Security Knowledge from Different Sources.* Barcelona, Spain, Proceedings of the 17th International Conference on Enterprise Information Systems, pp. 565-572.

Yakhyaeva, G., Yasinskaya, O. & Karmanova, A., 2014. Probabilistic question-answering system in the field of computer security. *Vestn. Novosib. gos. un-ta. Seriya: Informacionnye tehnologii,* 12(3), pp. 132-145.

Zhai, G. & Zhou, S., 2011. Construction and implementation of multistep attacks alert correlation model. *Journal of Computer Applications,* 31(5), p. 1276–1279.

Zhang, Y., Z. D. & Liu, J., 2014. The Application of Baum-Welch Algorithm in Multistep Attack Hindawi Publishing Corporate Scientific World Journal Volume 2014. *Scientific World Journal,* Том 2014, p. 7.