Effectiveness of Placing Attack Detection Detectors

Vyacheslav L. Tokarev [1] ,Aleksei A. Sychugov[2]

[1]Tula State University, Institute of Applied Mathematics and Computer Science, Tula, Russian Federation
[2]Tula State University, Institute of Applied Mathematics and Computer Science, Tula, Russian Federation
[2]xru2003@list.ru

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: The effectiveness of intrusion detection systems based on immune detectors requires rational placement of immune detectors on network nodes. For this, it is necessary to determine the risk of violating the information security of the computer network. Evaluating the network nodes' security risk is complicated by the number of vulnerabilities. To solve this problem, it is proposed to use a statistical formal model based on Markov chains in combination with vulnerability analysis metrics. To analyze vulnerabilities, we use scoring scores as metrics of basic, temporal, and contextual types. The main component of the proposed model is the graph of possible trajectories. An example network was used to test the proposed model. A formal model of information security risk assessment allows identifying critical nodes along the path of access to the target node that makes it possible to install immune detectors on these nodes in order to improve the protection system.

Keywords: Information security, Intrusion detection systems, Immune detectors, Markov chains, Scoring scores, Security risk of network nodes

1. Introduction

One of the most effective means of timely detection of attacks in computer networks (Kemmerer & Vigna, 2002) is intrusion detection systems (IDS), built on the basis of immune detectors (Rajasooriya, Tsokos, & Kaluarachchi, 2016). Such systems allow us to detect attacks of various classes, including those that are still unknown.

However, rational placement of immune detectors at network nodes is important for the effectiveness of such IDS. It is shown (Tokarev, Sychugov, & Anchishkin, 2019; Sychugov et al., 2019) that the composition and placement of immune detectors then allow achieving the greatest effect when they control nodes with a relatively high risk of information security violations.

Network security risk assessment is complicated by the fact that the vulnerability is often not the only one. It can be multi-stage, multi-variant, and cover multiple nodes.

To solve this problem, it is proposed to use a statistical formal model based on Markov chains in combination with vulnerability analysis metrics, which allows us to identify critical nodes where violators can be most dangerous. Based on the information obtained using the model, the network administrator can install immune detectors on these nodes, which will significantly improve the protection system, that is, reduce the value of the overall security indicator.

The proposed formal model can be used to assess the information security risks of networks of various topologies.

2. Background of the Proposed Approach

2.1. Vulnerability assessment metrics

CVSS (Common Vulnerability Scoring System) scoring is often used as vulnerability estimate (Chawla, Sharma, & Rawal, 2019; Li et al., 2019; Ruohonen, 2019), which uses three types of metrics: basic, temporary, and contextual. The main metrics and their qualitative values are shown in Table 1.

To get estimates in the range from 0 to 10, the qualitative values of indicators are converted to quantitative ones using the logistics curve (Databank of information security threats, n. d.), and the final metric estimates are obtained using Eq. (1).

$$\begin{aligned} &Exp = \varphi_1(AV, AC, AU);\\ &\mathrm{Imp} = \varphi_2(C, I, A);\\ &ase = \varphi_3(\mathrm{Imp}, Exp), \end{aligned} \qquad (1)$$

For example:

$$\mathrm{Exp} = 20 \cdot \mathrm{AV} \cdot \mathrm{AC} \cdot \mathrm{AU} \ ;$$

$$\mathrm{Imp} = 10{,}41 \cdot [1 - (1 - \mathrm{C}) \cdot (1 - \mathrm{I}) \cdot (1 - \mathrm{A})] \ ;$$

$$\mathrm{Base} = 0{,}6 \cdot \mathrm{Imp} + 0{,}4 \cdot \mathrm{Exp} \ .$$

Here, nodes with a score range from 0 to 3.9 are considered low–vulnerable, 4.0–6.9 are considered medium-vulnerable, and 7.0–10 are considered high-vulnerable.

Table 1. CVSS vulnerability metrics (Li et al., 2019; Ruohonen, 2019)

Names of the indicators		Quality values
Basic metrics (BASE)		
Access characteristics, "Exploitability" (Exp)	Method of obtaining access (AV)	Local, through an adjacent network, network
	Difficulty of obtaining access (AC)	High, medium, low
	Authentication (AU)	Multiple, single, no
Influence on (Imp):	Confidential (C)	Does not apply, partial, full
	Integrity (I)	Does not apply, partial, full
	Availability (A)	Does not apply, partial, full
Temporary metrics (Temp)		
Possibility of using	Not defined, theoretically, there is a concept; there is a scenario, high.	
The fix level	Not defined, temporarily, as recommended, not available.	
The degree of the source reliability	Not determined, not confirmed, not proven, confirmed	
Contextual metrics (Env)		
Probability of damage	Undefined, low, medium, high.	
The density of damage	Undefined, low, medium, high.	
Privacy requirements	Undefined, low, medium, high.	
Integrity requirements	Undefined, low, medium, high.	
Availability requirements	Undefined, low, medium, high.	

2.2. Graph of possible paths

Intruders usually enter computer networks using a chain of exploits, each element of which creates the basis for the next element. The combination of such exploits makes up a chain called the attack trajectory, the combination of which forms a graph of possible trajectories (GPT) ending in a state where the intruder can successfully achieve its goal. There are enough algorithms that have been developed to build GPT attacks (Tokarev, 2014; Jha, Sheyner, & Wing, 2002; Ingols, Lippmann, & Piwowarski, 2006; Mehta, et al., 2006). However, it is very difficult to analyze a network using GPT when the number of nodes and complexity of the network increases, as the complexity of construction and computational costs increase exponentially (Xi et al., 2009; Khlobystova, Abramov, & Tulupyev, 2018a; Suleimanov, Abramov, & Tulupyev, 2018; Khlobystova et al., 2018b; Navlani, 2018).

2.3. Use of Markov chains

To build a formal model of access to a node, it is proposed to use Markov chains that reflect the real behavior of the attacker.

A Markov chain can be defined as a discrete stochastic process (Dynkin, 1959; Bolch et al., 2006) defined on a finite set of states. Then the Markov chain can be represented as a sequence of random variables $x_0, x_1, \ldots, x_n \in S$ satisfying the "Markovian property", i.e.:

$$P\left[X_{n+1} = y \middle| X_0 = x_0,\, X_1 = x_1, \ldots, X_n = x_n\right] = P\left[X_{n+1} = y \middle| X_n = x_n\right]$$

Markov properties mean that: 1) transitions between states are devoid of memory; 2) the transition to the next step depends only on the current state and none of the previous states (Triverdi, 2002; Sahner, Trivedi, & Puliafito, 2012). We can relate these properties to the behavior of the intruder in the sense that the intruder can use different paths (a sequence of nodes) before reaching the target node (Abraham & Nair, 2014; Lawler, 2006; Shmygaleva et al., 2019).

It is supposed: 1) selection of the best relay node depends on three factors, namely: exploitability characterizing the vulnerability of the access subsystem; the impact of vulnerabilities to the confidentiality, integrity and availability, as well as the individual skill of the attacker; 2) the transients do not depend on time; 3) we can determine some matrix of transition probabilities P(x, y) and the initial probability distribution $R = \{r_1, r_2, ..., r_n\}$.

Then, having the matrix P (x, y), the initial risk vector R, using the basic properties of the Markov process, we can determine the risks of nodes and the risk of the entire network.

3. Model Building

The main component of the proposed model is the GPT, which is built by examining the network topology, services running on each node, rules defined on firewalls, and vulnerabilities associated with each node running different services.

The scheme of the proposed simulation is shown in Figure 1.

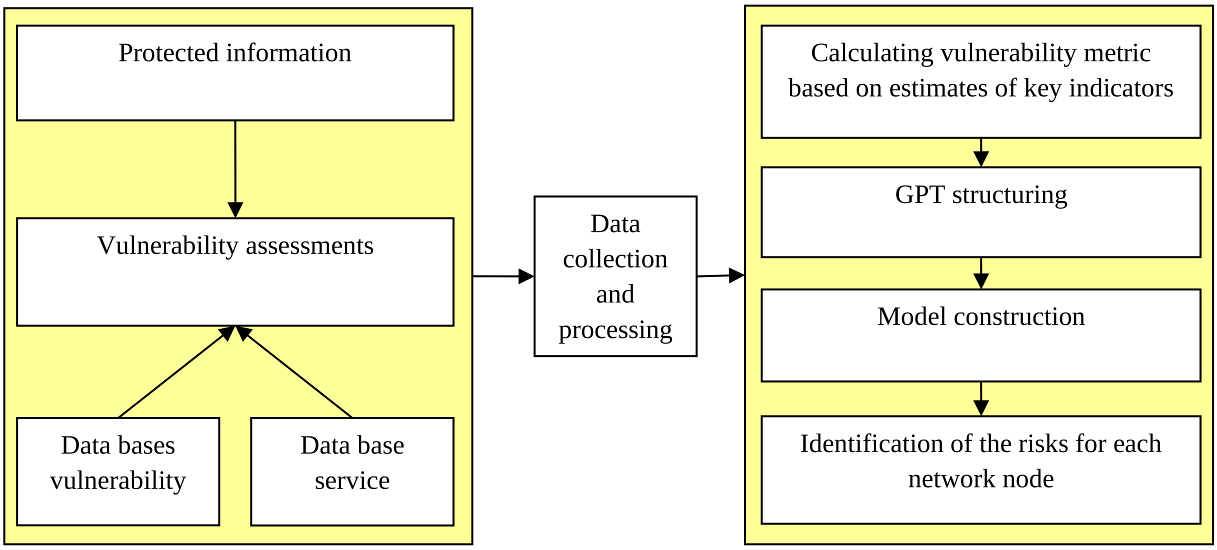

Protected information
Vulnerability assessments
Data bases vulnerability
Data base service
Data collection and processing
Calculating vulnerability metric based on estimates of key indicators
GPT structuring
Model construction
Identification of the risks for each network node

Figure 1. Scheme of simulation

It is assumed that:

(1) there are a limited number of nodes in the network under consideration, and each node is a node, each of which runs different types of services and there may also be different vulnerabilities, for which the CVSS system determines the corresponding Exp and Imp points, which can be used to mark the edges of the GPT used to determine the probability of an intruder using the i-th vulnerability;

(2) the intruder will choose a vulnerability that maximizes the chances of success in compromising the target node's state;

(3) if the intruder, for any reason, completes the attack, it will return to its original state.

The central component of the proposed model is the GPT access to the node. For example, consider a GPT built on three vertices (Figure 2).

In Figure 2a, Si–nodes, Sg–the target node. The number of nodes is equal to the number of stations in the network. Directed edges between two nodes represent the relationship between the corresponding two stations in the network: there are no multiple edges in the original GPT. Dotted edge lines mean that there may be other intermediate nodes between the nodes.

In Figure 2b, an additional node A has been added to the GPT to represent the intruder.

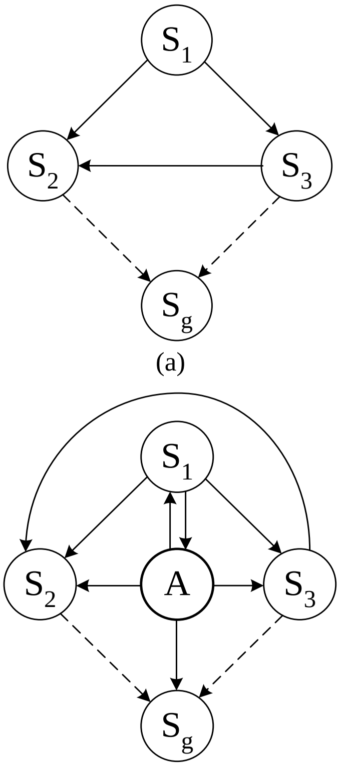
An intruder can attack a node that it has direct access to, and after overcoming the protection of this node, it develops attacks until it reaches the target Sg node. In this case, it is faced with the task of selecting the next node to attack, in order to reach the Sg node. This choice most likely depends on the two parameters outlined above: Exp, which characterizes the difficulty of overcoming the node's protection, and Imp, which characterizes the node's vulnerabilities. Here each node is rated in CVSS points on a scale (0 to 10), where 0 means the most secure node and 10 means the least secure.

In other words, to solve the problem of selecting the next node to attack, the intruder can use the value of the base metric (1).

When making the final decision to move from one node to another, the attacker also relies on their own skills and experience. This is a subjective factor that can have a significant impact on the offender's choice of the next node to attack: it can shift the decision in one direction or another. This factor can be taken into account in the model as a bias factor β. As a result, the selection function can be represented by the following Eq.:

$$\alpha_{jk} = \beta \cdot Exp(v_k) + (1-\beta)\cdot \mathrm{Imp}(v_k), \quad 0 < \beta < 1, \qquad (2)$$

where a_{jk} - is the "benefit" of moving away from the node j to the node k, v_k–vulnerability function, the value of which characterizes the ability to overcome the protection of the k-th node by the intruder; β–the offset coefficient, which takes a value from 0 to 1.



(a)

(b)

Figure 2. Examples of GPT: (a) without the intruder and (b) with the intruder

If the a_{jk} values are defined for each pair of nodes in the network, then its protection from attacks, from the point of view of the intruder, can be characterized by an adjacency matrix:

$$A = \begin{bmatrix} 0 & a_{01} & \cdots & a_{0g} & \cdots & a_{0n} \\ a_{10} & 0 & \cdots & a_{1g} & \cdots & a_{1n} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{n0} & a_{n1} & \cdots & a_{ng} & \cdots & 0 \end{bmatrix}$$

The diagonal values of the matrix A are equal to zero, since there are no obstacles for the violator to move from the j-th node to the j-th node itself. Normalization of the a_{jk} values of the matrix A allows you to get the values of the fiction of belonging to the fuzzy set "Node S_k is available for attack from node S_j":

$$\mu_{jk} = \frac{a_{jk}}{\sum_i a_{ji}} \qquad (3)$$

Then the characteristics of the network in matrix form is Eq. (4):

$$M = DA \qquad (4)$$

where M is the transition matrix that determines whether an intruder can move from one node to another, and D is the diagonal matrix calculated using the normalization rule

$$d_{jk} = \begin{cases} 1 \Big/ \sum_i a_{ji} & if \;\; j = k \\ 0 & otherwise \end{cases} \qquad (5)$$

Let the intruder starts attacking from the start node to the target node. GPT nodes can be used by an intruder if any of the child nodes is true. The risk analysis is based on the relative rank value for each GPT node. The initial value of the risk vector $R = \{r_1, r_2, \ldots, r_n\}$ is calculated based on the number of nodes present in the GPT. If there are *n* nodes in the GPT, then you we set all node ranks equal to 1/n and this initial risk becomes the starting node of the attacker.

The risk value r_k for node *k* is calculated using an iterative procedure until a stable value is obtained. Assuming that $r_k(t-1)$ is the previous risk value of node k, its next value $r_k(t)$ is calculated by the Eq. (6):

$$r_k(t) = \sum_j r_k(t-1)\mu_{jk} \tag{6}$$

In matrix form, Eq. (6) takes the form

$$R_t = R_{t-1}M \ . \tag{7}$$

Risk values are normalized: $0 \le r_k \le 1$, $\sum_j r_k = 1$.

The R value is calculated recursively until it converges $(|R_t - R_{t-1}|)^T(|R_t - R_{t-1}|) \le \varepsilon$, where ε is a specified small positive number. This iteration converges to the stable value of the vector R*, as the eigenvector of the matrix M.

The risk assessment algorithm includes the following steps:

Step 1. Initialization. Each value of the risk vector is assigned an initial value of 1/n.

Step 2. Iterative procedure for correcting risk values before the convergence condition occurs for each vertex of the GPT.

Step 3. Determine the network node protection priorities based on the received risk values.

Step 4. Determining the amount of risks as a general indicator of network information security.

4. Example

A network was used to test the proposed model (Figure 3).

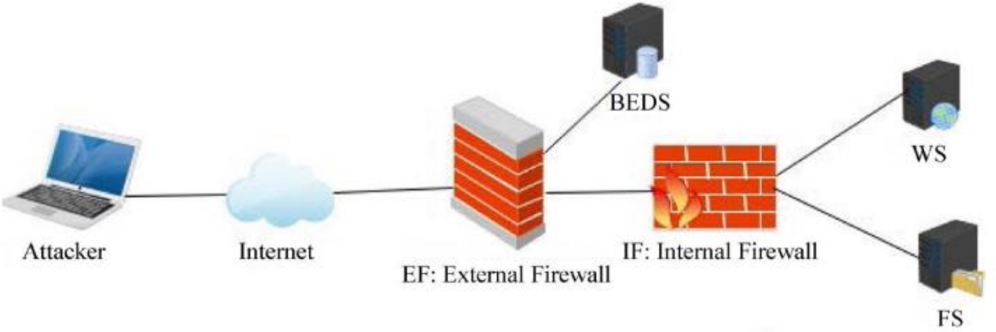


Figure 3. Example of network topology (Hewett & Kijsanayothin, 2008; Ammann et al., 2005)

This network has three target nodes: a public web server (WS), a public file server (FS), and a database backend server (BEDS). It is assumed that the intruder is located outside the network. Packet transmission to the target node is controlled by two firewalls: the external firewall (EF) and the internal firewall (IF). EF allows any packet to be transmitted by the WS and FS server from outside the network and prohibits access to BEDS resources directly from outside the network. IF manages the transmission of packets within the network. A brief description of the firewall rules and network scenario is provided in Table 2.

Table 2. Firewall rules

Source	Destination	Service	Action
All	WS	http	Allow
All	WS	Ftr	Allow
All	FS	ftp	Allow
WS	BEDS	Oracle	Allow
FS	BEDS	ftp	Allow
All	All	All	Deny

Let each of the target nodes contain a single vulnerability. The intruder uses a vulnerability assessment to compromise the node. This is shown in Table 3 below, along with the Exp and Imp vulnerability scores taken from (Sychugov et al., 2019).

Based on the network topology (Figure 3), taking into account the firewall rules and vulnerability scores associated with the corresponding node, a GPT of access to the node is generated (Figure 4).

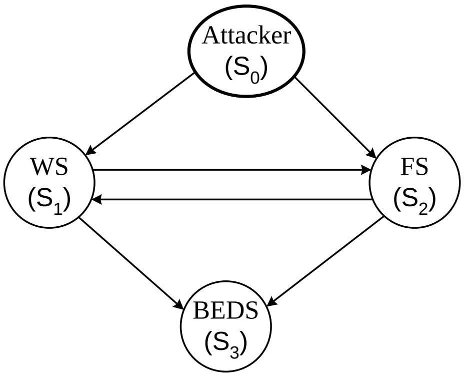


Figure 4. GPT for the example under consideration

The GPT designates the attacker, Web Server, File Server, and Backend Database Server (BIDS) nodes as S_0, S_1, S_2, and S_3 respectively. Edges from all nodes to node S0 are omitted.

Let β = 0.5, then using the expression (2), in accordance with GPT (Figure 4), we can calculate the adjacency matrix A. Since it is assumed that if the intruder stops the attack, it returns to the original node, the elements of the first column of matrix A are equal to 1, and all other elements are defined according to Eq. (2). For example, $a_{12}0.5\cdot 10 + 0.5\cdot 6.4$ This is the value α_{12}, that the intruder uses when choosing to move from node S_0 to node S_1. The remaining elements of the matrix A are also defined:

$$A = \begin{bmatrix} 0 & 8.2 & 9.3 & 0 \\ 1 & 0 & 9.3 & 8.2 \\ 1 & 8.2 & 0 & 8.2 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Table 3. Nodes' vulnerabilities

Node	Vulnerability	CVE-ID	Base	Imp	Exp
WS	Apache Chunked Code	CVE-2002-0392	7.5	6.4	10
FS	Wuftpd Sockprintf	CVE-2003-1327	9.3	10	8.6
BEDS	Oracle Tns listener	CVE-2002-1675	7.5	6.4	10

Next, the matrix A is converted to the matrix M using the Eq. (4), in which the matrix D:

$$D = \begin{bmatrix} 0.05714 & 0 & 0 & 0 \\ 0 & 0.05405 & 0 & 0 \\ 0 & 0 & 0.05747 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The transition matrix M:

$$M = \begin{bmatrix} 0 & 0.47 & 0.53 & 0 \\ 0.054 & 0 & 0.50 & 0.44 \\ 0.057 & 0.47 & 0 & 0.47 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Based on the proposed risk ranking algorithm, the initial risk vector R is determined R= (0.25, 0.25, 0.25, and 0.25). After running the iterative procedure–Eq. (7) and detecting its convergence, the following values are obtained (Table 4).

Table 4. Risks of nodes

Node	Risk
S_0	0.260
S_1	0.245
S_2	0.262
S_3	0.231

From the obtained r_i values, we can conclude that node S_2 is less secure than S_1 and S_3. Therefore, immune detectors should be installed first on the S_2 node. The total amount of risks associated with node S_1, S_2, and S_3 is

0.74. This value can be used as a security indicator, indicating that this network is not very secure with respect to these vulnerabilities and existing access relationships between servers. Therefore, access to protected resources requires better protection.

5. Conclusion

A formal model of information security risk assessment using GPT access to network resources is proposed. A model that uses Markov chains in combination with CVSS scores allows us to analyze vulnerabilities related to the network structure. The model allows us to identify critical nodes that exist in the node access GPT. Based on this information, the network administrator can make an appropriate decision, in particular, to install immune detectors based on priorities.

The proposed risk ranking algorithm is quite flexible in the sense that it allows us to predict the actions of the intruder taking into account their skills and experience, setting the β offset in the calculation of the values of the "benefit" function α. The algorithm was considered relative to the base metric (Base), but it can also be used to assess the risk by using a time metric (Temp) or a context metric (Env).

The proposed model can be integrated into the information security system of existing automated systems or used in the construction of automated systems in a protected version.

Acknowledgments

The study was performed with financial support RFBR, research project No. 19-07-01107\19.

References

1. Abraham, S., & Nair, S. (2014). Cyber security analytics: a stochastic model for security quantification using absorbing Markov chains. Journal of Communications, 9(12), 899–907. https://doi.org/10.12720/jcm.9.12.899-907
2. Ammann, P., Pamula, J., Ritchey, R., & Street, J. D. (2005). A host-based approach to network attack chaining analysis. In: Computer Security Applications Conference (pp. 72-84). IEEE.
3. Bolch, G., Greiner, S., de Meer, H., & Trivedi, K. S. (2006). Queueing networks and Markov chains: Modeling and performance evaluation with computer science applications. John Wiley&Sons. https://doi.org/10.1002/0471791571
4. Chawla, G., Sharma, N., & Rawal, N. K. (2019). IVSV: An improved CVSS base score mechanism with vulnerability type. International Journal of Engineering and Advanced Technology, 8(6), 4946–4950. https://doi.org/10.35940/ijeat.F9245.088619
5. Databank of information security threats. (n. d.) CVSS Calculator v2. https://bdu.fstec.ru/calc.
6. Dynkin, E. B. (1959). Foundations of the theory of Markov processes. Fizmatlit.
7. Hewett, R., & Kijsanayothin, P. (2008). Host-centric model checking for network vulnerability analysis. In Annual Computer Security Applications Conference (ACSAC) (pp. 225–234). IEEE. https://doi.org/10.1109/ACSAC.2008.15
8. Ingols, K., Lippmann, R., & Piwowarski, K. (2006). Practical attack graph generation for network defense. In Proceedings of 22nd Annual Conference on the Computer Security Applications (pp. 121–130). IEEE. https://doi.org/10.1109/ACSAC.2006.39
9. Janshanlo, R.E., Kogut, O.Y., Czerewacz-Filipowicz, K. (2019) Human capital management trends in the innovative economy of Kazakhstan. Polish Journal of Management Studies, 20(2), pp. 267-278. https://doi.org/ 10.17512/pjms.2019.20.2.22
10. Jha, S., Sheyner, O., & Wing, J. (2002). Two formal analyzes of attack graphs. In Proceedings of 15th IEEE Computer Security Foundations Workshop (pp. 49–63). IEEE. https://doi.org/10.1109/CSFW.2002.1021806
11. Kemmerer, R. A., & Vigna, G. (2002). Intrusion detection: A brief history and overview. Computer, 35(04), supl27-supl30, 27–30. https://doi.org/10.1109/MC.2002.1012428
12. Khlobystova, A. O., Abramov, M. V., & Tulupyev, A. L. (2018a). Identification of the most probable trajectories of socio-engineering attacks in risk management associated with users / personnel. In Proceedings of the Conference "Information Technologies in Control" (ITU-2018) (pp. 493–497). SSC JSC Concern CSRI Elektropribor.
13. Khlobystova, A. O., Abramov, M. V., Tulupyev, A. L., & Zolotin, A. A. (2018b). Finding the shortest trajectory of a socio-engineering attack between a pair of users in a graph with transition probabilities. Information and Control Systems, 6, 74–81. https://doi.org/10.31799/1684-8853-2018-6-74-81

14. Lawler, G. F., (2006). Introduction to stochastic processes. CRC Press, Boca Raton.
15. Li, E., Kang, C., Huang, D., Hu, M., Chang, F., He, L., & Li, X. (2019). Quantitative model of attacks on distribution automation systems based on CVSS and attack trees. Information, 10, 251. https://doi.org/10.3390/info10080251
16. Mehta, V., Bartzis, C., Zhu, H., Clarke, E., & Wing, J. (2006). Ranking attack graphs. In International Workshop on Recent Advances in Intrusion Detection (pp. 127–144). Springer. https://doi.org/10.1007/11856214_7
17. Navlani, A. (2018). Predicting employee churn in Python. Datacamp. https://www.datacamp.com/community/tutorials/predicting-employee-churn-python
18. Rajasooriya, S. M., Tsokos, C. P., & Kaluarachchi, P. K. (2016). Stochastic modelling of vulnerability life cycle and security risk evaluation. Journal of Information Security, 7, 269–279. https://doi.org/10.4236/jis.2016.74022
19. Ruohonen, J. (2019). A look at the time delays in CVSS vulnerability scoring. Applied Computing and Informatics, 15(2), 129-135. https://doi.org/10.1016/j.aci.2017.12.002
20. Sahner, R. A., Trivedi, K., & Puliafito, A. (2012). Performance and reliability analysis of computer systems: An example-based approach using the SHARPE software package. Berlin: Springer Science & Business Media.
21. Shmygaleva, T. A., Kupchishin, I., Kupchishin, A. A., & Shafii, C. A. (2019). Computer simulation of the energy spectra of PKA in materials irradiated by protons in the framework of the cascade-probabilistic method. IOP Conference Series: Materials Science and Engineering, 510(1), 012024. https://doi.org/10.1088/1757-899X/510/1/012024
22. Suleimanov, A., Abramov, M., & Tulupyev, A. (2018). Modelling of the social engineering attacks based on social graph of employees communications analysis. In 1st IEEE International Conference on Industrial Cyber-Physical Systems (ICPS 2018) (pp. 801–805). IEEE. https://doi.org/10.1109/ICPHYS.2018.8390809
23. Sychugov, A. A., Meltsov, V. Yu., Kuvaev, A. S., & Grishin, V. M. (2019). Network intrusions detection and prevention method using a team of intelligent agents. Journal of Mechanical Engineering Research and Developments, 42(2), 14–17. https://doi.org/10.26480/jmerd.02.2019.14.17
24. Tokarev, V. L. (2014). Recognition of the opposing side's strategy based on current observations. Reports of the Tomsk State University of Control Systems and Radioelectronics, 2(32), 184–187.
25. Tokarev, V., Sychugov, A., & Anchishkin, A. (2019). Detection of anomalies in the information networks of industrial automation systems based on artificial immune detectors. In 2019 International Russian Automation Conference (RusAutoCon) (pp. 1–5). IEEE. https://doi.org/10.1109/RUSAUTOCON.2019.8867593
26. Trivedi, K. S. (2002). Probability and statistics with reliability, queuing and computer science applications. New York: John Wiley&Sons.
27. Xie, A., Cai, Z., Tang, C., Hu, J., & Chen, Z. (2009). Evaluating network security with two-layer attack graphs. Annual Computer Security Applications Conference, 127–136. https://doi.org/10.1109/ACSAC.2009.22