UDC 004.056
DOI: 10.15587/2312-8372.2016.71677

Луцький Г. М.,
Волокита А. М.,
Якушев О. Ю.,
Регіда П. Г.,
Ву Дик Тхінь

РОЗРОБКА СПОСОБУ ВИЯВЛЕННЯ АТАК В РЕАЛЬНОМУ ЧАСІ НА ОСНОВІ ОБЧИСЛЮВАЛЬНОГО ІНТЕЛЕКТУ

Розроблено спосіб виявлення атак в реальному часі на основі обчислювального інтелекту, який відрізняється застосуванням методу генетичного програмування для побудови профілів атак, і дозволяє оцінювати та виявляти атаки, які ще не були досліджені чи визначені, але їх наслідки вже були виявлені. Було проведено експериментальне дослідження прототипу системи виявлення атак.

Ключові слова: *система моніторингу безпеки, розподілена комп'ютерна система, обчислювальний інтелект.*

1. Introduction

There are a number of possible actions that could compromise the protected computing system; accordingly, there are no fewer factors that indicate the presence or absence of intrusion into the system. Finding analytical solution for the determination of related events in the system to the presence of intrusion into distributed computing systems (DCS) is possible only in partial form for specific types of intrusions and specific events in the DCS. Alternatively, the empirical method of regression models can find a solution to the problem of a more general case, and less intervention of the user (DCS administrator).

Solving problems using a regression model is the search for the expression and its ratios to input parameters substituted in expressions, given the appropriate output values. Genetic programming is one of the method of solving the symbolic regression problem with the ability to create structure of decisions during implementation, and the method of presentation of solutions in the form of trees, which coincides with the structural notation of symbolic expressions.

Thus presenting problems in intrusion detection into DCS as symbolic regression tasks on event data in the system allows creating a monitoring tool for DCS based on symbolic regression, as one of the methods of computational intelligence.

2. Object of the study and its technological audit

The object of the study is a security monitoring system for distributed computing system. The subject of research is the way to implement security monitoring system for distributed computing system based on computational intelligence. There is a problem of detecting intrusion into computing systems, namely the lack of an effective way of monitoring for detecting distributed attacks for the anomalous behavior of the system in real time.

3. Aim and tasks of the study

The aim of the study is to provide such a security monitoring system that will automatically monitor events in a secure information system and detect suspicious events based on analysis of data on previous successful intrusions into other computing systems. Data analysis of previous successful intrusion into DCS is done by using the method of genetic programming.

The main tasks to achieve the aim of the study:

1. Analysis of existing approaches to security monitoring systems for different types of distributed systems and approaches to the analysis of security events to determine the possible compromising system.
2. Development of a new way of security monitoring using genetic programming method for the analysis of security events.
3. Design of models for classical and developed approaches to create a security monitoring system for further analysis and obtaining the results of the experiment.
4. Analysis of experiment results using developed models.

4. Literature review of intrusion detection into distributed systems

Intrusion detection systems (IDS) are the separate class of software tools by which to understand the program, rules and accompanying documentation and information relating to the functioning of information processing system. In [1] the basic features, principles and mechanisms of IDS functioning are considered and in [2] the issue of environment sustainability by information exchange are revealed.

Search of information about a single event in the audit logs of various protection tools. The complexity of the search results from the fact that different protection tools in different register the information about events. For example, information about virus attacks in the case of viral epidemic will be registered simultaneously by means of virus protection, intrusion detection systems, firewalls, and operating system logs in servers and workstations [3].

Monitoring systems in conventional terminology are indicated by the acronym SIM (Security Information Management) or SIEM (Security Information and Event Management) [4].

Currently, the most widely used security information commercial event monitoring system: ArcSight, Cisco

MARS, RSA Envision, NetForensics, NetIQ, Symantec and others. It should be noted that in addition there are also free open source monitoring system. An example of such a system is Prelude Universla SIM [5]. ArcSight ESM allows monitoring of all necessary resources in real time, receiving information at the level of the protection tools and the level of network resources, applications and databases, allowing building a comprehensive monitoring and event management system for information security [6].

A brief comparison of intrusion detection technologies are shown in Table 1.

Table 1

Comparative table of intrusion detection methods

Characteristics	Signature methods	Anomaly methods
Number of detected attacks	Limited by known types of attacks	Limited by IDS analysis methods
The probability of crossing attack	Average	Low
The probability of false alarm	Very low	High
Requirements for computing resources	Average	High

In the absence of mathematical foundations and formalize the process of detecting unauthorized actions and attacks it can be concluded that the current approach is not focused on the development of efficient mathematical support and software of computers, complexes and computer networks [1].

Genetic programming (GP) in the field of computational intelligence is an evolutionary algorithmic methodology inspired by the phenomenon of biological evolution. GP is one of the machine learning method to optimize a population of computer programs according to the fitness value, defined as the ability of the program to perform certain computational problems [7]. GP develops and improves computer programs that are traditionally represented in memory as a tree structure [8]. Thus, GP traditionally preferred programming languages that naturally embody tree structure (e. g., Lisp; other functional programming languages are appropriate). GP compares favorably with algorithms of hill climbing, gradient descent and simulated annealing by the fact that they aren't «greedy». GP and evolutionary computation are the good way to meet the challenges of building a topology of the system and its configuration [9].

The main features, principles and mechanisms of functioning of IDS tasks were considered and analyzed. An analysis of the known methods and approaches to detect attacks made the following conclusions:

1. IDS facilitate solution for information security administrators such problems as the recognition of known and unknown attacks, the analysis of patterns of abnormal actions, monitoring and analysis of user activity, system and network activity.

2. Use of monitoring systems can increase the efficiency of detection and response to incidents of information security. This is achieved by automating the collection and analysis of recording information. Application of monitoring system also improves the efficiency of already installed protection tools.

3. There are two main ways to detect attacks: signature method and anomaly detection method. The first method is simpler in terms of implementation, requires fewer resources from the computing system and ensures fewer false responses. The second method is more reliable in terms of security, as it has the ability to identify new and hitherto unknown attacks, and more flexible to configure. Some IDS combine these two methods to ensure better protection and acceptable performance of the security monitoring systems.

4. Existing systems for monitoring the DCS security do not guarantee detection of all attacks and the lack of false IDS responses.

5. Materials and methods of the study

Metrics of security. Important metrics for intrusion detection system are performance of the percentage of detected attacks or rather the inverse percentage characteristic of escaped attacks. In fact, there are two such characteristics because errors in detecting attacks are the first and the second kind. Detection error of the first kind is when IDS pass off the attack as a normal behavior of the system.

Error of the second kind is when the system believes the intrusion as a normal behavior of the system:

$$K_{\text{I}} = 1 - \frac{N_{det}}{N_{at}}, \quad K_{\text{II}} = \frac{N_{tot} - N_{det}}{N_{at}},$$

where N_{tot} — the total number of IDS responses; N_{det} — the number of detected attacks; N_{at} — the number of attempts to attack the system. It should be noted that the value of K_{II} (coefficient of error of the second kind) may be more than one.

Also as a metric the authors would consider the number of runs for computational intelligence methods (genetic programming), which lead to the correct result. Since the methodologies of computational intelligence are heuristic, their convergence can be proved analytically. Therefore, the authors examined the concept of the success rates of the genetic programming for 100 runs. The criterion for the rub success the authors considered a deviation from the ideal result not more than 5 % on 50 generations of the algorithm — is the ideal case, and the second criterion — getting a result for 200 generations. The system automatically stopped working after reaching the 200-th generation cycle and began again:

$$S_{50} = \frac{M_{det\,50}}{100}, \quad S_{200} = \frac{M_{det\,200}}{100}.$$

As mentioned, there are two main types of intrusion detection systems — profile and anomalous. Both types have certain advantages and disadvantages. IDS on the basis of attack profile requires few resources and less susceptible to errors of the second kind, but can't provide protection against unknown and modified attacks. IDS on the basis of anomalies can adjust to change and unknown attacks, but has a high delay the work, requires a lot of computing resources and an increased sensitivity to errors of the third kind.

The proposed intrusion detection system is a hybrid IDS type. It is based on the use of attack profiles and

is fast and doesn't depend on computing resources; but the profiles of attacks are by generated empirical method for security message of previous attacks and secure state of different systems.

Fig. 1 shows a schematic diagram of developed hybrid IDS. Generally it works like intrusion detection system based on signatures, but signatures (profiles of attacks) are created automatically by means of genetic programming. GP system takes as input secure archives of events of security information system and other information systems, along with a note on the estimated level of risk in the system (for example, if the system was pre- or post-factum detected attack, the appropriate system log is sent to the server of GP system).

Thus, the proposed intrusion detection system *differs from* existing that combines performance of profile IDS and accurate of attack detection of abnormal IDS, through the use of computational intelligence to build profiles of attacks (not in real time) based on the archives of security events and their subsequent usage to detect attacks in real time.

Search of subroutine libraries for implementation of these functions was conducted.

Popular open source library Zippers (clojure.zip), which includes the implementation of tree operations made fully in functional style. As a library for the use of symbolic algebra it was enough operational features of Clojure language, which representing expressions in the form of symbols and the compilation of these expressions in the program.

Search of existing libraries was conducted to realize the directly generalized genetic operations. All founded Java-libraries were closed, so genetic operations were written manually for the prototype program.

Software package ECJ (Evolutionary Computing in Java) was also used for the second part of the testing genetic programming systems [10].

Apache Storm platform was used for implementation of the processing security events from different sources of information exchange in real time [11]. Apache Storm is a free and open source distributed computing system in real time.

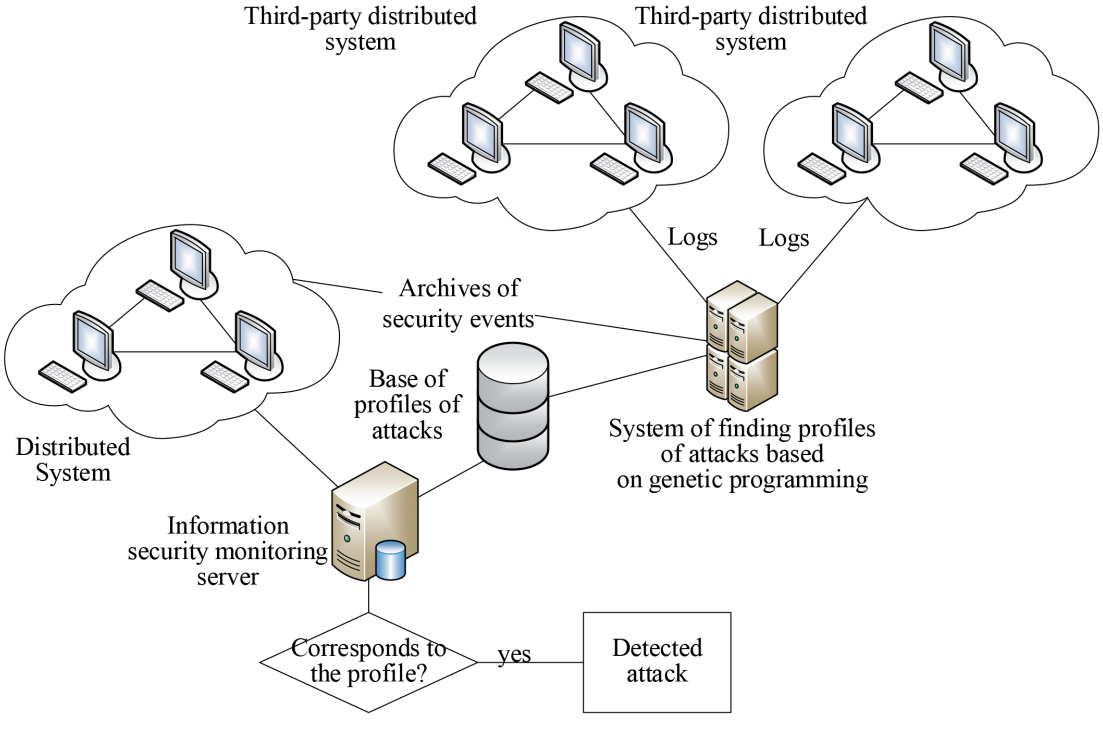

Fig. 1. Scheme of developed hybrid IDS

Unlike standard IDS types, proposed intrusion detection system allows evaluating and detecting attacks that have not been explored or identified, but their effects have been found. According to submitted for entry archive of security events (log of events) GP system is able to find the correlation of certain events and messages that are present in the logs at the time of the attack, and absent in a secure condition of the system.

Based on the scheme of the system, it can determine that the complete subroutine libraries should be used for the next operations:

— Processing tree data structures (creation, tree traversal, choose of random elements).

— The use of symbolic algebra (representing expressions in the form of symbols of the programming language).

Storm allows easy and reliable process endless streams of data and is Hadoop analogue for data processing in real time. Storm has many options for application: real time analytics, machine learning, continuous computing, DCS and other. Storm is fast platform: it provides more than one million processed data items per second on a single node. It is a scalable, fault-tolerant, ensures that the data will be processed, and easily configured and managed.

6. Research results and their discussion

6.1. Description of the experimental environment. Simulation model of secure information system was developed for comparative analysis. It generates events occurring in the system. These events are the typed logs with a direct

access of the intrusion detection system to the fields. Thus, the step of analysis (parsing) of messages in the system was missed in an experimental environment.

6.2. Experiments. There were three experiments in creation of attack profile using genetic programming and using the resulting profile in intrusion detection system in real time. Appropriate generators of messages were created for each of experiments in the system. They were matched the type of attack. Below there are expanded information on each of the experiments.

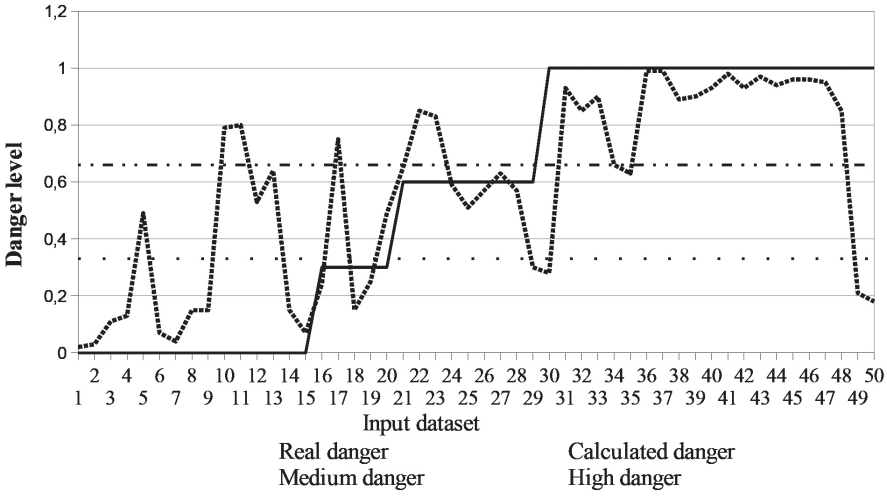

Fig. 3. Output and calculated danger value for different input data (experiment 1)

6.3. Attempt to log in the system as root user. 50 sets of input data for genetic programming system were generated for this experiment. Each set is consistent of presence or absence of intrusion attempts, respectively, each set contained a number of unsuccessful attempts to enter the system as user «root». Based on input data, genetic programming system was built attack profile, which was the formula by which the level of danger was determined. Fig. 2 shows a simplified example of obtained profile.

The following results were obtained from amount of generations to finding a potential solution at every run of genetic programming system: 62 runs were able to find a solution with an error of less than 5 % in less than 50 generations. In 16 runs, the system was unable to get even a rough attack profile.

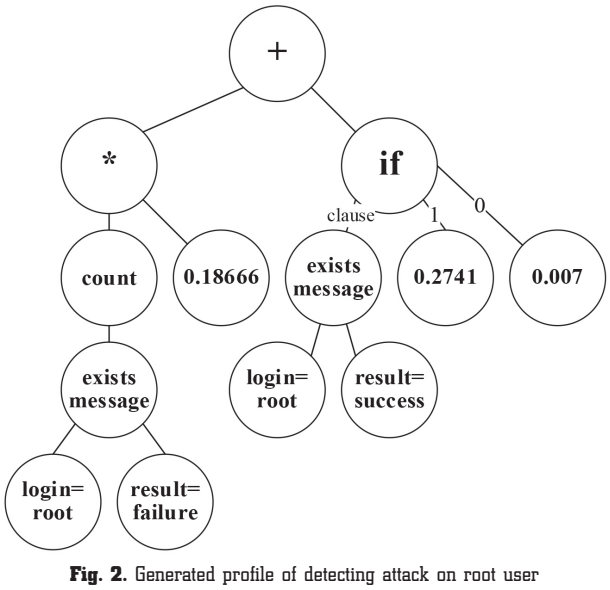

Fig. 2. Generated profile of detecting attack on root user

After receiving attack profiles, one of them was tested on another set of input data (generated again, which was not involved in the study). Fig. 3 shows a comparison of a real threat in the dataset (by setting the generator) and calculated threat. The threat of intrusion changes from 0 to 1 and is divided into three categories — low risk, medium risk and high risk.

It can be see that in 35 of 50 cases the attack profile correctly identified the level of danger for input data; in 6 cases profile did not reveal a possible attack where it was (error of the first kind); and in 9 cases the attack was more likely found where it was not (error of the second kind).

6.4. Malicious use of «Heartbleed» error. In April 2014 it was discovered error (bug) in the code of OpenSSL library, which is used by 66 % of servers on the Internet to support encryption protocols SSL/TLS. This vulnerability allows obtaining valuable information from the server, including logs of recent events, user private keys and other data that could accidentally find themselves on the same page memory area that OpenSSL library [12].

This experiment aims to investigate whether able the developed intrusion detection system to determine the pattern of input data, which leads to compromising OpenSSL server. The function, which returns the size of the message, was added in the set of terminal operators. Fig. 4 shows a greatly simplified obtained profile. Profile has a look that truly reflects the nature of the vulnerability.

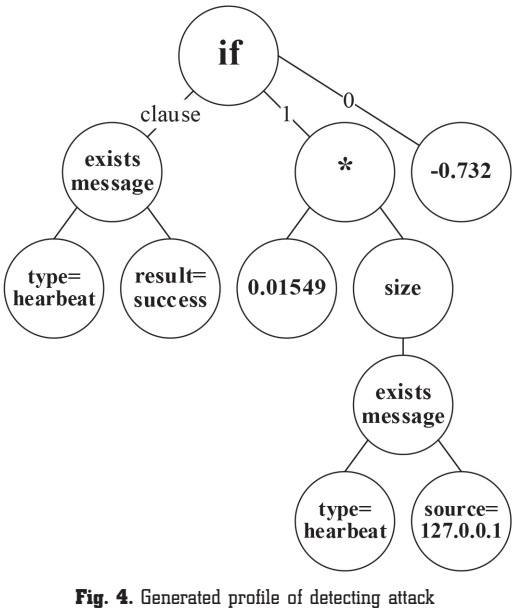

Fig. 4. Generated profile of detecting attack on «Heartbleed» vulnerability

Taking into account the number of generations to finding a potential solution at every run of genetic programming system, the following results were obtained: only in 30 cases, the system was able to find a possible solution, and only three were spent on it at least 50 generations. Start of genetic programming was automatically stopped when reaching the mark of 200 generations.

Checking the best profile that was obtained after 52-th run is shown in Fig. 5. Despite the difficulty of obtaining the profile in the previous step, it was relatively accurate in dealing with no training data. According to the graph, in 25 of 50 cases the attack profile correctly identified the level of risk; in 23 cases profile made a error of the first kind; and in 2 cases — an error of the second kind. Such high ratio of errors of the first kind to the second kind is possible explain that the resulting profile as a whole does not hold high values of danger. Since the error of the first kind, in practice, more important, this can be attributed to a significant disadvantage of the generated attack profile.

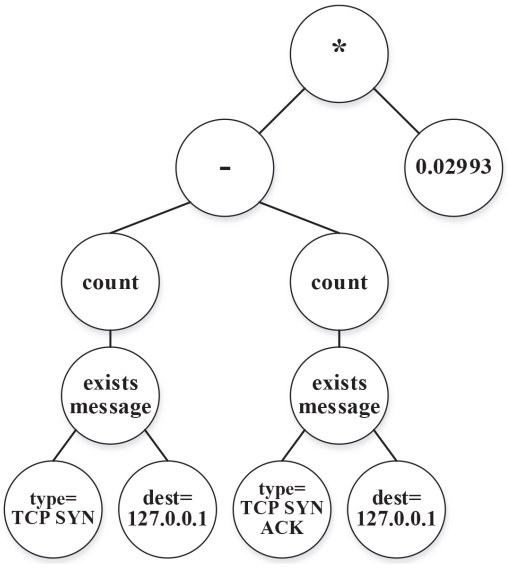

Fig. 6. Generated profile of detecting attacks by port scanning

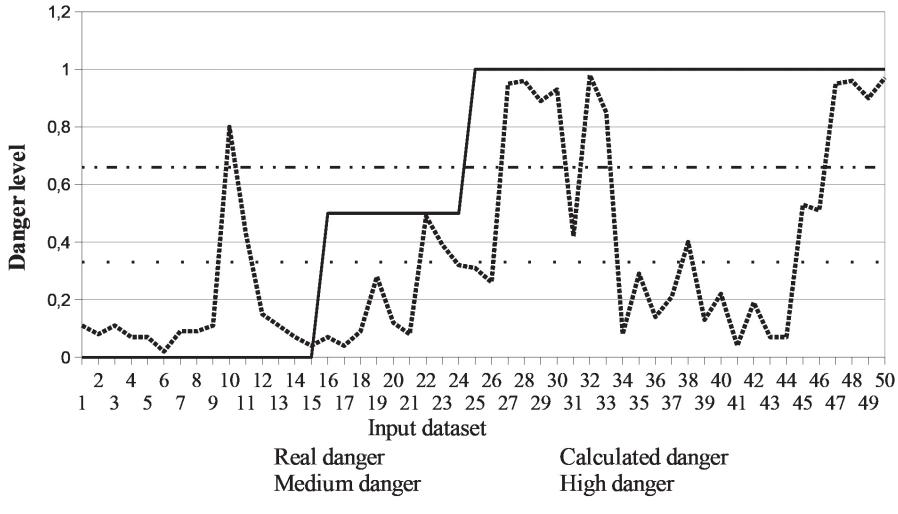

Fig. 5. Output and calculated danger values at different input data (experiment 2)

This is a good result, given that the profile in Fig. 6 often met in the results of the GP system.

Fig. 7 shows the results of one of the obtained profiles for new data that did not participate in the learning system.

The graph clearly noticeable that profile often works on data sets that do not attempt to attack. This high number of errors of the second kind can be explained by the fact that port scanning is not necessarily an attribute or signs or intrusion into the system. Since the generated profile isn't found relationship between the number of TCP SYN-messages from one address and the probability of attack (i. e., profile calculated the total number of SYN-packets from all addresses), over the test of server ports by legitimate client applications was deemed as the attack.

6.5. Server port scanning. Data set was automatically generated to emulate this scenario that emulates the output of utility *tcpdump*, i. e. log of TCP-packets. Scenario was considered for case when potential attacker use TCP SYN-packets to check which ports are open on the server for further intrusion. This is a particular problem in that port scanning is also engaged in legitimate programs to determine which services are available on the selected host.

Fig. 6 shows a simplified profile obtained by genetic programming of the system. The following results were obtained for amount of spent generations to finding a potential solution at every start of genetic programming system: in 79 cases, the system was able to find a possible solution, 45 of these solutions was conducted over 50 generations or less.

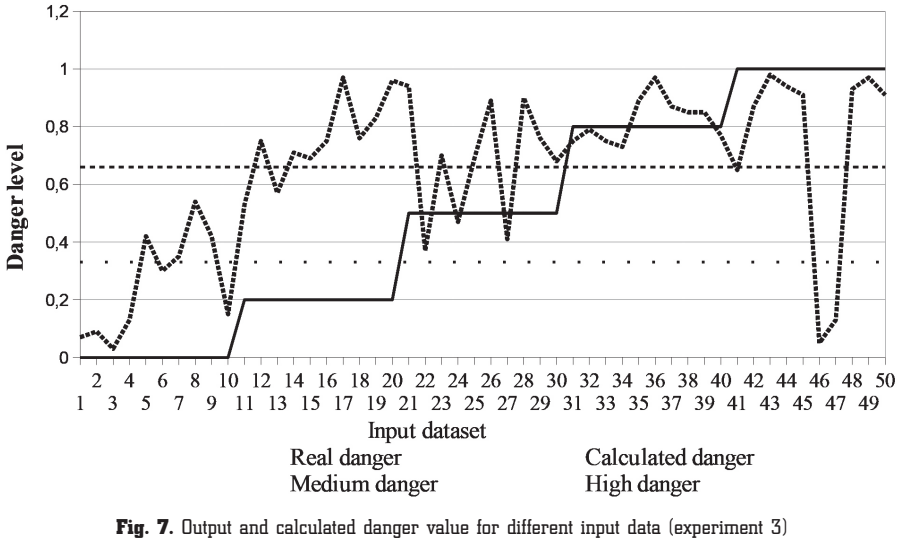

Fig. 7. Output and calculated danger value for different input data (experiment 3)

Thus, according to the results of testing IDS profile correctly identified the level of risk in 28 of 50 cases, committed 3 errors of the first kind and 19 errors of the second kind.

Comparison of experimental results for three types of attacks on distributed system is given in Table 2.

Table 2

Comparative table of IDS simulation results

Experiment	KI	KII	S50	S200
Attempt to login in the system as root	0,2	0,3	0,62	0,84
OpenSSL vulnerability «Heartbleed»	0,657	0,057	0,03	0,32
Server port scanning	0,1	0,633	0,45	0,79

As shown in the Table 2 developed IDS model allows:
— Accurately identify potential conventional attacks.
— Quite firmly, but with a lot of errors of the second kind, detect not obvious attacks that are often similar to normal operation of the system and its users.
— With mediocre reliability and complexity of obtaining profiles detect new types of attacks and vulnerabilities, which is unable to carry out any of the existing IDS at the profiles of attacks (as opposed to IDS based on anomalies).

The practical value of the results determined by their focus on creating effective tools that support the security management of DCS resources at a constant amount of resources that can increase the security level of the computer network and the effectiveness of security controls, as well as the creation of specialized environments for research and analysis of DCS security mechanisms. The methods can be used to monitor distributed computing systems that provide the gas transmission system, power supply systems.

This article is a continuation of the authors' works in the development of methods and tools for intrusion detection in distributed computing systems using artificial intelligence [13]. In future studies the authors plan to develop mechanism for detection of distributed attacks using mobile security agents to find the best options for placing components of information security.

7. Conclusions

As a result of research:

1. Existing approaches and basic features, principles and mechanisms of security monitoring of distributed systems were analyzed. It was shown that the existing monitoring system of DCS security not guarantee detection of all attacks and the lack of false positives.

2. These metrics intrusion detection system: error of intrusion detection and the number of start for method of genetic programming that lead to the correct result. A hybrid detection system was proposed based on the use of attack profiles that generated by empirical method based on previous attacks and secure state of the system.

3. The model of proposed intrusion detection system was developed that allows evaluating and detecting attacks that have not been explored or identified, but their effects have been found.

4. The experimental prototype research intrusion detection system based on genetic programming was done. We consider three scenarios that are fundamentally different from each other by different characteristics. Each of them was held two-step full cycle of the system — identification of attack profile and checking the profile for new input data. It was established that the developed model can detect: with high precision — traditional potential attacks, with many errors of the second kind — not obvious attacks, with the mediocre reliability and complexity of obtaining profile — new types of attacks and vulnerabilities.

References

1. Barman, S. Writing Information Security Policies [Text]: Translation from English / S. Barman. — Moscow: Publishing House «Williams», 2002. — 208 p.
2. Gubenkov, A. A. Informatsionnaia besopasnost' [Text] / A. A. Gubenkov. — Saratov: Novyi isdatel'skii dom, 2005. — 128 p.
3. Beale, J. Snort 2.1 Intrusion Detection [Text] / J. Beale et al. — Syngress, 2004. — 608 p. doi:10.1016/b978-193183604-3/50003-5
4. Kaspersky, K. Hacker Disassembling Uncovered: Powerful Techniques To Safeguard Your Programming [Text] / K. Kaspersky. — A-List Publishing, 2003. — 600 p.
5. Bace, R. G. Intrusion Detection [Text] / R. G. Bace. — Sams Publishing, 1999. — 368 p.
6. Roman, R. Applying intrusion detection systems to wireless sensor networks [Text] / R. Roman // Consumer Communications and Networking Conference. — 2006. — Vol. 1. — P. 640–644. doi:10.1109/ccnc.2006.1592966
7. Luke, S. Genetic programming produced competitive soccer softbot teams for robocup97 [Text] / S. Luke // Genetic Programming 1998 Conference (GP-98), July 1998. — Madison, Wisconsin, USA: University of Wisconsin, 1998. — P. 214–222.
8. Stijven, S. Separating the wheat from the chaff: on feature selection and feature importance in regression random forests and symbolic regression [Text] / S. Stijven, W. Minnebo, E. Vladislavleva // Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation — GECCO'11. — Dublin, Ireland, 2011. — P. 623–630. doi:10.1145/2001858.2002059
9. Koza, J. R. Genetic Programming IV: Routine Human-Competitive Machine Intelligence [Text] / J. R. Koza, M. A. Keane, M. J. Streeter, W. Mydlowec, J. Yu, G. Lanza. — New York, NY, USA: Springer, 2005. — 590 p. doi:10.1007/b137549
10. Luke, S. ECJ: a java-based evolutionary computation and genetic programming research system [Electronic resource] / S. Luke, L. Panait, Z. Skolicki, J. Bassett, R. Hubley, A. Chircop. — 2001. — Available at: \www/URL: http://cis-linux1.temple.edu/~pwang/3203-AI/Project/2004/Flanigan/ec/ec/
11. Sakaki, T. Earthquake shakes Twitter users: real-time event detection by social sensors [Text] / T. Sakaki, M. Okazaki, Y. Matsuo // Proceedings of the 19th international conference on World wide web (WWW2010). — Raleigh, North Carolina, ACM, 2010. — P. 851–860. doi:10.1145/1772690.1772777
12. Queal, Z. D. Necessary Implementation of Adjustable Work Factor Ciphers in Modern Cryptographic Algorithms as it Relates to HeartBleed and OpenSSL [Electronic resource] / Z. D. Queal. — Available at: \www/URL: https://gist.github.com/zQueal/3b0db5ba2532e04ad9ed
13. Volokyta, A. Obnaruzhenie vtorzhenii v raspredelennye komp'iuternye sistemy na osnove geneticheskogo programmirovaniia [Text] / A. Volokyta, Vu Duc Thinh, O. Yakushev // Visnyk Chernihivs'koho Derzhavnoho Tekhnolohichnoho Universytetu. — 2012. — № 2(57). — P. 128–134.

РАЗРАБОТКА СПОСОБА ОБНАРУЖЕНИЯ АТАК В РЕАЛЬНОМ ВРЕМЕНИ НА ОСНОВЕ ВЫЧИСЛИТЕЛЬНОГО ИНТЕЛЛЕКТА

Разработан способ обнаружения атак в реальном времени на основе вычислительного интеллекта, который отличается применением метода генетического программирования для построения профилей атак и позволяет оценивать и выявлять атаки, которые еще не были исследованы или определены, но их последствия уже были обнаружены. Было проведено экспериментальное исследование прототипа системы обнаружения атак.

ISSN 2226-3780

Ключевые слова: система мониторинга безопасности, распределенная компьютерная система, вычислительный интеллект.

Луцький Георгій Михайлович, *доктор технічних наук, професор, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут», Україна.*
Волокита Артем Миколайович, *кандидат технічних наук, доцент, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут», Україна,* ***e-mail: artem.volokita@kpi.ua****.*
Якушев Олександр Юрійович, *кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут», Україна.*
Регіда Павло Геннадійович, *аспірант, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут», Україна.*
Ву Дик Тхінь, *кандидат технічних наук, факультет інформаційних технологій, Хошимінський університет харчової промисловості, В'єтнам.*

Луцкий Георгий Михайлович, *доктор технических наук, профессор, кафедра вычислительной техники, Национальный технический университет Украины «Киевский политехнический институт», Украина.*
Волокита Артем Николаевич, *кандидат технических наук, доцент, кафедра вычислительной техники, Национальный технический университет Украины «Киевский политехнический институт», Украина.*
Якушев Александр Юрьевич, *кафедра вычислительной техники, Национальный технический университет Украины «Киевский политехнический институт», Украина.*
Регида Павел Геннадиевич, *аспирант, кафедра вычислительной техники, Национальный технический университет Украины «Киевский политехнический институт», Украина.*
Ву Дык Тхинь, *кандидат технических наук, факультет информационных технологий, Хошиминский университет пищевой промышленности, Вьетнам.*

Loutskii Heorhii, *National Technical University of Ukraine «Kyiv Polytechnic Institute», Ukraine.*
Volokyta Artem, *National Technical University of Ukraine «Kyiv Polytechnic Institute», Ukraine,* ***e-mail: artem.volokita@kpi.ua****.*
Yakushev Oleksandr, *National Technical University of Ukraine «Kyiv Polytechnic Institute», Ukraine.*
Rehida Pavlo, *National Technical University of Ukraine «Kyiv Polytechnic Institute», Ukraine.*
Vu Duc Thinh, *Ho Chi Minh City University of Food Industry, Vietnam*

УДК 621.311.001.57
DOI: 10.15587/2312-8372.2016.71973

Зубенко Д. Ю.,
Шавкун В. М.,
Скурихін В. І.,
Донець О. В.,
Лукашова Н. П.

ДОСЛІДЖЕННЯ ТА РОЗРОБКА ТЕХНОЛОГІЙ СИНТЕЗУ НЕЙРОМЕРЕЖЕВИХ АЛГОРИТМІВ БАГАТОРЕЖИМНОГО УПРАВЛІННЯ ТРАНСПОРТНИМ ПІДПРИЄМСТВОМ

Розглядається проблема проектування інтелектуальних систем управління (ІСУ) динамічно-змінними об'єктами (ДО), що функціонують в умовах суттєвої апріорної невизначеності. Представлено аналіз існуючих підходів до побудови ІСУ ДО, методів, моделей і алгоритмів їх побудови на основі інтеграції класичних методів теорії управління і методів штучного інтелекту. В якості прикладів ДО розглядаються рухомий склад багаторежимних підприємств (ТП).

Ключові слова: *інтелектуальні системи, динамічно-змінні об'єкти, транспортні підприємства, нейромережеві алгоритми, нейронні мережі.*

1. Вступ

Задача синтезу нелінійного управління ТП (транспортним підприємством) є опис процесів в ТП в різних станах. Однак, як вже зазначалося раніше [1] вплив факторів невизначеності (зміна умов оптимізації та режимів роботи ТП) значно впливає на процес зміни параметрів моделі структури, внаслідок чого виникає завдання адаптації параметрів САК (Система Автоматичного Керування) для підтримки необхідної якості процесів управління. На практиці найбільш часто використовуються два методи вирішення цього завдання: синтез адаптивного регулятора з фіксованою структурою і параметрами алгоритму адаптації, і синтез САК в класі систем з навчанням, тобто систем, які в процесі свого функціонування можуть змінювати свою структуру і параметри для досягнення необхідної мети управління. Актуальністю роботи в даному напрямку є те що в постійному розвитку промисловості і транспортних підприємств у світі в умовах жорсткої економіки і невизначеності подальшого розвитку кон'юнктури ринку необхідно розробляти нові підходи щодо управління підприємством на основі нейронних мереж.

2. Аналіз літературних даних та постановка проблеми

Дослідження в галузі оцінки та оптимізації складності САК динамічними об'єктами мають півстолітню історію. Разом з тим, указані вище підходи, визначаючи необхідні напрямки досліджень стосовно до САК динамічно-змінними об'єктами, не вказують формальних алгоритмів та методик синтезу ІСУ ТП на основі критерію мінімальної складності при виконанні заданих вимог до якості процесів управління в умовах невизначеності режимів роботи ТП і зміни зовнішнього середовища. В останніх публікаціях та дослідженнях відображається ця проблема, так у [1]

