UDC 004.056

Anna Verner, Valerii Simonenko.

OBJECT-ORIENTED METHOD FOR ENHANCING COMPUTING NODE SECURITY BY MEANS OF OPERATING SYSTEM AUDIT SUBSYSTEM

The article considers an object-oriented way of enhancing the security of a computer node based on the operating system audit subsystem. To increase security, it is proposed to use the audit subsystem as a means of monitoring the processes and system calls they make in the system. In order to analyze the maliciousness of the process being monitored a classifier is used implemented using a neural network represented as a multilayer Rosenblatt perceptron. Training of the model is carried out by using a data set consisting of system calls received as a result of malicious software.

Keywords: Security enhancement, operating system audit subsystem, malicious software, neural networks.

Target setting. With the active development of information technology, there is a continuous exponential increase in the total number of malicious software aimed at the failure of computing nodes. Thus, the problem is the invention of a method for improving the security of computing nodes.

Actual scientific researches and issues analysis. In recent years, there appeared an increasing number of papers dedicated to improving the safety of computing nodes, due to the emergence of many new approaches to malware detection. However, there are no approaches to improving the security of computing nodes based on the use of integrated means of operating systems.

Uninvestigated parts of general matters defining. This article is devoted to the study and analysis of the proposed method of improving security, based on the use of the operating system audit subsystem. The research focuses on the study of the application of audit subsystem and neural network classifier.

The research objective. The objective is to develop a method for increasing the security of computing nodes, by which malicious activity can be detected through dynamic analysis during system operation.

The statement of basic materials. According to open statistics [4], the curve of the number of new types of malicious software applications is declining. Security software developers report [1] that more than seventy-five percent of infected nodes have had the latest security updates installed. This indicates the imperfection of existing remedies, which makes it impossible to detect and prevent existing threats promptly.

This suggests the question of the necessity to strengthen existing information security tools by inventing new ways to identify threats. In this regard, the need to strengthen existing information systems security means by inventing new ways to identify threats is quite acute.

In today's information systems, to ensure data integrity, hardware, and software data protection means are mostly used. The latter is most common due to the flexibility in usage and the ability to configure for different types of architectures without the necessity to make significant changes in their structure.

The basic software means of protection of the information are presented in the form of the following means: the anti-virus software, identification, and authentication of users, logging, management of access and means of an audit. Application of such specialized software tools of information protection assumes revealing of threats by use of some methods regarding the analysis of a code of harmful applications . Two main approaches should be singled out among them: a static and dynamic analysis.

During static analysis, no malicious software is executed. Evidence of application danger is obtained by analyzing metadata of files, line signatures, n-grams, calls to libraries etc. This approach is rather effective due to its speed, but it does not allow you to identify threats in a proper fashion as it is not resistant to obfuscation, which actually caused the necessity of dynamic analysis.

Dynamic analysis involves analyzing software directly during its execution in the system, by observing the interaction of the collected sample of a malicious application with the information system. Compared to static analysis, this approach is more efficient and allows detecting a significant portion of malware. However, it also has drawbacks, one of which is the high time and resource costs, as a special isolated virtual environment is created to monitor the malicious application sample. To obtain better results, developers combine the advantages of these approaches by using hybrid analysis.

Recent research in the field of security tools development [5], [6], [2] shows that more and more attention is paid to the use of machine learning (ML) in improving analysis methods. The application of ML allows to reduce considerably the percentage of false positives about software security and also proves [5] the possibility to prevent zero-day attacks. To increase the safety of computing nodes it is suggested to enhance the analysis methods in the following way based on the application of the audit subsystem of operating systems.

The audit is defined as the fixation by specialized software means of protection of what is being done in the system for the purpose of further analysis of such information. Audit data is collected on separate host systems from logs, which are formed in the process of work. The received data contain a considerable quantity of useful information allowing not only to trace events occurring in the system in detail,

but also to restore a course of events in case of necessity of studying of attacks specificity. The audit subsystem itself has only diagnostic properties, but due to the integration of the subsystem to the kernel modules and its work at a low level with the possibility of intercepting kernel system calls, it is proposed to use it as a basic component for analysis.

The audit subsystem controls three main types of events: system calls, allowing viewing their contextual information; file access events as an alternative way to monitor file access activity [3]; and patterns previously configured to capture the event. Receiving system calls from the user space of the audit subsystem core component performs filtering. After passing through one of the input filters, the call is directed to pass through the exception filter, based on the configuration of the audit rules for further processing directly to the audit daemon. A sophisticated subsystem of system call filters allows you to control the behavior of any process in the system and, in case of coincidence with the malware behavior pattern defined by the behavioral model-based machine learning algorithms, prevent further process execution in the system.

During the detection of threats based on behavior it is necessary to primarily determine the behavior, and then create a dataset, next it is possible to identify distinct features from the data set and to classify them using the algorithms of machine learning.

The mechanism of the behavioral analyzer is based on the application of machine learning methods, which allows identifying the features needed to classify objects in accordance with the statistical structure of input data. To solve the problem of system calls determination the approach, which is widely used in text classification, has been chosen. In this way, it is possible to prevent the effects of the data order and to determine the sample harmfulness according to the existing context. Despite the change in code, behavior remains similar, thus allowing this approach to be used to determine most types of new malicious programs.

Experiments. In order to test the possibility of implementing an object-oriented method of enhancing the security of the computing node the usage of operating system audit subsystem modeling was performed. The results of the model training are shown in Fig. 1.

The graph obtained during the modeling demonstrates the correctness of the proposed method, as according to the results of the test set, the accuracy of malware recognition increases from 70%. The final check on the model test example shows the correctness of the conducted research.

Conclusions. The presence of ultra-high amounts of malware leads to daily losses not only for large corporations but also for society as a whole. Due to constant malware improvement, there is a need to complicate algorithms and analysis methods in order to protect information systems not only from existing ones but also from zero-

day threats. The proposed method can be used to prevent the execution and spread of malicious applications based on any operating system, as the audit subsystem is a native diagnostic tool. The main advantage of the developed method is not only the ability to identify security threats to the information system, but also the ability to prevent its execution and distribution due to the presence of high privileges.

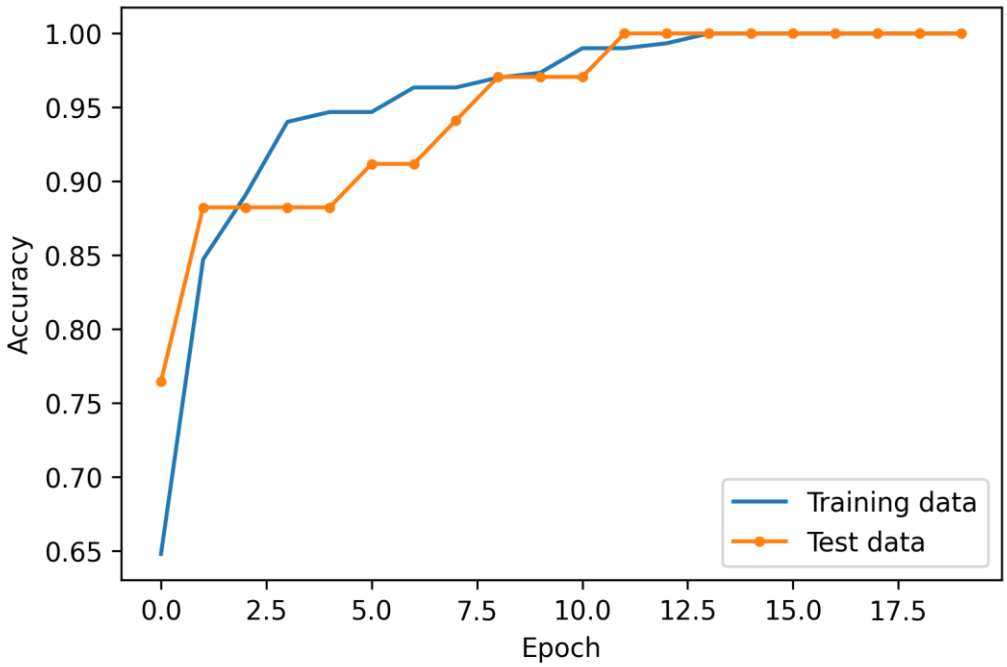

Fig. 1. Classifier learning graph

The developed method has its advantages and disadvantages. The advantage of the given method is the usage of the built-in tool in the form of a subsystem of audit as the system on interception and monitoring of events due to which it is not necessary to develop the additional mechanism of interaction with the system calls. The main disadvantage is the increase in the load on the productivity of the system. Therefore, further research is needed to eliminate this disadvantage and optimize the protection system operation.

References

1. Businesses Impacted by Repeated Ransomware Attacks and Failing to Close the Gap on Exploits, According to Sophos Global Survey [Електронний ресурс] // Sophos. URL: https://www.sophos.com/en-us/press-office/press-releases/2018/ 01/businesses-impacted-by-repeated-ransomware-attacks-according-to-sophos-global-survey.aspx (дата звернення: 20.03.2020).

2. J. W. Stokes, D. Wang, M. Marinescu, M. Marino and B. Bussone, "Attack and Defense of Dynamic Analysis-Based, Adversarial Neural Malware Detection

Models, MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM), Los Angeles, CA, 2018, pp. 1-8.

3. Kelly Shortridge, What is the Linux Auditing System [Електронний ресурс] // Capsule8. January 7, 2020/ URL: https://capsule8.com/blog/auditd-what-is-the-linux-auditing-system/ (дата звернення: 20.03.2020).

4. Lapowsky, “Malware last 10 years” [Електронний ресурс] // AV-TEST. URL: https://www.av-test.org/en/statistics/malware/ (дата звернення: 20.03.2020).

5. Liu, Xinbo & Lin, Yaping & Li, He & Zhang, Jiliang. A Novel Method for Malware Detection on ML-based Visualization Technique. Computers & Security. 89. 101682. 10.1016/j.cose.2019.101682. (2019)

6. R. Agrawal, J. W. Stokes, M. Marinescu and K. Selvaraj, "Neural Sequential Malware Detection with Parameters," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, 2018, pp. 2656-2660.

AUTHORS

Anna Verner - student, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: annverner7@gmail.com

Valerii Simonenko (supervisor) - professor, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: svp@comsys.kpi.ua