Vol. 05, No. 2 (2023) 227-238, doi: 10.24874/PES05.02.005

Proceedings on Engineering Sciences

www.pesjournal.net

ALGORITHM FOR THE DEVELOPMENT OF INFORMATION REPOSITORIES FOR STORING CONFIDENTIAL INFORMATION

Zohid A. Hakimov[1]
Asilbek Medatov
Viktor Kotetunov
Yuriy Kravtsov
Alisher Abdullaev

Received 10.11.2022.
Accepted 31.01.2023.
UDC – 004.056.5

Keywords:

Databases, Unauthorised Access, Internet Security, Network Processes, Virus Software

A B S T R A C T

Due to the intensive use of the Internet, network security is becoming a key foundation for all web applications. Intrusion detection by analysing records in network processes is an important way to solve problems in the field of network security. It has been identified that an intrusion can threaten not only the integrity of the data but also the system itself. With the development of information technology and an increase in data transfer speeds, there are threats of incorrect use of the Internet. The authors determine that more reliable control systems are needed that solve the problem of network protection without human intervention. In a number of sources, attention is focused on the possibility of autonomous detection of the vulnerability of programmes and protocols by analysing the criteria for the behaviour of the system itself. Many models are built on informal methods, such as signature ones, in which it is difficult to obtain a correct assessment of effectiveness and completeness. The authors start from the fact that the attack is characterised by states and transitions. The possibility of using neural networks has been tested. A distinctive feature of a neural network is that they start working only after the learning process. This is one of the main advantages of a neural network over conventional algorithms. The paper shows that the development of information storages is possible provided that an equilibrium state is reached when the system does not allow expanding the attack space and the information storage is available for both external access and remote disconnection. The learning model consists of arrays of data with a distributed storage environment. This is the main component of improving the performance of the intrusion detection system. The experimental results obtained showed that the proposed approach identifies anomalies more effectively than known methods. The paper is devoted to the development of a method for detecting attacks based on information about the behaviour of deviating values in the network.

[1] Corresponding author: Zohid A. Hakimov
Email: zohid.hakimov@outlook.com

1. INTRODUCTION

One of the first works in this field was a work from the USA, which defined the basic concepts and solutions to problems. The first works were rather conceptual – there was an attempt not to build instrumental filters or methods but to try to apply probability theory to solve these problems. A number of sources describe the attack from the intruder's standpoint and are based on the concepts of the invasion target. The use of anomaly detection and attack detection tools is hampered by the areas of destination. The narrower the scope of application, the easier it is to apply certain research tools to it since it is easier to choose the appropriate model of network objects' behaviour (Song et al., 2015). Learning is based on connections between neurons that determine the ratio of input and output signals of a neuron. The neural network is based on the "level of training" and does not allow analytical calculation of errors. The disadvantages include the fact that the network topology and the location of nodes are determined only after a sufficiently large number of trials and errors. The main disadvantages of the neural network are inefficiency in User to Root (U2R) and right-to-left (R2L) intrusions and low reliability (Xiaoru & Ling, 2021). To solve these problems, a new approach to the deviation method based on intrusion detection is proposed. The detection of deviations is carried out to increase the stability of intrusion detection. This approach consists of two stages: training with working datasets and testing with datasets with intrusion patterns. Such data is used to prepare intrusion detection at the initial stage of implementation. Normal data sets increase the performance of intrusion prevention. If the number of errors exceeds the threshold value, the tested data set will be characterised by the system as an unauthorised action. Various methods can be used to detect intrusion but each of them is specific to a particular method. The main purpose of the intrusion detection system is to detect attacks effectively. It is important to identify the attack at the initial stage to reduce its negative consequences. In this paper, an approach of deviating values is proposed, in which the anomaly is measured by deviation factors (Gaoyu et al., 2019).

In recent years, not only the complexity of software products has increased but also the threat from virus software (Lavrov et al., 2017). It is precisely such software elements that are very popular in the black markets and are currently evolving very rapidly. They enable corporations and sometimes countries, to cause great harm to their economic and political opponents (Zhang, 2021). The virus software market is constantly growing and developing (Son et al., 2008). From small programme blocks embedded in executable files of other programmes to complex independent multi-level systems consisting of a large number of components that have different goals and objectives: installers, loaders, masking programmes, etc. The main medium of distribution of such software in the modern world is the Internet. There are a lot of intrusion detection methods but most of them are either impossible to apply in practice or are so cumbersome that they considerably reduce the performance of the user's system or the network itself. Therefore, the question of the relevance of these developments lies in the concretisation and modernisation of existing methods that theoretically fulfill the set goals but in practice they are difficult to implement (Wang et al., 2021). The use of anomaly detection and attack detection tools is complicated by the areas of destination. The more specific the speciality is, the easier it is to apply certain tools to it. Most often, certain methods come from the specific features themselves, which can "cover" all the weaknesses in ensuring the operability of the system (Dong et al., 2019).

The possibility of working with neural networks was also considered but a huge disadvantage when working with them was the difficulty of verifying the results of the study of training samples (Liu & Zhang, 2020). The adaptive method is suitable for working with network elements. It is characterised by the fact that even with low computational complexity it will have a low level of false messages (Akritidis et al., 2020). Since this method has high performance and minimal computing power, it is suitable not only for working with databases (for which it was originally developed) but also for describing the behaviour of network elements. This method will provide high reliability when detecting attacks or unrecorded actions (Sivapriya & Kartheeban, 2018). Adaptability, in turn, will help to attract a solution to the problem in one system for a number of others. It is the problem of adaptability that is critical for most ready solutions. By combining this method, it is possible to preserve and improve the very formula of attacks using network elements (Aksvonov & Antonova, 2018). The relevance of these works is to obtain a product that can be used for any systems (Rui et al., 2020).

2. MATERIALS AND METHODS

Given the growing threats of unauthorised access to user or company data, enterprises must constantly spend huge amounts of money to protect their data. Small enterprises are under threat, as well as large companies and holdings (Sivaram et al., 2021). Security for small businesses can depend on many aspects, such as:

- defining security policies and procedures;
- IT investment solutions;
- personnel security issues;
- data security issue;
- network security;
- virus protection;
- intrusion detection policy;
- using access cards;
- backup procedures and disaster recovery plans.

Unauthorised access is often referred to as a weakness in a controlled system where control elements cease to be effective. In addition, unauthorised access to the system can be opened due to an error in the software, which can be used by attackers to gain access to the system or network. Detection of vulnerabilities and unauthorised actions are essential to improve the security of the network. The operating system is exposed to unauthorised access only when there are all the prerequisites for running the application, not part of it. Such applications can be various kinds of editors (photo, video), as well as numerous software complexes for performing various tasks (players, engineering applications, navigation centres) (Lavrov et al., 2015). Both the system and all applications that are compatible with this system can be subjected to unauthorised access. The reason for the occurrence of such illegal actions and abuses is the fulfilment of the following conditions:

- wide application of this system;
- open core access (opensourse systems);
- insufficient protection.

All these conditions are necessary to gain unauthorised access to the network, programme, or operating system. The condition of popularity is necessary for the attack to make sense and its impact to have visible consequences. It makes no sense to hack or gain access to networks that several people know about. If the system has only a couple of copies and is used by a narrow circle of specialists, then its attractiveness for hacking decreases. Another condition is the widespread distribution of the system, which affects its attractiveness for hackers (Jeong et al., 2008). System security is based on architectural or software solutions that prevent unknown functions from accessing user files and vital parts of the system. Security blocks unauthorised activity but therewith imposes certain restrictions on the capabilities of some software elements of the user (Tian, 2021). Most of the methods and software solutions available on the market cannot guarantee a consistently high level of protection of the system, local and global networks. The problem lies in the rapid growth of new threats. The main danger of such growth is that with intensive flows of intrusions, it is almost impossible to ensure their detection by 100%.

The reasons for this increase in threats are the following factors. Mostly, the threats are created to defeat computers on the global network and the volume of writing viruses is growing every day. At such a pace of development of tools for gaining access to network components, it is impossible to write signature database updates to antivirus companies on time and considering the number and variety of threats (Jain et al., 2013). The rapid growth of threats implies that the vast majority of computers will be affected even before the release of new virus signatures. Antivirus companies need to constantly release signature updates to compete with each other and this reduces the time for analysing malicious code and negatively affects the quality of the final product. Employees simply do not have enough time for qualitative analysis of intrusion codes (Shokrzadeh et al., 2015). Neutralising malicious code is also not an easy task. Since the writing of viruses stands on a commercial basis, technologies using methods of hiding from antiviruses based on the vulnerabilities found are being developed and applied. These technologies complicate the task of recognising and removing intrusions. Threats can encapsulate self-defence procedures to prevent deletion by deactivating system utilities for registry access or process control. It also uses code that monitors the integrity of threat files and the keys necessary for operation in system registries. There are acute problems of efficient consumption of system resources. To track traffic in real time, antiviruses must have modules for working with system events, which will allow filtering streams and making threats that can compromise protection impossible. Most system events and the frequency of their occurrence can greatly slow down the work when elaborating on threats. There are problems of incompatibility of antiviruses. Often, due to conflicts of system event interception routines, it is impossible to work with different antiviruses at the same time. There is a need for new methods of combating threats that will be based on behaviour analysis and will be able to bypass encryption. Such approaches should effectively combat old and new modifications of viruses, while maintaining high performance and minimally loading the system. It is also important to teach the system to autonomously detect unauthorised actions without accessing databases.

3. RESULTS AND DISCUSSION

Modern malicious software contains a wide range of viruses that harm not only the infected system but sometimes the entire local or global network. Viruses are divided into classes with common characteristics: habitat; algorithms of operation, and destructive power. The habitat is divided into: file; boot; macro viruses, and network. File viruses are mounted in files. Boot ones are hidden in boot sectors on the hard disc. Macro viruses infect documents and spreadsheets of text editors. Network viruses spread through mail or networks. Viruses can be combined to complicate their detection. Combining masks, the presence of viruses in the system while attackers destroy it or steal user data. Examples of such combinations are file-boot and network macro viruses. They have a complex algorithm of operation and use stealth and polymorphic technologies to get into the system. Virus algorithms characterise:

1) residency;
2) using stealth algorithms;
3) self-encryption and polymorphism;
4) the use of non-standard measures.

The resident virus encapsulates its part in RAM, which then intercepts the operating system's access to the infected objects and is written in them. Resident viruses are stored in memory and are active until the computer is turned off or the operating system (OS) is restarted. Stealth algorithms hide viruses in the system. The most common stealth algorithm is the interception of OS requests to read/write infected objects. Stealth viruses therewith "substitute" unaffected areas of information instead of themselves. In the case of macro viruses, this is a ban on calls to the macro viewing menu (Liu et al., 2020). Self-encryption and polymorphism are used by all types of viruses to complicate the virus detection procedure as much as possible. Polymorphic viruses have no signatures and do not contain a single permanent piece of code. In most cases, two samples of the same polymorphic virus will not have a match. This is achieved by encrypting the main body of the virus and modifying the decrypting programme. According to destructive capabilities, threats are divided into:

1) harmless viruses – do not affect the operation of the computer in any way, except for the reduction of free memory on the disc as a result of their distribution;
2) safe viruses – the impact of viruses is limited by the reduction of free disc memory and graphic, sound, and other effects;
3) dangerous viruses – can lead to serious computer failures;
4) very dangerous viruses can lead to programme loss, data destruction, erasure of information necessary for computer operation, which is recorded in system memory areas and even contribute to accelerated wear of moving parts of mechanisms, such as hard disc drive heads.

The analysis shows that it is necessary to develop methods and models for recognising both old and new modifications. Signatures allow detecting already known viruses. A signature is a set of features that can be used to characterise an object. The signs allow briefly describing huge objects. Hash functions that characterise an object using short features also function on this principle. Signs of file type, date, address, and size are called "weak signatures". A signature is a unique sign of an intrusion, through which a fragment containing it can be attributed to threats. If the intrusion is not unique, the signatures that describe it will be of the same type and will be a sequence of consecutive bytes and addresses in the file of this sequence. If the file size is known, this will be an additional trigger for the reliability of threat detection. The more information there is about the attack, the more accurately it can be characterised. Different signatures are used for different types of intrusions. The strongest signature includes an unchanged part of the virus (if it is polymorphic), which considerably increases the size of the signature. Fragmentation is used to minimise the size and length of signatures. Fragmentation allows using intermittent signatures that have two parts: common (characterising the entire type of intrusion) and unique (manually modified).

There is also a method of "halved" signatures, which allows using parts of different signatures when identifying polymorphic threats. The method operates with bit fields and may not be used with all signatures (everything will depend on their type). The difficulty of recognising polymorphic intrusions is that they remain unchanged after decoding the body into memory. The difficulty consists in the determination of the decryption time, which is the timer for the start of the intrusion. Halved signatures can detect even such intrusions by decrypting the executing fragment and already executed one. The disadvantage of the method is the size of the signatures, so control codes are used. They are formed from the virus code and are unique. There may be several intrusions with the same control codes (collisions) but this will not affect their detection. Control codes can replace the hash functions Secure Hash Algorithm (SHA) and Message Digest 5 (MD5), despite the complexity of their application they are very compact and fit into one word (32 or 64 bytes) and eliminate the possibility of collisions. When using hash functions, the following fields are available in the antivirus database: offset, length, and hash of the file. When working with a file, the antivirus checks the hash of the fragment in the database with the specified offset and compares it with the reference value. The values are equal, which means the fragment will be the one of interest. This method is not inferior in accuracy to signature methods and has high performance and minimal requirements for system resources. The main advantage of signature methods is the accurate detection of the type of virus. This feature allows adding both signatures and methods of blocking threats to the database. Disadvantages of the signature method: threat samples are needed; the need for updates; manual threat analysis in case of collisions; identifies only known threats.

The main disadvantage of the method is minimal autonomy and dependence on updates. This method is the best from the standpoint of monetisation – the user should always pay for the possibility of updating his system. The main purpose of heuristic analysis is to track unknown and new modifications of unauthorised actions. It accepts and examines programme files, and based on the results of the work, a conclusion is made about the presence of intrusions. To get the correct result, the following steps should be performed:

1. Semantic analysis It allows recognising and converting the executed commands into an operable form. After that, these commands are analysed to find sequences in the code of programmes that implement dangerous actions.

2. Interpretation. This step helps to find polymorphic programmes (when the action begins to perform an

intrusion not immediately but at the end of a certain predetermined time or cycle). This step requires launching the application. A command stream is used for detection, which can have negative consequences for user information, so executing this code on a computer is not desirable. To do this, the emulator simulates hardware and software functions that record the activity of the executable code.

3. Pragmatic analysis. Allows determining the purpose of the attack algorithm based on the content of the teams and their groups.

Semantic analysis The programme affects many factors (values of registers, processor flags, memory areas). Most of these parameters will not be considered when detecting intrusions. When detecting, discrete models of viruses are used, and not every programme action acquires the status of "events", which are programme actions related to system calls that lead to changes in the system. Semantic analysis searches for and recognises sequences of commands in the disassembler listing that implement events belonging to "discrete" models. Recognition occurs as follows: a set of any elements (bits, bytes, words, or their sequences) that can be represented as an "alphabet", and these elements themselves are "letters of the alphabet". Combining elements and making different sequences out of them, different "phrases" are acquired. Many phrases are described and will be the "language". For reference types of intrusions, finite automata are created to recognise a sequence of assembler commands that implement their behaviour. The programme is first disassembled and then recognised by automata. Based on the number of recognised fragments and their functionality, the analyser makes a verdict on the harmfulness of the executable file. Semantic analysis is divided into static and dynamic. Static one consists in disassembling the executable file image from the drive and analysing it by finite automata. This method is ineffective due to the distribution of packers and anti-hacking software protection systems. Such programmes archive/encrypt the contents of executable files, after which the programme code cannot be disassembled. The problem is solved by using libraries of unpacking algorithms, with which the antivirus can extract packaged files. The effectiveness of the method depends on timely updating of the packer type and unpacking support.

The dynamic approach uses semantic analysis with a debugger/emulator. In this case, it is not the disassembler listing that is checked but code fragments during step-by-step execution or interpretation that have preprocessing. Commands arrive at the input of the machine sequentially, as they are interpreted by the emulator, which gives an advantage when analysing packaged or self-modified objects since it becomes possible to examine objects after the wrappers/cryptographers restore the original bodies in memory and transfer control to them. This method works without using packer libraries but is resource-intensive. Dynamic semantic analysis is widely used in most antiviruses. Its advantage is a low level of possible errors and the disadvantage is low efficiency when working with non-standard codes since the machine is most often configured to recognise a given sequence of certain characters. It is enough to find a sequence equivalent in functionality, for the recognition of which the automat is configured, and the method will lose effectiveness. For example, replacing some commands with equivalent blocks. An application programming interface (API) functions can be used instead of InternetOpenURL, and wsock32.dll, socket, send, recv instead of InternetReadFile libraries to perform similar functions. This will lead to the fact that the machine will not be able to switch to its final state and recognise the threat. For the automat to continue recognising the fragment, it is necessary to modify it considering all possible variants of the equivalent code, and this is practically impossible.

Interpretation of malicious code by the emulator. The emulator creates an artificial environment to simulate the necessary functionality for research with a high level of protection of the user's system. Even if a malicious programme or virus is running, it will not be able to harm the system or network from the emulator. The need for emulation is that the programme is not a static object, and it is not always possible to recognise it by static methods. Examples of such programmes are polymorphic viruses and packers. The static signature is programmed to appear after executing a certain number of commands. Commands can be executed, as well as debuggers – sequentially, making a stop after executing the command, setting the processor to single-stepping mode. This is how code tracing is used in real conditions. The condition will be the need to monitor the work of the human debugger to control the process and stop it. In addition, the operator will be able to determine any access of programmes to the external environment. Antiviruses, when monitoring the actions of running programmes, also use debuggers that are completely controlled by their internal control system – the proactive protection module. This module can recognise and block standard types of unauthorised actions, working as a kind of filter. Notably, such an analysis will take place in real-time execution of programmes, and this will allow receiving up-to-date data on the presence of threats. The analyser examines the interaction of the programme with the system (checks the arguments of API function calls), giving the programme the access to real computer resources, which guarantees the correctness of the results. The disadvantage of this method is the possibility of threats entering the system and malfunction. For security purposes, programmes are run in the emulator to proactively protect against unauthorised access. During emulation, the programme is executed on an interpreter that reproduces the external environment: devices,

memory, system calls. The logic of threat recognition remains similar to proactive methods. The disadvantages of emulation are:

1. The need to model computer hardware nodes and OS components. This is a complex process and requires a detailed study of the systems on which these methods and software products will work. There are many programmes on the market that emulate the main modules of the system when working with intrusions but virus writers do not stand still and have learnt to bypass even emulators. Important in emulation is modelling of work on the Internet. Here it is necessary to simulate the functions of downloading and receiving files and then saving them to disc.

2. Low performance. The emulator describes the processes of a running system, and this requires the use of huge computing resources. The speed of the software model is much lower than that of the hardware counterpart, even considering its maximum possible optimisation. Some developers create special emulation systems using physical processors, which allows them to conduct research with maximum capacity.

3. The limitation of the "depth of emulation". Most often, a limited set of commands is used during emulation. Only a part of the programme is emulated and the emulator stops working either after executing each instruction, or by emulating the operation of blocks, and only then conducts tests for threats. At the end of the programme, its emulation loses its meaning since it goes through a cycle of waiting for new events. Such actions can be new input data or commands. When switching to standby mode, the emulator should shut down and move on to the next programme. There are two approaches to determine the onset of such a cycle. The first is by setting the emulation steps. For example, to finish the emulation after performing a specified number of different steps or operations. To finish the emulation by completing a certain number of steps or a certain number of different commands. The second approach is to set the time for emulation. After the time allotted for execution, the programme finishes the emulation. This method will be able to bypass the limitations of the command counter. It will not slow down the system because empty cycles do not load the system even when repeated. The time for performing these operations is set experimentally.

The disadvantages of heuristic methods are that heuristic analysis produces a lot of false detections when working, as well as complexity: despite the complexity of the correct setup, the method can slow down the system. These disadvantages, due to the specific features of heuristic methods, are difficult to compensate even with hardware. With their further use, it is advisable to modify and refine them to fulfil the intrusion detection task assigned to them. For the effective use of heuristic analysis methods, it is necessary:

1) To increase the effectiveness of recognising new threat modifications.
2) To reduce the level of false responses.
3) To increase the speed of the method.
4) To minimise the degree of use of system resources.
5) The possibility of adaptability.

Modern heuristic methods based on behavioural analysis and emulators allow effectively identifying unauthorised actions of a known type and unknown to the system. Artificial intelligence methods allow implementing learning opportunities for threat diagnostics. These methods can considerably improve the reliability and protection of computer systems and networks. Numerous heuristic methods are implemented based on the use of neural networks, artificial immune networks, and multi-agent systems that can detect intrusions. Notably, these methods have certain disadvantages that either slow down the overall performance of data processing systems or sacrifice accuracy. To choose the optimal tools, let us consider the most popular methods of threat detection. Production systems are systems that use a production model of knowledge representation. The production model of knowledge representation is one of the most common. The representation of knowledge through the rules has similarities in some respects with the rules of inference of logical models. This allows carrying out effective inference through products and, in addition, due to the natural analogy of the human reasoning process, these models more clearly reflect knowledge. In production systems, knowledge is represented using sets of rules of the form: "if A, then B". Here A and B can be understood as "situation – action", "cause – effect", and "condition – conclusion". However, one should not equate the production rule and the logical sequence relation. The fact is that the interpretation of products depends on what is located to the left and right of the logical sequence sign. Commonly, A implies some information structure (for example, a frame), and U implies some action that consists in its transformation (transformation). The logical interpretation of the expression under consideration imposes restrictions on A and B. Generically, such a model has the following form:

$$P = (K, U, A \rightarrow B, E), \quad (1)$$

where: K is the class of this situation; U is the activation condition; A → B is the essence of the product; E is the termination condition. The production model can be simplified by the order or priority that the entire set of products can be introduced. The order means that each subsequent product is used only when the previous product is not suitable. With priorities, the products with the highest priority are initially used. To combat contradictions when expanding databases (for example,

the same priority), it is possible to use returns. The components of such a system are a knowledge base, working memory, and an output mechanism. The knowledge base, using products, describes the subject area. The working memory contains facts about the current stage of logical inference. The output mechanism selects the rules that the product data can fulfil. Production systems are used in the signature analysis method. Such an analysis is effective only when intrusions occur according to the same algorithms (signatures). If the attack scenario is known, then it is compared with the user's actions and if actions similar to intrusions are detected, they are blocked or deleted. If the signature does not fully correspond to the user's action but partially, then the option of notifying the operator or the antivirus system about the possibility of intrusion is possible. Most often, this method is used by network attack detection systems (Snort, RealSecure, eTrustID, Antivirus by Zaitsev (AVZ), KasperskyLab, Microsoft Security Essentials (MSE)). Signature methods are widely used today and need constant updates to maintain efficiency and competitiveness. Production models can confirm an intrusion by working through audit files, running processes, and network ports. The advantages of production systems are as follows:

1) modularity – each rule describes a small, relatively independent piece of knowledge;
2) incrementality – the ability to add new rules independently of other rules;
3) The convenience of modification as a consequence of modularity and incrementality;
4) transparency of the system (ease of tracing the logic and explanation of the conclusion).

Disadvantages of production models:

1) the withdrawal process has low efficiency since, with a large number of products, a considerable part of the time is spent on non-production verification of the conditions for applying the rules;
2) checking the consistency of the production system becomes very difficult due to the non-deterministic choice of the products to be performed from the conflicting set.

Most of the shortcomings can be corrected by optimising for a particular production system. The statistical method considers the appearance of characteristic signs, according to which it is concluded that there is unauthorised activity. This method produces probabilistic conclusions about the presence of intrusions (when the behaviour of the system has stopped proceeding in a routine way, the statistical method will be able to identify it). When calculating the frequency of access to processor commands, a table of their activity is built, based on which a decision is made about the presence or absence of an intrusion. This method perfectly detects polymorphic viruses that use a minimal set of commands in the descriptor. A popular method is based on the operators of probability theory and described by the following Bayes' equation:

$$P(D_i|S_j)= P(D_i) * P(S_i|D_j)| (P(D_1) * P(S_j |D_1) + P(D_2) * P(S_j|D_2) + (...), \quad (2)$$

where: Sj – events; Di – diagnosis; P(Di|Sj) is the probability of correctness of the I-th diagnosis detection of the j event; P(Di) is the probability of the i diagnosis; P(Sj|Di) is the conditional probability of occurrence of the i feature of the event j. A sample of viruses and programmes is being built, where the final data will be designated P(D i). Then viruses are taken and ratings P(SKDi) are determined. The results are used to build data on the presence of threats. Intrusions are scanned in the system to generate a sample of events K, S = { S1, S2, ..., SK}. Then the probability is found that this activity will be diagnosed Di. To diagnose, the probabilities P(Di|S j) for Sj from the set S are calculated. Next is the probability:

$$P(D_i |S) = P(D_i | S_1) * P(D_i | S_1) * P(D_i| S_2) *\cdots* P(D_i | S_K). \quad (3)$$

Then the probabilities of intrusions are calculated and the largest one is selected. This approach is not effective when working with email clients since not all signs can be described by this method. Artificial neural systems (ANN) are models of the neural structure of the brain that mainly learn from experience. The natural analogue proves that many problems that are not yet subject to the solution by machines can be successfully solved through neural networks. The ANN is a network of processors (neurons). They process and transmit information to the following neurons. ANN are solving increasingly complex tasks step by step. The feature of ANN is training. During training, connections of neurons are detected that determine the ratio of input and output signals. During training, the ANN begins to identify additional dependencies between input and output data. When the network is trained, it can get the correct answer for new data that was not in the initial sample. The correct solution will be obtained if the original data is incomplete. The ANN allows independently receiving and processing data and the ability to generalise data. The ANN is combined with various types of architectures. The most popular application of ANN is recurrent and of direct distribution architectures. The training of the ANN can be carried out according to the initial data, without the initial data (the ANN produces a solution from the input data) and with reinforcement (using penalties). ANN can be implemented with existing complexes (Neuro solutions, Matrix Laboratory (MATLAB)) or known algorithms. To build such a network, it is necessary to have training samples. The development of a training sample depends on the process and purpose of implementation. ANN can also be implemented when building network protection systems. Based on the

ANN, it is possible to organise the detection of unauthorised behaviour in networks.

The main thing in the development of the ANN is to set up the initial sample since if an error occurs, the learning process will take a lot of time and will consume a lot of resources. After the selection is set, the process is irreversible since it will be completely controlled by the embedded algorithm. It can only signal the input and check the output. Due to this, tracking the work of the ANN is quite difficult. The network topology is selected according to the implementation environment. The method has a number of disadvantages since it is impossible to correctly select all the functions and their description from the first time. But the configuration of the entire system and its testing before implementing it into the network allows getting excellent performance both in speed and reliability. The most important thing will be to describe the algorithm considering all the subtleties of the process.

Multi-agent systems consist of agents performing tasks assigned to them. They work out only those tasks for which they are designed. Such agents are independent among themselves and there are no disputes between them. The main task is divided into subpoints, the elaboration of which is responsible for the relevant agents. The correct organisation of agents will allow correctly and quickly completing the task, which is performed through the distribution of the entire task into sub-tasks. To solve the problem, agents are grouped under the direction of the operations centre. Multi-agent systems have proven themselves well in protecting against Distributed Denial-of-service attacks (DDoS), which result in the failure of hosts, services, or the shutdown of DNS servers (Domain Name System server) and disruption of the network. This method is proposed in the works when working out DDoS attacks. User agents, violators, and defenders are programmed that can interact. Then the agents of the violators are divided into "demons" (the attack itself) and "masters" (control the attacks). Defenders are divided into: "samplers", "detectors", "filters", and "investigation agents". The "sampler" accumulates information and transmits it to the "detectors", which are responsible for responding to the onset of an attack. "Filters" monitor incoming information, and "investigation agents" counteract attack agents. An agent-oriented system has been created, which shows the work of agents to detect intrusions in the network. The test results showed satisfactory effectiveness of the method in laboratory conditions. This method has enough positive aspects both for independent use and for combining it with other methods to increase productivity. Currently, all modifications are applied mutually with the methods of both neural networks and artificial networks since this is facilitated by sufficiently high adaptability of multi-agent systems.

Artificial immune systems are one of the tools for detecting threats. They interact closely with neural networks, genetic algorithms, and artificial neural networks. Artificial neural networks that work based on immunological methods that were first discovered in medicine and were based on the work of leukocytes. The natural immune system consists of many functional complexes. The function of the immune system in the classification of body cells for the organisation of the immune response. The analysis of works and models allows identifying the characteristics of immune systems that can be adapted to detect threats in computer networks:

- the use of antibodies to process antigens (similar to forming a response to an intrusion into a computer network);
- the use of antigens as intrusions and the development of appropriate immune responses using antibodies;
- the possibility of training antigens to work on the detection of antigens. Cloning, selection, and removal allow detecting antigens quickly and as correctly as possible through antibodies;
- the possibility of forming an immune response to any invasion (by preserving antibodies corresponding to antigens);
- regulation of the mode of selection of the necessary antibodies by working with the similarity of antibodies (threshold of compliance of the antibody to the antigen). The similarity allows counteracting the antigen with an antibody as close to it as possible;
- the immune system allows remembering and storing all antibodies that correspond to the antigen known to the system (the memory of the immune system is formed, which functions as a database of signatures);
- the use of antibodies with high similarity allows responding to new unknown intrusions;
- the ability for scaling and adaptability.

By describing the functions of antibody paratopes and epitopes, it is possible to create mathematical models of natural neural networks and apply them in their implementation into information systems. It is through paratopes and epitopes that antibodies and antigens are combined. Such systems have a wide range of applications, ranging from optimisation tools, scanning and pattern recognition systems, building computer and network systems, and ending with the classification of information. Natural neural networks can be used to solve the following tasks: computer security, optimisation of numerical functions, combinatorial optimisation, training, bioinformatics, robotics, adaptive control system, data output, anomaly detection or error diagnosis. Immune systems are a universal model that can be used to solve a variety of tasks. At the moment, natural neural networks are of the greatest interest for use in computer systems. It is in the works that the principles of natural neural networks are used to detect

intrusions and attacks in networks. The work describes the creation of natural neural networks with process monitoring elements based on the principles of negative selection (removal of unnecessary antibodies from the memory matrix) when identifying differences in user actions and attacks on the system. Various methods based on differential equations of delay or derivatives, as well as agent-based models and stochastic differential equations are used to describe the operation of natural neural networks.

A model for detecting unauthorised actions using elements of immune models has been developed. The work of anomalies used in the construction of hybrid elements of network security and capable of detecting attacks with a low level of false responses is described. There is a disadvantage in such systems, which consists in generating a large amount of attack traffic before starting the method. Thus, these methods cannot work in real-time conditions. An adaptive approach based on immune mechanisms has been developed. Natural neural networks repeat the behaviour of the defence of the immune systems of living organisms. The above works demonstrate the prospects of natural neural networks and when working out errors. The application of natural neural networks for error processing is described. The work of agents in the implementation of the detection of unauthorised actions is described. These methods are also based on algorithms of the immune systems of living organisms. Agents act as monitors for each other when performing a common task for them. Each agent monitors other agents for compliance with their task. If the agents cannot deal with the task, they are removed and replaced by others. Most often, the agents work on the algorithms of dendritic cells of the immune system.

The method of combined use of neural networks when working with traffic is described. Such a system is independently trained on input data for the possibility of self-detection of intrusions. The main advantages of such a system are autonomy and high accuracy of threat detection. Low performance is conditioned upon the long learning process. Describe the creation of a multi-agent neural network with the ability to maintain statistics of vulnerable network elements. The approach allows not only monitoring such nodes more closely but also testing them for intrusion. The possibility of predicting the values of time series using a neural network is considered. A method of clonal selection has also been developed for evaluating the correctness of data. The works are based on the application of diagnostic methods of diseases in medicine to build systems for detecting computer attacks using the theory of neural networks. The search for unauthorised actions was carried out based on combining methods of immune and neural networks. The analysis of modern intrusion detection methods shows that the main area of improving the efficiency of systems lies in combining various methods and technologies to solve problems. There is also a need to create software systems that will help to improve the reliability and efficiency of existing systems for detecting unauthorised actions in computer networks.

In the field of computer security, many resources are aimed at improving the effectiveness of protecting users from unauthorised actions in cyberspace. One of the ways to increase the security of a computer system is to use an intrusion detection system. The diagnostic method is related to intrusion detection systems but is based on principles developed in medicine. Through informatisation in the field of medicine, the development of new milestones in computer technology becomes possible. Unfortunately, such methods are often purely theoretical, that is, those that work only on paper and mathematical models. The main difficulty in their design is the selection of the right technological and methodological base, which will allow such methods to be applied in real conditions. Such methods are based on the diagnosis of intrusions. When diagnosing intrusions, the corresponding functions of the system are monitored, in the same way as in medicine, an examination is carried out, and symptoms of the disease are detected. Symptoms use the values of examinations to calculate the probability of detecting a certain disease in the human body. Binary symptoms, as a special class of symptoms, use change detection algorithms: threshold or Correspondences by Sensitivity to Movement (CSM) to calculate and form signatures of attacks or unauthorised actions. The Dempster-Schafer theory [20] is used to characterise such beliefs, operating with the basic concepts of combining and using merge operators.

The diagnosis is made by analysing the probability of assessment reliability of various sets of the combined evidence penetration structure states of symptoms for the diagnostic tree represented by the specification tree to determine the security of the system or organism as accurately as possible. With this process, the intrusion detection system can combine the characteristics of both signatures and anomaly detection systems, and can detect known or new attacks and unauthorised actions of the system. The system can detect previously known and new attacks similar to the types of intrusions previously declared by the system. The systems built in this way have shown good efficiency when working with pre-defined DoS attacks. The intrusion detection system was built and configured against several known cases of Transmission Control Protocol (TCP) and DoS attacks, which the method could correctly diagnose at runtime. Pseudo-intrusions that were not known to the system in advance were also declared. Such types of intrusions were constructed from the characteristics of known attacks for the testing system and were quite correctly diagnosed and detected. Existing software solutions have some support in terms of intrusion prevention and detection but they lack the ability to diagnose. There are prerequisites for the application of

medical diagnostic methods in computer systems. Existing methods cannot fully identify all threats in networks, so combining new approaches with existing methods can give good results and increase reliability. These systems have disadvantages in the timeliness of detecting attacks but the use of diagnostic methods can positively affect the performance of known methods, minimising the cost of monitoring the network and streams.

4. CONCLUSIONS

Collecting information at multiple architectural levels using multiple security filters to perform a correlation analysis of intrusion symptoms allows identifying the symptoms of intrusion first, and then their causes. With their use, it is possible to assess losses in individual components of the system. Previous theoretical methods have shown effectiveness but have not been implemented in real conditions and systems. When talking about diagnostics, the authors imply the ability to clearly identify the causes of intrusions and assess their effects on an individual system of components. Through diagnostics, it is possible not only to detect an intrusion but also to prevent it in advance, using methods of constructing a tree of diagnoses (similar to signatures but working without updates and identifying new forms of attacks and modifications of old ones). This technology can expand the capabilities of intrusion detection systems, raising them to a qualitatively new level. The main idea is to collect information at several architectural levels (network, operating system, databases, and applications) using security filters and complex event processing technology to perform a correlation analysis of intrusion symptoms.

The idea of collecting information from sources was theoretically described earlier. The methods presented in this paper use the concepts of correlation and multi-analysis but do not solve the problem of diagnosing anomalies in the system. The proposed approach considers the process of escalation from the symptoms of the invasion to determining the causes of the invasion and assessing the damage caused using ontologies. Two sets work in pairs: the first allows observing the symptoms, and the second allows making verdicts about the presence of attacks or anomalies. The output of this process can then be used for recovery processes, and eventually to ensure the reliability of detecting unauthorised actions. Theoretical tests have shown that this approach improves the results of detecting unauthorised actions in terms of increasing the reliability of the decision-making process. This method can detect new attacks without having any information about them. It is possible to encapsulate signatures, which will not only detect an intrusion but also determine the class and type of attacks.

References:

Akritidis, L., Fevgas, A., Tsompanopoulou, P., & Bozanis, P. (2020). Evaluating the effects of modern storage devices on the efficiency of parallel machine learning algorithms. *International Journal on Artificial Intelligence Tools, 29*(3-4), article number 2060008. https://doi.org/10.1142/S0218213020600088

Aksvonov, K., & Antonova, A. (2018). Development of a Hybrid Decision-Making Method based on a Simulation-Genetic algorithm in a Web-Oriented Metallurgical Enterprise Information System. International Conference on Ubiquitous and Future Networks, 2018-July, 197-202. https://doi.org/10.1109/ICUFN.2018.8436676

Dong, B., Zhu, X., Yan, R., & Wang, Y. (2019). Development of optimization model and algorithm for storage and retrieval in automated stereo warehouses. *Journal Europeen des Systemes Automatises, 52*(1), 17-22. https://doi.org/10.18280/jesa.520103

Gaoyu, J., Jingchang, P., & Bo, Z. (2019). Storage design and implementation of information reconstruction system. In: *ACM International Conference Proceeding Series* (pp. 1-5). New York: Association for Computing Machinery. https://doi.org/10.1145/3372454.3372455

Jain, S., Chaudhary, H., & Bhatnagar, V. (2013). An information security-based literature survey and classification framework of data storage in DNA. *International Journal of Networking and Virtual Organisations, 13*(2), 176-201. https://doi.org/10.1504/IJNVO.2013.059688

Jeong, D., Son, J., Baik, D.-K., & Yang, L.T. (2008). View-based storage-independent model for SPARQL-to-SQL translation algorithms in semantic grid environment. In: *Proceedings of the 11th IEEE International Conference on Computational Science and Engineering, CSE Workshops 2008* (pp. 381-386). Piscataway: Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/CSEW.2008.72

Lavrov, E., Barchenko, N., Pasko, N., & Borozenec, I. (2017). Development of models for the formalized description of modular e-learning systems for the problems on providing ergonomic quality of human-computer interaction. *Eastern-European Journal of Enterprise Technologies*. Series "*Information Technology", 2*(2(86)), 4–13.

Lavrov, E., Pasko, N., & Krivodub, A. (2015). Automated analysis of the effectiveness of ergonomic measures in discretecontrol systems. *Eastern-European Journal of Enterprise Technologies,4/3*(76),16–22.

Liu, S., Ye, Z., & Chen, M. (2020). Next generation information storage technology: DNA Data Storage. In: *ACM International Conference Proceeding Series* (pp. 213-217). New York: Association for Computing Machinery. https://doi.org/10.1145/3383972.3383999

Liu, Y., & Zhang, S. (2020). Information security and storage of Internet of Things based on block chains. *Future Generation Computer Systems, 10*6, 296-303. https://doi.org/10.1016/j.future.2020.01.023

Rui, H., Huan, L., Yang, H., & YunHao, Z. (2020). Research on secure transmission and storage of energy IoT information based on Blockchain. *Peer-to-Peer Networking and Applications, 13*(4), 1225-1235. https://doi.org/10.1007/s12083-019-00856-7

Shokrzadeh, S., Jafari Jozani, M., Bibeau, E., & Molinski, T. (2015). A statistical algorithm for predicting the energy storage capacity for baseload wind power generation in the future electric grids. *Energy, 89*, 793-802. https://doi.org/10.1016/j.energy.2015.05.140

Sivapriya, K., & Kartheeban, K. (2018). Design and development of security algorithm for data storage in multicloud environment. *Proceedings of the 2017 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing*, 2018-February, 1-7. https://doi.org/10.1109/ITCOSP.2017.8303137

Sivaram, M., Kaliappan, M., Shobana, S.J., Prakash, M. V, Porkodi, V., Vijayalakshmi, K., & Suresh, A. (2021). Secure storage allocation scheme using fuzzy based heuristic algorithm for cloud. *Journal of Ambient Intelligence and Humanized Computing, 12*(5), 5609-5617. https://doi.org/10.1007/s12652-020-02082-z

Son, J., Jeong, D., & Baik, D.-K. (2008). Practical approach: Independently using SPARQL-to-SQL translation algorithms on storage. *Proceedings – 4th International Conference on Networked Computing and Advanced Information Management, 2*, 598-603. https://doi.org/10.1109/NCM.2008.151

Song, L.-L., Wang, T.-Y., Zhang, L.-Y., & Song, X.-W. (2015). Research and application of redundant data deleting algorithm based on the cloud storage platform. *Open Cybernetics and Systemics Journal, 9*(1), 50-54. https://doi.org/10.2174/1874110X01509010050

Tian, J. (2021). Based on the optimal learning algorithm of Vortex Search, the prediction of oil field development index is studied. *Journal of Physics: Conference Series, 1881*(3), article number 032095. https://doi.org/10.1088/1742-6596/1881/3/032095

Wang, Z., Yu, K., Wang, W., Yu, X., Kang, H., Li, Y., & Yang, T. (2021). Android Secure Cloud Storage System based on SM algorithms. *Advances in Intelligent Systems and Computing, 1274* AISC, 772-779. https://doi.org/10.1007/978-981-15-8462-6_88

Xiaoru, L., & Ling, G. (2021). Combinatorial constraint coding based on the EORS algorithm in DNA storage. *PLoS ONE, 16*(7 July 2021), article number e0255376. https://doi.org/10.1371/journal.pone.0255376

Zhang, D. (2021). Storage optimization algorithm design of cloud computing edge node based on artificial intelligence technology. *Journal of Ambient Intelligence and Humanized Computing.* https://doi.org/10.1007/s12652-021-03272-z

Zohid A. Hakimov
Department of Information Technologies
Urgench Branch of Tashkent University of Information Technologies named after Muhammadal Khwarizmi
220100, 110 Al-Khorezmi Str., Urgench, Republic of Uzbekistan
zohid.hakimov@outlook.com
ORCID 0000-0002-4264-6438

Asilbek Medatov
Department of Information Technology
Andijan State University
170100, 129 Universitetskaya Str., Andijan, Republic of Uzbekistan
asilmedatov@outlook.com
ORCID 0000-0002-3840-6589

Viktor Kotetunov
Department of Information Systems and Technologies
National Transport University
01010, 1 Mykhailo Omelianovych-Pavlenko Str., Kyiv, Ukraine
vikkotetunov@gmail.com
ORCID 0000-0002-0214-3369

Yuriy Kravtsov
Department of Sociology
Dniprovsk State Technical University
51918, 2 Dniprobydivska Str., Kamianske, Ukraine
kravtsov_yuriy@yahoo.com
ORCID 0000-0003-4968-9112

Alisher Abdullaev
Department of Informatics Teaching
Nukus State Pedagogical Institute named after Ajiniyaz
230100, 1 P. Seyitov Str., Nukus, Uzbekistan
al-abdullaev@outlook.com
ORCID 0000-0003-4904-7734