UDC 004.056
M78

Published in 2023
by PC TECHNOLOGY CENTER
Shatylova dacha str., 4, Kharkiv, Ukraine, 61165

Approved by the Academic Council of National Technical University «Kharkiv Polytechnic Institute», Protocol No. 5 of 23.03.2023

Reviewers:
Dudykevych Valerii, Doctor of Technical Science, Professor, Head of the Department of Information Security of Lviv Polytechnic National University;
Korchenko Alexandr, Doctor of Technical Sciences, Professor, Head of the Department of Information Technology Security of National Aviation University.

M78 **Authors:**
Edited by **Serhii Yevseiev, Yuliia Khokhlachova, Serhii Ostapov, Oleksandr Laptiev**
Serhii Yevseiev, Yuliia Khokhlachova, Serhii Ostapov, Oleksandr Laptiev, Olha Korol, Stanislav Milevskyi, Oleksandr Milov, Serhii Pohasii, Yevgen Melenti, Hrebeniuk Vitalii, Alla Havrylova, Serhii Herasymov, Roman Korolev, Oleg Barabash, Valentyn Sobchuk, Roman Kyrychok, German Shuklin, Volodymyr Akhramovych, Vitalii Savchenko, Sergii Golovashych, Oleksandr Lezik, Ivan Opirskyy, Oleksandr Voitko, Kseniia Yerhidzei, Serhii Mykus, Yurii Pribyliev, Oleksandr Prokopenko, Andrii Vlasov, Nataliia Dzheniuk, Maksym Tolkachov

Models of socio-cyber-physical systems security: monograph / S. Yevseiev, Yu. Khokhlachova, S. Ostapov, O. Laptiev and others. – Kharkiv: PC TECHNOLOGY CENTER, 2023. – 168 p.

The monograph discusses the methodology for cooperative conflict interaction modeling of security system agents. The concept of modeling the structure and functioning of the security system of critical infrastructure facilities is demonstrated. The method for assessing forecast of social impact in regional communities is presented. Counteracting the strategic manipulation of public opinion in decision-making by actors of social networking services based on the conceptual model for managed self-organization in social networking services are developed. Algorithms for thinning the critical infrastructure identification system and their software are implemented.

The monograph is intended for teachers, researchers and engineering staff in the field of cybersecurity, information technology, social engineering, communication systems, computer technology, automated control systems and economic information security, as well as for adjuncts, graduate students and senior students of relevant specialties.

Figures 60, Tables 32, References 132 items.

DOI: 10.15587/978-617-7319-72-5
ISBN 978-617-7319-72-5 (on-line)
ISBN 978-617-7319-73-2 (print)

9 786177 319725

AUTHORS

SERHII YEVSEIEV
Doctor of Technical Science, Professor, Head of Department
Department of Cyber Security
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0003-1647-6444

YULIIA KHOKHLACHOVA
PhD, Associate Professor
Department of Security of Information Technologies
National Aviation University
ORCID ID: https://orcid.org/0000-0002-1883-8704

SERHII OSTAPOV
Doctor of Physical and Mathematical Sciences, Professor, Head of Department
Department of Computer Systems Software
Y. Fedkovych Chernivtsi National University
ORCID ID: https://orcid.org/0000-0002-4139-4152

OLEKSANDR LAPTIEV
Doctor of Technical Science, Senior Researcher
Department of Cyber Security and Information Protection
Taras Shevchenko National University of Kyiv
ORCID ID: https://orcid.org/0000-0002-4194-402X

OLHA KOROL
PhD, Associate Professor
Department of Cyber Security
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0002-8733-9984

STANISLAV MILEVSKYI
PhD, Associate Professor
Department of Cyber Security
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0001-5087-7036

OLEKSANDR MILOV
Doctor of Technical Science, Professor
Department of Cyber Security
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0001-6135-2120

SERHII POHASII
PhD, Associate Professor
Department of Cyber Security
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0002-4540-3693

YEVGEN MELENTI
PhD, Associate Professor
Special Department No. 5
National Academy of Security Service of Ukraine
ORCID ID: https://orcid.org/0000-0003-2955-2469

VITALII HREBENIUK
Doctor of Science in Law, Senior Reseacher
First Vice-Rector
National Academy of Security Service of Ukraine
ORCID ID: https://orcid.org/0000-0002-5169-8694

ALLA HAVRYLOVA
Senior Lecturer
Department of Cyber Security
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0002-2015-8927

SERHII HERASYMOV
Doctor of Technical Sciences, Professor
Department of Cyber Security
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0003-1810-0387

ROMAN KOROLEV
PhD
Department of Cyber Security and Information Technology
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0002-7948-5914

OLEG BARABASH
Doctor of Technical Sciences, Professor
Department of Automation of Designing of Energy Processes and Systems
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0003-1715-0761

VALENTYN SOBCHUK
Doctor of Physical and Mathematical Sciences, Professor
Department of Integral and Differential Equations
Taras Shevchenko National University of Kyiv
ORCID ID: https://orcid.org/0000-0002-4002-8206

ROMAN KYRYCHOK
PhD
Department of Information and Cyber Security named after Professor Volodymyr Buriachok
Borys Grinchenko Kyiv University
ORCID ID: https://orcid.org/0000-0002-9919-9691

GERMAN SHUKLIN

PhD, Associate Professor, Head of Department
Department of Information and Cybersecurity Systems
State University of Telecommunications
ORCID ID: https://orcid.org/0000-0003-2507-384X

VOLODYMYR AKHRAMOVYCH

Doctor of Technical Sciences, Senior Researcher, Professor
Department of Information and Cybersecurity Systems
State University of Telecommunications
ORCID ID: https://orcid.org/0000-0002-6174-5300

VITALII SAVCHENKO

Doctor of Technical Sciences, Professor, Director of Institute
Educational and Scientific Institute of Information Protection
State University of Telecommunications
ORCID ID: https://orcid.org/0000-0002-3014-131X

SERGII GOLOVASHYCH

PhD, Associate Professor
Department of Software Engineering and Management Intelligent Technologies
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0009-0004-2468-1952

OLEKSANDR LEZIK

PhD, Associate Professor
Department of Air Defense Forces Tactics of the Land Forces
Ivan Kozhedub Kharkiv National Air Force University
ORCID ID: https://orcid.org/0000-0002-7186-6683

IVAN OPIRSKYY

Doctor of Technical Science, Professor
Department of Information Security
Lviv Polytechnic National University
ORCID ID: https://orcid.org/0000-0002-8461-8996

OLEKSANDR VOITKO

PhD, Head of Center
Educational and Scientific Center of Strategic Communications in the Field of Ensuring National Security and Defense
Institute of Troops (Forces) and Information Technologies
National Defence University of Ukraine named after Ivan Cherniakhovskyi
ORCID ID: https://orcid.org/0000-0002-4610-4476

KSENIIA YERHIDZEI

PhD
Educational and Scientific Center of Strategic Communications in the field of Ensuring National Security and Defense
Institute of Troops (Forces) and Information Technologies
National Defence University of Ukraine named after Ivan Cherniakhovskyi
ORCID ID: https://orcid.org/0000-0003-4634-133X

SERHII MYKUS

Doctor of Technical Science, Professor
Institute of Information and Communication Technologies and Cyber Defense
National Defence University of Ukraine named after Ivan Cherniakhovskyi
ORCID ID: https://orcid.org/0000-0002-7103-4166

YURII PRIBYLIEV

Doctor of Technical Science, Professor
Department of Information Technologies Employment and Information Security
National Defence University of Ukraine named after Ivan Cherniakhovskyi
ORCID ID: https://orcid.org/0000-0003-1941-3561

OLEKSANDR PROKOPENKO

PhD
Laboratory Detection and Forecasting of Information Threats
Educational and Scientific Center of Strategic Communications in the Field of Ensuring National Security and Defense
National Defence University of Ukraine named after Ivan Cherniakhovskyi
ORCID ID: https://orcid.org/0000-0002-5482-0317

ANDRII VLASOV

PhD, Associate Professor
Department of Information Technology Security
Kharkiv National University of Radioelectronics
ORCID ID: https://orcid.org/0000-0001-6080-237X

NATALIIA DZHENIUK

Associate Professor
Department of Information Systems
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0003-0758-7935

MAKSYM TOLKACHOV

Associate Professor
Department of Information Systems
National Technical University "Kharkiv Polytechnic Institute"
ORCID ID: https://orcid.org/0000-0001-7853-5855

ABSTRACT

The development of technologies and computing resources has not only expanded the range of digital services in all spheres of human activity, but also determined the range of targeted cyber attacks. Targeted attacks are aimed at destroying not only the business structure, but also its individual components that determine critical business processes. The continuity of such business processes is a critical component of any company, organization or enterprise of any form of ownership, which has a critical impact on making a profit or organizing production processes. The proposed concept of determining the security level of critical business processes is based on the need to use multiloop information security systems. This makes it possible to ensure the continuity of critical business processes through a timely objective assessment of the level of security and the timely formation of preventive measures. This approach is based on the proposed rules for determining the reach of a given security level, based on assessments of the integrity, availability and confidentiality of information arrays, as well as computer equipment for different points of the organization's business processes. The issues of applying situational management methods to ensure the safe functioning of objects of socio-cyber-physical systems, logical and transformational rules that form the foundation for building a situational type cybersecurity management system are considered. One of the main tasks of systems of this type is described – the task of replenishing the description of the situation. The use of pseudophysical logics, various types of pseudophysical logics, the method of their construction and their interconnection are proposed. Particular attention is paid to causal pseudophysical logic, as the least developed for the purposes of ensuring cybersecurity. The formation of smart technologies, as a rule, uses the wireless standards of communication channels IEEE 802.11X, IEEE 802.15.4, IEEE 802.16, which use only authentication protocols and privacy mechanisms that are formed on the basis of symmetric algorithms. In the conditions of the post-quantum period (the appearance of a full-scale quantum computer), the stability of such algorithms is questioned. Such systems, as a rule, are formed on the basis of the synthesis of socio-cyber-physical systems and cloud technologies, which simplifies the implementation of Advanced Persistent Threat attacks, both on the internal loop of control systems and on the external one.

The proposed creation of multi-circuit information protection systems allows for an objective assessment of the flow state of the system as a whole and the formation of preventive measures against cyber threats.

In the third chapter, models of probable threats and information protection in public networks are proposed. The most general model of the formal description of the protection system is the model of the security system with full overlap, in which a complete list of protection objects and threats to information is determined, and means of ensuring security are determined from the point of view of their effectiveness and contribution to ensuring the security of the entire tele-

communications system. It is also shown that the combination of four models (M1, M2, M3, M4) in various variants provides wide opportunities for modeling various known types of threats and their implementation. However, in connection with the continuity of the process of developing new and improving existing methods and means of implementing threats, it is necessary to use such approaches to ensuring information protection that allow detecting and preventing threats of unknown types and carrying out dynamic correction of protection behavior, adapting it to specific application conditions. The M5 basic model is described, which enables continuous refinement of threat classes and response measures, and continuous training of the adaptive component of the CSI, which, in turn, detects and prevents threats of unknown types. The M6 basic model is introduced with the aim of obtaining higher security due to the presence of a special module of internal diagnostics that diagnoses the entire protection system, decides on the correction of the SHI behavior algorithm, and makes it possible to achieve SHI fault tolerance; a special module that diagnoses the communication channel with subsequent changes in the level of protection, allows to achieve the adaptability of the SHI.

The fourth chapter is deal with the development of cryptographic primitives based on cellular automata. The definition of a cellular automaton is given and the elementary rules of intercellular interaction are described.

A number of generators of pseudorandom binary sequences have been developed based on a combination of elementary rules of intercellular interaction, as well as cell interaction according to a rule of our own development.

In the "cryptographic sponge" architecture, a cryptographic hashing function with a shuffling function based on cellular automata was developed and its statistical characteristics and avalanche effect were investigated.

A block cipher in the SP-network architecture is constructed, in which cellular automata are used to deploy the key, and the encryption process is based on elementary procedures of replacement and permutation. Substitution blocks are used from the well-known AES cipher, a description of a stream cipher is given, where a personal computer keyboard and mouse are used as the initial entropy. Random data received from the specified devices is processed by a proprietary hashing function based on a "cryptographic sponge". All developed cryptographic functions and primitives demonstrated good statistical characteristics and avalanche properties.

The fifth chapter proposes a methodology for analyzing the quality of the mechanism for validating the identified vulnerabilities of a corporate network, which is based on integral equations that take into account the quantitative characteristics of the vulnerability validation mechanism under study at a certain point in time. This technique allows to build the laws of distribution of quality indicators of the vulnerability validation process and quantify the quality of the mechanism for validating detected vulnerabilities, which allows to monitor and control the validation of identified vulnerabilities in real time during active security analysis. A method is proposed for constructing a fuzzy knowledge base for making decisions when validating vulnerabilities of software and hardware platforms with an active analysis of the security of a target corporate network based on the

use of fuzzy logic, which makes it possible to provide reliable information about the quality of the mechanism for validating vulnerabilities indirectly. The constructed knowledge base allows to form decisive decision-making rules for the implementation of a particular attacking action, which allows to develop expert systems to automate the decision-making process when validating the identified vulnerabilities of target information systems and networks. An improved method of automatic active security analysis is proposed, which, based on the synthesis of the proposed models, techniques and methods, allows, in contrast to the existing ones, to abstract from the conditions of dynamic changes in the environment, i.e. constant development of information technologies, which leads to an increase in the number of vulnerabilities and corresponding attack vectors, as well as an increase in ready-to-use exploits of vulnerabilities and their availability, and take into account only the quality parameters of the vulnerability validation process itself.

KEYWORDS

Cybersecurity, models of the threat, crypto-code constructions, simulation modelling, automation, radio engineering research, security measures.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

A	availability
ABS	automated banking system
Aff	affiliation
AFIS	automatic fingerprint identification
Au	authenticity
C	confidentiality
CCC McEliece	crypto code constructs McEliece/Niederreiter
CI	critical infrastructure
CIFS	critical infrastructure facilities systems
CIO	critical infrastructure objects
CPS	cyber-physical systems
CPSS	cyberphysical social system
CS	cybersecurity
DCS	distributed control systems
DDoS	denial of service attack
GIS	geospatial information systems
I	integrity
ICS	information and communication networks
IoTS	internet of things systems
IR	information resources
IS	information security
ISS	information security system
LDPC	low-density parity-check codes
LSI	a latent semantic indexing method
MCC	Matthew correlation coefficient
MCMC	method of Markov chain Monte Carlo
PLC	programmable logic controllers
SCADA	supervisory control and data acquisition
SI	information security

4 RESEARCH AND SIMULATION OF THE MECHANISM OF VULNERABILITIES VALIDATION IN ACTIVE ANALYSIS OF INFORMATION NETWORK SECURITY

ABSTRACT

A technique for analyzing the quality of the mechanism for validating the identified vulnerabilities of a corporate network has been developed, which is based on integral equations that take into account the quantitative characteristics of the mechanism for validating vulnerabilities under study at a certain point in time. This technique allows to build the laws of distribution of quality indicators of the vulnerability validation process and quantify the quality of the mechanism for validating detected vulnerabilities, which allows to monitor and control the validation of identified vulnerabilities in real time during active security analysis.

A method is proposed for constructing a fuzzy knowledge base for making decisions when validating vulnerabilities of software and hardware platforms with an active analysis of the security of a target corporate network based on the use of fuzzy logic, which makes it possible to provide reliable information about the quality of the mechanism for validating vulnerabilities indirectly.

The constructed knowledge base allows to form decisive decision-making rules for the implementation of a particular attacking action, which allows to develop expert systems to automate the decision-making process when validating the identified vulnerabilities of target information systems and networks.

The method of automatic active security analysis has been further developed, which, based on the synthesis of the proposed models, techniques and methods, allows, unlike the existing ones, to abstract from the conditions of dynamic changes in the environment, i.e. constant development of information technologies, and take into account only the quality parameters of the vulnerability validation process itself.

KEYWORDS

Cyber incident, vulnerability, warfare, risks, information security, electronic intelligence.

4.1 EXPERIMENTAL STUDY OF THE FUNCTIONING OF MODERN AUTOMATED VULNERABILITIES EXPLOITING MEANS

As mentioned earlier, modern systems of active security analysis (SASA) of information systems and networks, based on various methods of detecting and confirming vulnerabilities (in particular, methods of conducting penetration testing), allow simulating a potential cyber attack on the organization's information infrastructure and establishing its actual state security.

At the same time, the generalized algorithm of such systems consists of the following steps [77–95]:

– scanning of the target network, which allows to determine the list of available hosts, detect open ports on them and identify running services;

– making assumptions about the presence of vulnerabilities in the software, in particular in the detected services, based on the vulnerabilities knowledge base, errors in the configuration of equipment and other gaps in the protection perimeter of the information infrastructure;

– verification and confirmation of the possibility of implementing identified vulnerabilities by attempting to exploit them using specialized software, in particular, using so-called vulnerability exploits (malicious scripts, executable modules, etc.). This process is schematically shown in **Fig. 4.1**;

– generation of a report, which necessarily includes a list of validated vulnerabilities and their level of criticality (danger), as well as, optionally, recommendations for eliminating these vulnerabilities.

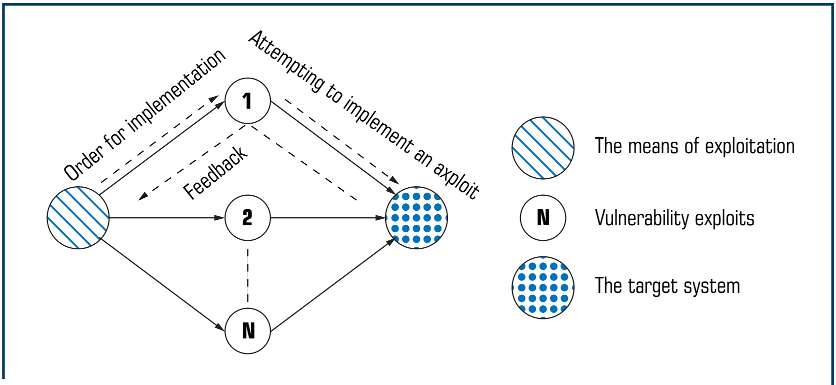

Fig. 4.1 The process of selecting and implementing the next exploit, with the subsequent recall of the target

In general, the need to check the possibility of implementing detected vulnerabilities, that is, their validation, arises because security scanners allow detecting only potential vulnerabilities of target systems, while allowing for the fallacy of such activations, which consists in the impossibility of actual implementation of the detected vulnerability on the part of the attacker. And since each SASA, having its own databases of ready-to-use exploits of known vulnerabilities and algorithms for their automatic implementation, the validation of detected vulnerabilities is carried out differently. So, for example, such algorithms can be based on the sequential implementation of all exploits available in the database, or taking into account simple criteria, such as the family of the operating system, the service and the rank of the exploit.

Thus, even in automated SASA, the number of vulnerability checks of only one target system, that is, attempts to exploit them, can reach several thousand (the main limitation is the number of exploits in the database). This can be critical from the point of view of the time required to conduct an active security analysis in corporate networks, moreover, the sequential launch of all available exploits significantly increases the risk of a critical error in the functioning of the target system and its complete failure. Accordingly, in view of these limitations, it becomes expedient to solve the problem of determining the quality of the mechanism for validating vulnerabilities of software and hardware platforms.

For this, first of all, by processing the results of a large number of observations of the functioning of the means of exploiting the identified vulnerabilities, in particular, the feedback scheme shown in **Fig. 4.2**, the general characteristics of the vulnerability validation process, which take into account the above factors, were highlighted.

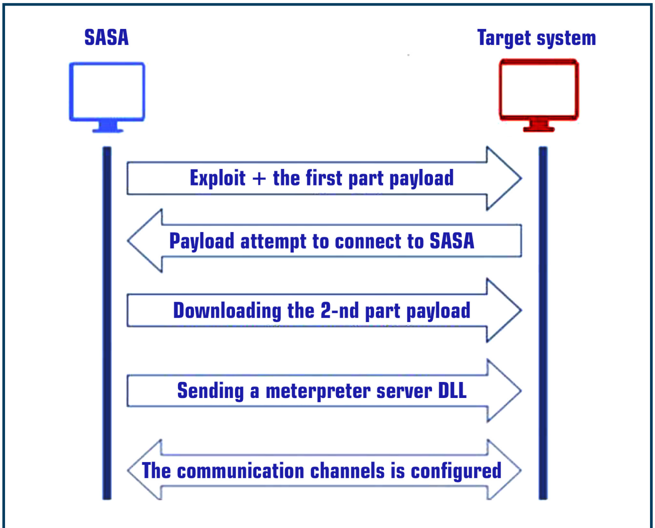
SASA
Target system
Exploit + the first part payload
Payload attempt to connect to SASA
Downloading the 2-nd part payload
Sending a meterpreter server DLL
The communication channels is configured

Fig. 4.2 Generalized scheme of step-by-step loading of the command interpreter – "meterpreter"

The feedback scheme (**Fig. 4.2**) is demonstrated on the example of the step-by-step implementation of the exploit with the previously mentioned payload, namely, the meterpreter command interpreter. At the same time, interaction (connection establishment) at the transport level takes place in accordance with the TCP protocol (**Fig. 4.3**) [84].

At the first stage, the exploit is transmitted to the target system along with the backed-up first part of the payload. After exploiting the vulnerability, the payload tries to connect back to the active security analysis system (i.e., the exploit tool, in this case metasploit) and establish a communication channel.

The next stage involves loading the second part of the payload DLL (Dynamic-link library – dynamically linked library) of the injection, after its successful execution, the exploit tool sends the DLL to the meterpreter server to establish the proper communication channel.

Thus, if the selected exploit, as well as the corresponding payload, worked and an active access session to the target system was obtained, it is possible to speak of a successful validation of the vulnerability.

Otherwise, when the selected exploit did not work, the vulnerability is not validated. If the exploit tool did not receive a response from the target system within the specified RTD delay interval and lost communication with it, the attempt to launch the selected exploit was unsuccessful and resulted in a critical error in the target system.

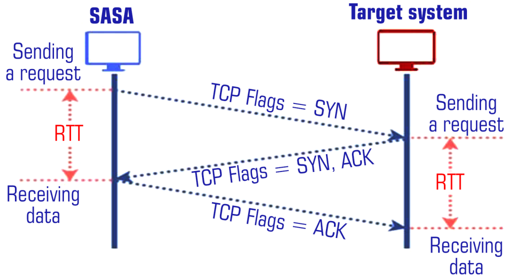

Fig. 4.3 The scheme of using the socket interface to establish a TSR connection

Thus, it was established that the quality of host vulnerability validation of the target corporate network is determined by the vector (q_s, q_f, q_c) of the three-dimensional vector space [96], where q_s – abscissa, which defines the number of successfully validated vulnerabilities, q_f – ordinate, which defines the number of unvalidated vulnerabilities and q_c – an application that determines the number of cases of vulnerability validation that resulted in critical errors on the target host and subsequent loss of communication with it.

However, given the risk of causing critical errors in the functioning of the target systems, their number in the corporate network, the dynamics of changes in the target systems themselves (changes in their configurations), as well as the constant increase in the number of new vulnerabilities and their exploits, it becomes difficult, experimentally, to ensure a full check of all objects (in this case, attempts to exploit vulnerabilities) related to this problem, in order to obtain a general population.

Therefore, it was decided to use a sample population in order to search for and study regularities in the process of active analysis of the security of corporate networks.

And since the sample is a subset of the general population, on the basis of its research using the tools and methods of mathematical statistics, it will be possible to draw a conclusion about the properties of the validation process that occurs in the general population.

At the same time, in order for the conclusions regarding the properties of the general population, which are made during the study of the sample, to be justified, it is necessary that this sample population is necessarily representative, that is, it accurately reflects the general population.

In general, the issue of forming a sample population is the first step on the way to obtaining results that objectively reflect the processes and phenomena that take place in the general population. And since the most common practice is to first select the required number of objects, and only then conduct their research, a list of 11 target systems was formed.

At the same time, the platforms were chosen on the basis of statistical data from Netmarketshare and Statcounter companies (**Fig. 4.4**) regarding the prevalence of the use of specific operating systems in the world [77–89] and in particular in Ukraine [89]. Also, two specially designed platforms for conducting penetration testing with known vulnerabilities already present were included in the list.

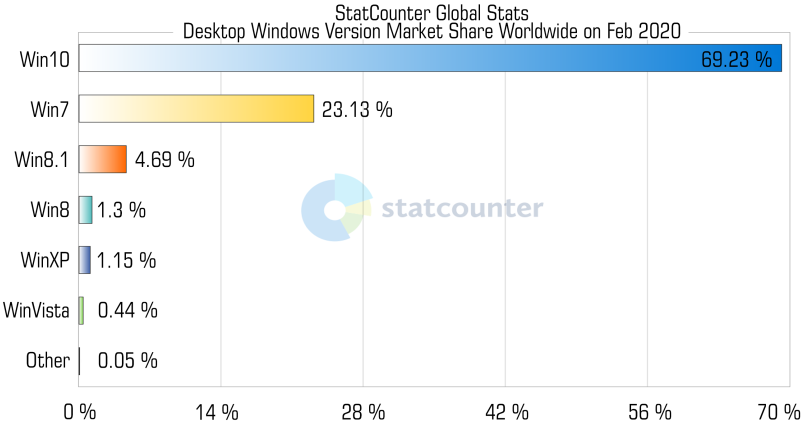

Fig. 4.4 Statistical data on the use of operating systems in the world

The experiment itself, in order to obtain the so-called functional dependencies, was carried out on a specially developed test bench, according to the proposed methodology of experimental research on the functioning of modern automated means of exploiting vulnerabilities, and the results were designed and presented in the form of a table (**Table 4.1**).

⬢ **Table 4.1** Results of vulnerability validation using armitage and autopwn

Platform (OS)	**Armitage**					**Metasploit-autopwn**				
	£	q_s	q_f	q_c	t	**£**	q_s	q_f	q_c	t
Windows XP SP2	312	3	306	3	345	63	3	58	2	244
Windows XP SP3	98	3	93	2	86	58	3	53	2	286
Windows 7	85	2	80	3	65	63	3	60	2	369
Windows 8.1	83	1	81	1	58	65	0	64	1	281
Windows 10	84	0	83	1	154	1255	0	1255	0	1523
Windows Server 2008 R2	96	2	92	2	82	84	1	82	1	363
Windows Server 2016	39	0	39	0	71	32	0	32	0	43
Mac OS X 10.13	63	1	61	1	115	59	1	58	0	249
Mac OS X 10.14	46	1	45	0	83	41	1	40	0	58
Metasploitable 2	765	3	762	0	293	1445	3	1442	0	1462
Metasploitable 3	780	3	777	0	330	1911	3	1908	0	1933

where £ – the total number of attempts to exploit detected vulnerabilities of a separate host of the target corporate network; t – total validation time of detected vulnerabilities of a separate host of the target corporate network, expressed in seconds

4.1.1 TEST STAND DESCRIPTION

All experiments were performed on a machine running Windows 10 Pro x64 v1803 operating system with an Intel Core i5-3210M CPU 2.50 GHz and 12 GB RAM using the VMware Workstation 12 Pro v12.5.9 build-7535481 virtualization platform, on which a special test stand was deployed. The schematic representation of this stand is presented in **Fig. 4.5**.

Virtual machines running the Kali GNU / Linux Rolling 2019.3 OS with the following tools for automating the work of the Metasploit framework vulnerability exploitation tool is installed and configured as a system of automatic active security analysis (SAASA) in various experiments:

– metasploit-autopwn;

– armitage.

As a target host, a number of virtual machines with different installed platforms and the corresponding standard set of software act:

– MS Windows 10;

– MS Windows Server 2008 R2;

– MS Windows Server 2016;

– Mac OS X 10.13 and 10.14;

– Metasploitable 2 and 3.

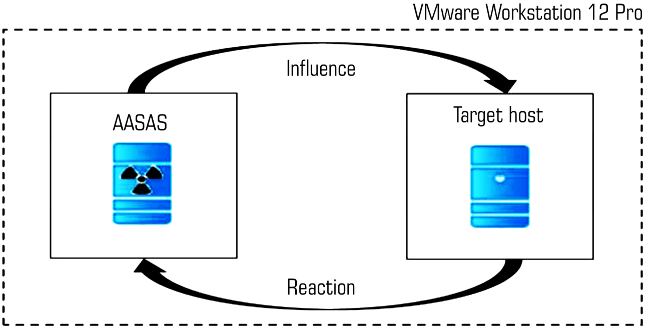

Fig. 4.5 Generalized scheme of the test stand

4.1.2 METHODOLOGY OF EXPERIMENTAL STUDY OF THE FUNCTIONING OF MODERN AUTOMATED MEANS OF EXPLOITING VULNERABILITIES

First of all, it should be noted that all experiments consist in conducting automatic active security analysis of a number of the same target hosts using various automated software tools of active security analysis defined in the previous subsection and further analysis of their work results.

The purpose of this experimental study is to determine the general characteristics of the vulnerability validation process. For this purpose, the following method of experimental research is proposed, where the following system of actions is provided:

1. After deployment of the test bench and configuration of all target hosts, create snapshots (VMware snapshot) [97–112] of data of virtual machines to save their original (initial) state. A virtual machine snapshot is a point-in-time copy of the virtual machine disk file (VMDK) that allows to restore the saved state of the virtual machine.

2. Conduct an analysis of the security of the next host using the Armitage graphical cyber attack management tool using the Hail Mary vulnerability exploitation mode, save the results and restore the initial state of the target host under investigation with VMware tools.

3. Carry out an analysis of the security of the next host using the db_autopwn automatic exploit and cyber attack plugin, save the results and restore the initial state of the target host under investigation with VMware tools.

4. If a critical error occurs in steps 2 and 3 during active security analysis, restore the initial state of the target host under investigation, and re-analyze it with exclusion from the list of exploits that led to this error.

5. Submit the results of the conducted experiments in the form of a table.

4.2 MATHEMATICAL MODELING OF INFORMATION SYSTEMS AND NETWORKS IDENTIFIED VULNERABILITIES VALIDATION MECHANISM

4.2.1 REGRESSION ANALYSIS OF EXPERIMENTAL RESEARCH RESULTS

On the basis of the conducted experimental studies, it was established that each of the coordinates of the vector on the one hand, it changes continuously over time, during which an active analysis of the security of an individual target host and the corporate network as a whole is carried out, and on the other hand, all three coordinates are connected to each other by some functional dependence.

However, unlike deterministic dynamic systems, which can be described by systems of differential equations built on the basis of the system`s nature, the task of detecting vulnerability validation is not unambiguous. Therefore, it was decided to solve the task of building a mathematical model of information systems and networks vulnerabilities validation mechanism by means of regression analysis [78, 87, 91], creating analytical dependencies, which in turn are solutions of some differential equations system.

In general, regression analysis refers to the study of the regularity of the relationship between two variables, when one *x* value corresponds to a set of *y* values, i.e. the relationship between them is not fully defined [88].

Thus, in regression analysis, statistical dependencies are described by a mathematical model, that is, a regression equation that reproduces the relationship between factor values and variable characteristics of the investigated process establishing the corresponding analytical dependence, and has the form $y = f(x)$.

At the same time, the regression equation, if possible, should be quite simple and adequate.

The analysis itself is carried out directly in several steps:

– checking for the presence of a correlation relationship;

– approximation of experimental data;

– statistical analysis of regression equations.

First of all, the statistical relationship between two variables is evaluated based on the results of experimental observations using the correlation coefficient.

Provided that *N* observations are made, resulting in two samples:

$$x_1, x_2, \ldots, x_n, \;\; y_1, y_2, \ldots, y_n,$$

the correlation coefficient is determined by the following formula:

$$R = \frac{\sum_{i=1}^{n} (x_i - \tilde{x})(y_i - \tilde{y})}{(n-1)\sigma_x \sigma_y}, \tag{4.1}$$

where $\tilde{x}$, $\tilde{y}$ – the mean value of the sample *X* and *Y*, respectively, which establishes the center of the sample population and is determined by formulas:

$$\tilde{x} = \frac{1}{n} \cdot \sum_{i=1}^{k} x_i, \quad \tilde{y} = \frac{1}{n} \cdot \sum_{i=1}^{k} y_i, \tag{4.2}$$

σ_x, σ_y – the mean squared deviation for *X* and *Y*, respectively, is defined as the square root of the sample variance:

$$\sigma_x = \sqrt{D_x}, \quad \sigma_y = \sqrt{D_y}, \tag{4.3}$$

D_x, D_y – sample variance, which characterizes the variability of the values in the sample of *X* and *Y*, respectively, that is, the variation of observations, and is determined by the formulas:

$$D_x = \frac{1}{n-1} \cdot \sum_{i=1}^{k} \left(\tilde{x} - x_i\right)^2, \quad D_y = \frac{1}{n-1} \cdot \sum_{i=1}^{k} \left(\tilde{y} - y_i\right)^2. \tag{4.4}$$

The expression $(n-1)$ from formulas (4.4) is called the number of degrees of freedom. This number is equal to the number of independent values involved in determining any parameter of a statistical population. When determining the variance, one degree of freedom is spent on determining the average value [90].

It should be noted that the value of the correlation coefficient is always within the limits $-1 \le R \le 1$. At the same time, it characterizes only a linear relationship between random variables. That is, with a positive value of the coefficient, it can be assumed that when one value increases, the other also increases on average, and with a negative value, on the contrary, the growth of one value leads to a decrease in the other value on average. The closer the value *R* to $+1$ or -1, the closer the linear relationship between the *x* and *y* values, however, if the value $R = 0$, this indicates its absence. In general, a satisfactory value of relationship density is considered to be $R \ge 0{,}5$, good at $R = 0{,}8 \ldots 0{,}85$.

Verification of the correspondence of the sample value of the correlation coefficient *R* to the correlation value (ρ) between general populations *x* and *y*, occurs with the use of *t* – distribution of Student [112]. For this, the calculated value t_{est} is first found according to formula (4.5) and compared with the table (**Table 4.2**).

$$t_{est} = |R| \sqrt{\frac{n-2}{1-R^2}}. \tag{4.5}$$

If $t_{est} > t_{tab}$ with the number of degrees of freedom $f = n-2$ and significance level $\alpha = 5\ \%$, then the correlation relationship exists and is confirmed for general populations.

⬢ **Table 4.2** Student's test value for significance $\alpha=0.05$

f	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**60**
t_{tab}	12,71	4,303	3,182	2,775	4,571	2,447	2,305	2,228	2,086	2,042	2,00

The next step, in order to present in an understandable and concise form the empirical dependencies between the parameters describing the behavior of the system, is the approximation of the experimental data.

In general, approximation is the process of constructing an approximate (approximating) function based on the results of experimental studies that passes through all points of the initial data and is closest to a given continuous function. The process itself consists of two main stages [87]:

1) the selection of the general form of a typical functional dependence (approximants), which is carried out either for theoretical reasons or with the help of a graphical representation of the results of the experiment, analyzing the location of the points (x_n, y_n) on the Cartesian coordinate plane. At the same time, the values of the factor are placed on the abscissa axis, respectively, the values of the evaluation parameter are placed on the ordinate axis, and the actual results are indicated by dots. By directly connecting these points with a straight line, let's obtain a graph of the results of the conducted experiment, and by drawing another straight line (curve) through the middle points of each of the obtained segments, there is an approximate representation of the graph of the desired approximant. After that, the obtained graph is compared with the graphs of typical functions and the general appearance of the approximating function is selected, which will most similarly describe the investigated dependence;

2) determination of the best numerical values of the parameters (coefficients) of the approximant.

4.2.2 MATHEMATICAL METHODS OF FUNCTION APPROXIMATION

In general, the need to approximate a function by some analytical model arises when solving a fairly wide range of tasks, in particular tasks of statistical radio engineering and radio physics, control theory and data transmission systems [105, 112–114].

For example, approximation tasks are solved when modeling acoustic signals and designing telecommunication systems [103, 104], mobile communication systems [102], when optimizing the parameters of the reversible (lossless) digital data compression procedure [113], as well as when creating mathematical models of control systems [31] and simulation of processes under the influence of random factors [80].

All methods of function approximation can be divided into two groups: parametric and non-parametric. The first use a priori information about the type and / or parameters of the general distribution. Non-parametric ones, in turn, work with greater uncertainty regarding a priori information, including even its complete absence, and therefore have a much wider scope of application.

However, non-parametric methods, in comparison with parametric ones, are more time-consuming from the point of view of performing mathematical calculations.

It should be noted that currently, one of the main approaches to solving nonparametric approximation problems is polynomial estimation based on the first theorem of *K*. Weierstrass, which is as follows: if the function $f\left(x\right)$ is continuous on a segment $\left[a,b\right]$, then there exists a sequence of polynomials $\left\{Pn\left(x\right)\right\}$, which uniformly on the segment $\left[a,b\right]$ converges to $f\left(x\right)$, that is, for any $\varepsilon > 0$ polynomial will be found $P_n(x)$ with the number *n* depending on ε, such that $\left|P_n\left(x\right)-f\left(x\right)\right|<\varepsilon,$ at once for all *x* from the interval $\left[a,b\right]$.

This theorem was proved in 1912 by the famous Soviet scientist S. N. Bernshtein [4, 5]. As approximating polynomials $P_n(x)$, polynomials of the following form were used:

$$B_n\left(f;x\right)=B_n\left(x\right)=\sum_{k=0}^{n}f\left(\frac{k}{n}\right)b_{k,n}\left(x\right), \tag{4.6}$$

where $b_{k,n}\left(x\right)=C_n^k x^k\left(1-x\right)^{n-k}$, $C_n^k = n!/\,k!\left(n-k\right)!.$

Function $b_{k,n}\left(x\right)$ are called the basis Bernstein polynomial of degree *n*, operators $B_n\left(f;x\right)$ respectively, Bernstein polynomials of order *n* functions $f\left(x\right)$, and the coefficients $f(k\,/\,n)$ – Bernstein coefficients.

S.N. Bernstein, relying on elementary results from the theory of probabilities, proved that the sequence of polynomials $\left\{B_n\left(f;x\right)\right\}$ at $n\to\infty$ converges to $f\left(x\right)$ uniformly on $\left[0,1\right]$, that is

$$\lim_{n\to\infty}\left\|f-B_n\left(f\right)\right\|=0.$$

Thus, the following theorem holds.

Theorem 2.1. If the function $f\left(x\right)$ on a segment $\left[0,1\right]$ satisfies the Lipshitz condition [31] with a constant M, then with every $n\geq 2$ and every $x\in\left[0,1\right]$, a fair estimate

$$\left|B_n\left(f;x\right)-f\left(x\right)\right|\leq M\sqrt{\frac{x\left(1-x\right)}{n}}. \tag{4.7}$$

4.2.3 MATHEMATICAL MODEL OF ANALYSIS OF VULNERABILITY VALIDATION PROCESS QUANTITATIVE CHARACTERISTICS

Based on the results of an experimental study of the functioning of modern automated means of exploiting vulnerabilities obtained in 4.1 (**Table 4.1**), let's build a mathematical model for the analysis of quantitative characteristics of the process of validating information system vulnerabilities by means of regression analysis. To do this, let's first estimate the statistical relationship between variables t and q_s, q_f, q_c, obtained during the study of the validation mechanism of the Armitage cyber-attack management graphic tool, using the correlation coefficient (4.1).

According to the data in **Table 4.1**, let's calculate the auxiliary values: the average value of the sample (4.2), the sample variance (4.4) and the mean squared deviation (1.3). Let's summarize the results in **Table 4.3**.

● **Table 4.3** Estimated values of the correlation coefficient

Searched values	**Armitage**		
	t, q_s	**t, q_f**	**t, q_c**
$\tilde{t}$	152,91	152,91	152,91
$\tilde{q}$	1,73	219,91	1,18
D_t	12716,09	12716,09	12716,09
D_q	1,42	79027,89	1,36
σ_t	112,77	112,77	112,77
σ_q	1,19	281,12	1,17
R	0,6	0,84	-0,07

In addition, let's also check the correspondence of the sample value of the correlation coefficient R to the correlation value (ρ) between general aggregates of quantities t and q_s, q_f, q_c using Student's distribution. Let's find the value t_{est} by formula (4.5). The results are presented in **Table 4.4**.

● **Table 4.4** Criterion of the correlation coefficient significance

Calculated value	**Armitage**		
	t, q_s	**t, q_f**	**t, q_c**
t_{est}	2,254	4,693	0,216

Comparing the obtained values t_{est} with theoretical ones (**Table 4.1**) with the number of degrees of freedom $f = n - 2 = 9$ and significance level $\alpha = 5\%$ there are the following results:

– for a pair t, q_s – $t_{est} > t_{tab}$ since $2{,}254 > 2{,}096$, this indicates that there is a direct relationship between the time of validation of detected vulnerabilities and the number of successfully validated vulnerabilities;

– for a pair t, q_f – $t_{est} > t_{tab}$ since $4{,}693 > 2{,}096$, this indicates that there is a significant direct relationship between the time of validation of detected vulnerabilities and the number of unvalidated vulnerabilities;

– for a pair t, q_c – $t_{est} < t_{tab}$ since $0{,}216 < 2{,}096$, this indicates that there is a very weak relationship between the time of validation of detected vulnerabilities and the number of cases of validation of vulnerabilities that lead to critical errors on the target host, however, based on previous results, it can be argued that this situation is solved by increasing the number of observations.

Next, in order to present the empirical dependences between the parameters in a clear and concise form, let's approximate the experimental data. To do this, first, let's find out the class of functions to which the desired approximant belongs by constructing a graph of the experiment results and an approximate graph of the desired approximant for each of the pairs of variables (**Fig. 4.6**).

Fig. 4.6 shows that the functions have a polynomial representation and, at the same time, a value R^2 (reliability of the approximation) testify to the accuracy of the description of the initial dependence of the experimental data by the approximating function.

That is why, in order to obtain the most reliable coefficients of the approximant, let's use Bernstein's theorem.

From the data in **Table 4.1**, it can be seen that the time of the rational cycle of vulnerability validation, in the case of using the Armitage tool, is 345 seconds. Therefore, first it is possible to carry out normalization of the time segment $[0;345]$, as follows:

$$t_n = \frac{t_i}{T}, \tag{4.8}$$

where t_n – is the normalized time; T – target host vulnerability validation time in seconds (rational cycle time); t_i – the time for which the relevant characteristics (q_s, q_f, q_c) assumed their values within the rational cycle.

The results of normalization of the time segment are presented in **Table 4.5**.

Table 4.5 Normalization of the rational cycle time

real time – t	0	58	65	71	82	83	86	115	154	293	330	345	0
normalized time – t_n	0	0,168	0,188	0,206	0,238	0,241	0,249	0,333	0,446	0,849	0,957	1	0

Then the values of the variables $q_s(t_n)$, $q_f(t_n)$, $q_c(t_n)$, as functions of normalization time, presented in **Table 4.6**.

After that, using data from **Table 4.6** and representation (4.6), initial analytical dependencies for the number of successfully validated vulnerabilities were obtained $q_s = q_s(t_n)$.

$$q_s(t_n)=q_s(0)b_{0.11}(t_n)+q_s(0{,}168)b_{1.11}(t_n)+q_s(0{,}188)b_{2.11}(t_n)+q_s(0{,}206)b_{3.11}(t_n)+$$
$$+q_s(0{,}238)b_{4.11}(t_n)+q_s(0{,}241)b_{5.11}(t_n)+q_s(0{,}249)b_{6.11}(t_n)+q_s(0{,}333)b_{7.11}(t_n)+$$
$$+q_s(0{,}446)b_{8.11}(t_n)+q_s(0{,}849)b_{9.11}(t_n)+q_s(0{,}957)b_{10.11}(t_n)+q_s(1)b_{11.11}(t_n).$$

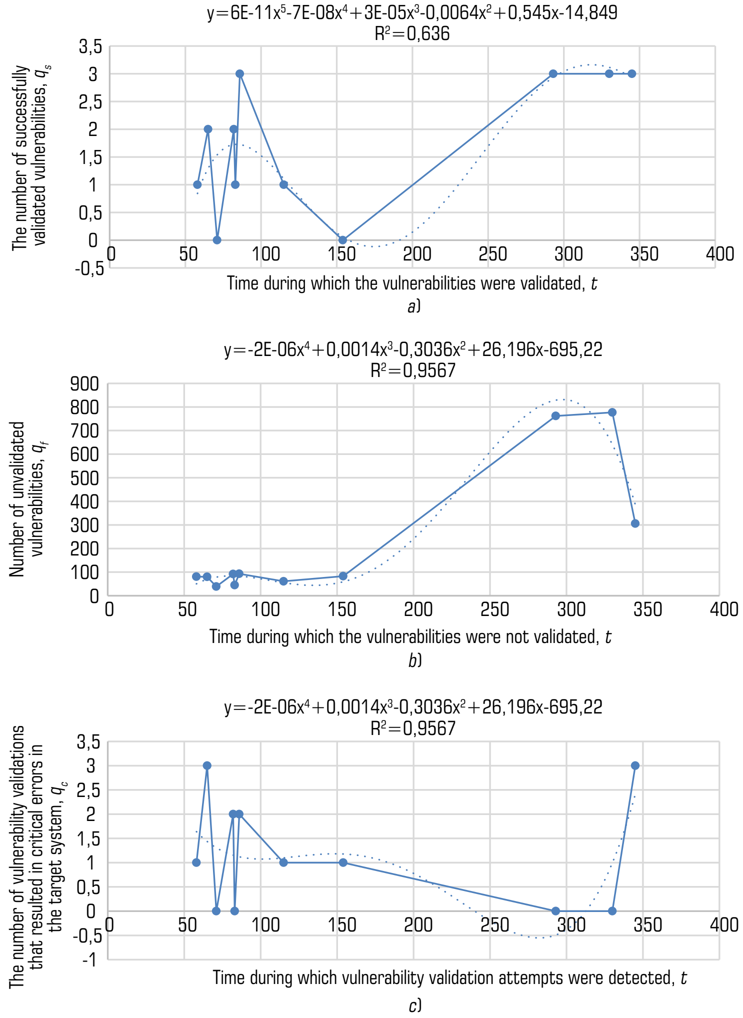

Fig. 4.6 Approximate graphs of the sought approximants for each of the pairs of variables: a – t, q_s; b – t, q_f; c – t, q_c

Table 4.6 Value of number of successfully validated $q_s(t_n)$, unvalidated vulnerabilities $q_f(t_n)$ and cases of validations that led to critical errors $q_c(t_n)$

t_n – normalized time	0	0,168	0,188	0,206	0,238	0,241	0,249	0,333	0,446	0,849	0,957	1	0
$q_s(t_n)$	0	1	2	0	2	1	3	1	0	3	3	3	0
$q_f(t_n)$	0	81	80	39	92	45	93	61	83	762	777	306	0
$q_c(t_n)$	0	1	3	0	2	0	2	1	1	0	0	3	0

Table 4.7 The value of polynomials $b_{k,11}(t_n)$

k	$b_{k,11}(t_n)$
0	$(1-t)^{11}$
1	$11t(1-t)^{10}$
2	$55t^2(1-t)^9$
3	$165t^3(1-t)^8$
4	$330t^4(1-t)^7$
5	$462t^5(1-t)^6$
6	$462t^6(1-t)^5$
7	$330t^7(1-t)^4$
8	$165t^8(1-t)^3$
9	$55t^9(1-t)^2$
10	$11t^{10}(1-t)$
11	t^{11}

After substituting the corresponding values from **Tables 4.6** and **4.7**, simplifying the expression, there is

$$q_s(t_n) = b_{1.11}(t_n) + 2b_{2.11}(t_n) + 2b_{4.11}(t_n) + b_{5.11}(t_n) + 3b_{6.11}(t_n) + b_{7.11}(t_n) + \\ +3b_{9.11}(t_n) + 3b_{10.11}(t_n) + 3b_{11.11}(t_n). \quad (4.9)$$

After comparing the values of the calculation results and the data from **Table 4.6**, it follows (**Table 4.8**) that the deviations between the empirical and calculated data are permissible, and when the number of values increases, these deviations become smaller and smaller. At the same time, it should be noted that for further research related to false vulnerability validation attempts and validation cases that led to critical errors, this difference is not significant.

⬢ **Table 4.8** Comparative values for $q_s(t_n)$

t_n – normalized time	Empirical values $q^e_s(t_n)$	Calculated values $q^p_s(t_n)$	Deviation $\theta = \lvert q^e_s(t_n) - q^p_s(t_n) \rvert$
0	0	0	0
0,168	1	1,065446	0,065446
0,188	2	1,100162	0,899838
0,206	0	1,126111	1,126111
0,238	2	1,167208	0,832792
0,241	1	1,171013	0,171013
0,249	3	1,181262	1,818738
0,333	1	1,309026	0,309026
0,446	0	1,494756	1,494756
0,849	3	2,425641	0,574359
0,957	3	2,970647	0,029353
1	3	3	0

The dependence graph (4.9) is presented in **Fig. 4.7**, which shows that the function $q_s = q_s(t_n)$ of successful validation of vulnerabilities satisfies the Lipshitz condition [31], i.e., for arbitrary $t_n^{(1)}, t_n^{(2)} \in [0;1]$ exist $K > 0$, that the inequality is fulfilled

$$\left| q_s(t_n^{(1)}) - q_s(t_n^{(2)}) \right| \le K \left| t_n^{(1)} - t_n^{(2)} \right|. \tag{4.10}$$

It follows from the condition (1.10) that there is a rectangular region beyond which the graph of the function $q_s = q_s(t_n)$ does not step out.

This makes it possible to further build the laws of probability distribution of the number of successfully validated vulnerabilities. In addition, when condition (4.10) is fulfilled, estimate (4.7) is valid, i.e.

$$\left| B_n(q_s, t_n) - q_s(t_n) \right| \le K \sqrt{\frac{t_n(1 - t_n)}{n}}. \tag{4.11}$$

It follows from the inequality (4.7) that there exists such a positive number K, at which

$$\theta = \left| q_s^e(t_n) - q_s^p(t_n) \right| = K\sqrt{\frac{t_n(1-t_n)}{n}}. \tag{4.12}$$

Dependence (4.12) makes it possible to set the appropriate precision for determining the power n of the Bernstein polynomial.

Thus, using data from **Table 4.8** and dependence (4.12), the maximum value was obtained K for $q_s = q_s(t_n)$:

$\max(k_i) = 13{,}949121$, where $i \in [1;11)$.

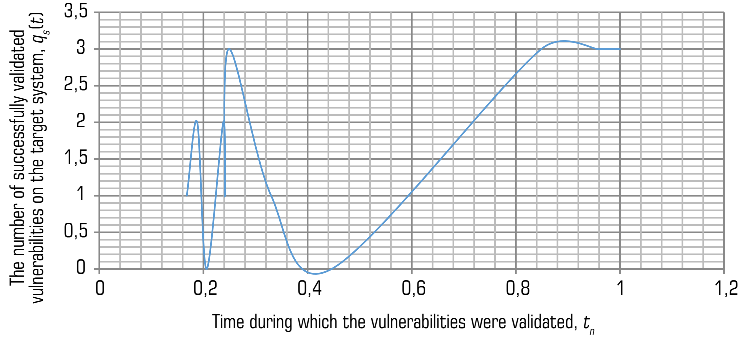

Fig. 4.7 The target system on the time of the rational cycle

Similarly, using representation (4.6), data from **Tables 4.6** and **4.7**, let's obtain initial analytical dependencies for the number of unvalidated vulnerabilities $q_f = q_f(t_n)$ (dependency (4.13), **Fig. 4.8**) and the number of cases of vulnerability validation that led to critical errors $q_c = q_c(t_n)$ (dependency (4.14), **Fig. 4.9**).

Tables 4.9 and **4.10** present the corresponding comparative values of the calculation results and data from **Table 4.6**.

$$\begin{aligned} q_f(t_n) &= 81b_{1.11}(t_n) + 80b_{2.11}(t_n) + 39b_{3.11}(t_n) + 92b_{4.11}(t_n) + 45b_{5.11}(t_n) + 93b_{6.11}(t_n) + \\ &+ 61b_{7.11}(t_n) + 83b_{8.11}(t_n) + 762b_{9.11}(t_n) + 777b_{10.11}(t_n) + 306b_{11.11}(t_n). \end{aligned} \tag{4.13}$$

$$q_c(t_n) = b_{1.11}(t_n) + 3b_{2.11}(t_n) + 2b_{4.11}(t_n) + 2b_{6.11}(t_n) + b_{7.11}(t_n) + b_{8.11}(t_n) + 3b_{11.11}(t_n). \tag{4.14}$$

Also, it should be noted that **Fig. 4.8** and **4.9** shows that the functions $q_f=q_f(t_n)$ and $q_c=q_c(t_n)$ also satisfy the Lipshitz condition.

Table 4.9 Comparative values for $q_f(t_n)$

t_n – normalized time	Empirical values $q^e_f(t_n)$	Calculated values $q^p_f(t_n)$	Deviation $\theta=\lvert q^e_f(t_n)-q^p_f(t_n)\rvert$
0	0	0	0
0,168	81	62,547827	18,45217
0,188	80	63,809242	16,19076
0,206	39	64,596778	25,596778
0,238	92	65,508850	26,49115
0,241	45	65,575300	20,5753
0,249	93	65,743882	27,256118
0,333	61	67,844585	6,844585
0,446	83	78,745219	4,254781
0,849	762	538,115125	223,884875
0,957	777	478,499059	298,500941
1	306	306	0

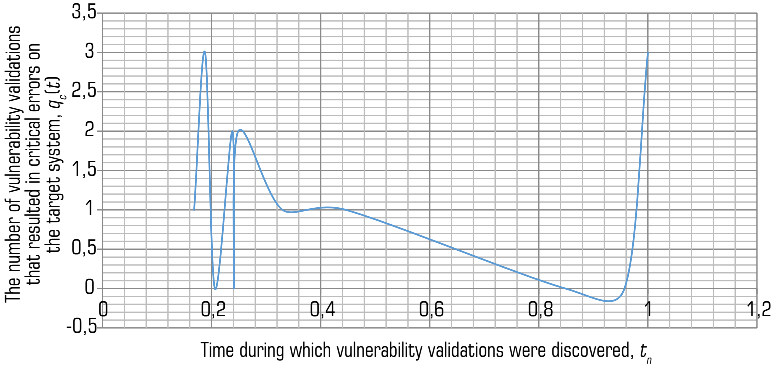

Fig. 4.8 Dependence of the number of unvalidated vulnerabilities of the target system on the time of the rational cycle

Table 4.10 Comparative values for $q_c(t_n)$

t_n – normalized time	Empirical values $q^e_f(t_n)$	Calculated values $q^p_f(t_n)$	Deviation $\theta = \lvert q^e_f(t_n) - q^p_f(t_n) \rvert$
0	0	0	0
0,168	1	1,337389	0,337389
0,188	3	1,360285	1,639715
0,206	0	1,364959	1,364959
0,238	2	1,346917	0,653083
0,241	0	1,343984	1,343984
0,249	2	1,335418	0,664582
0,333	1	1,221982	0,221982
0,446	1	1,125939	0,125939
0,849	0	0,731249	0,731249
0,957	0	1,860081	1,860081
1	3	3	0

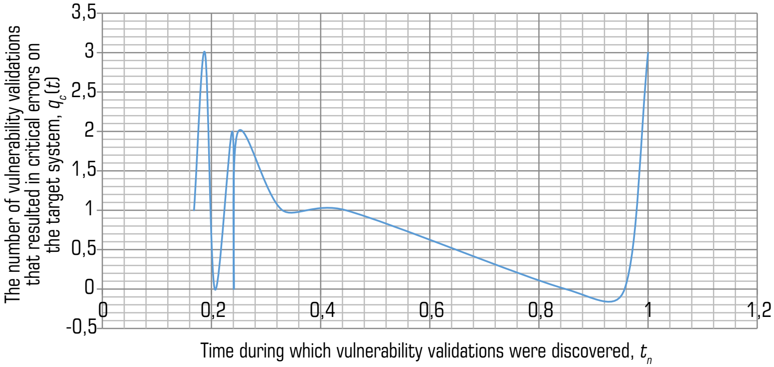

Fig. 4.9 Dependence of the number of validations of vulnerabilities that led to critical errors in the target system on the time of the rational cycle

In addition, using data from **Tables 4.9** and **4.10** and dependence (4.12), let's obtain the maximum K values for $q_f = q_f(t_n)$:

max(K_i)=4880,359905, where $i \in [1;11)$,

and $q_c = q_c(t_n)$

max(K_i)=30,411511, where $i \in [1;11)$.

Thus, as a result, analytical dependencies were obtained for the studied characteristics of the process of validation of vulnerabilities of information systems [19]:

$$q_s(t_n) = \sum_{i=0}^{n} q_s(t_n^{(i)}) b_{k,n}(t_n),\ q_f(t_n) = \sum_{i=0}^{n} q_f(t_n^{(i)}) b_{k,n}(t_n),\ q_c(t_n) = \sum_{i=0}^{n} q_c(t_n^{(i)}) b_{k,n}(t_n). \quad (4.15)$$

4.3 METHODOLOGY FOR ANALYZING THE QUALITY OF WORK OF THE MECHANISM FOR CORPORATE NETWORK DETECTED VULNERABILITIES VALIDATING

Based on the practical analysis of the vulnerability validation process carried out in the previous section and the analytical dependencies of the basic characteristics of the vulnerability validation process (4.15) obtained with the help of Bernstein polynomials, it became possible to highlight and characterize additional key indicators that will allow more precisely determining the quality of the vulnerability validation mechanism, and also assert with high credibility about the positive progress or consequences of validating the vulnerabilities of the target corporate network.

As a result, the following quality indicators of the corporate network vulnerability validation mechanism were selected [95, 112–116]:

1) *A* – *accuracy* – the share of correctly made decisions regarding the implementation of specific exploits relative to all made decisions. This parameter characterizes the ability of the validation mechanism of detected vulnerabilities to successfully check and confirm the possibility of their implementation due to correctly made decisions regarding the use of selected exploits with the appropriate payload for these vulnerabilities;

2) *E* – *error* – the share of decisions made regarding the implementation of specific exploits that did not confirm the possibility of implementing the corresponding vulnerabilities in relation to all decisions made. The error parameter characterizes the ability of the mechanism of validation of detected vulnerabilities to make decisions regarding the use of selected exploits that do not work for certain reasons. Such reasons include, for example, the non-compliance of the target system with the conditions for implementing the selected exploit, changing the ports on which vulnerable services work by default, the reaction of the protection system – blocking the possibility of implementing the exploit;

3) *Ce* – *critical error* – the share of decision-making cases regarding the implementation of specific exploits, which led to critical errors in the target system and subsequent loss of

communication with it in relation to all decisions made. A critical error characterizes the ability of the mechanism of validation of detected vulnerabilities to make decisions regarding the use of selected exploits, which in the process of their implementation lead to a critical error in the functioning of the target system and its subsequent failure.

According to (4.15), these indicators are determined as follows:

$$A=\frac{\int_0^1 q_s(\theta)d\theta}{\int_0^1 (q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}, \tag{4.16}$$

$$E=\frac{\int_0^1 q_f(\theta)d\theta}{\int_0^1 (q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}, \tag{4.17}$$

$$Ce=\frac{\int_0^1 q_c(\theta)d\theta}{\int_0^1 (q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}. \tag{4.18}$$

In addition, in order to evaluate the quality of the mechanism for validating detected vulnerabilities, taking into account all the above quality indicators, let's reduce expressions (4.16)–(4.18) into a single integral indicator:

$$J_{qv}=\frac{A}{E}-Ce, \tag{4.19}$$

where J_{qv} – integral index of the vulnerability validation mechanism quality.

At the same time, if $J_{qv}>1$, then the vulnerability validation mechanism has high quality.

Thus, it is possible to highlight the following steps of the methodology of quality analysis of the mechanism of validation of corporate network vulnerabilities [98]:

Step 1. Collection of statistical data regarding the process of validation of detected vulnerabilities of the corporate network of the evaluated validation mechanism.

Step 2. Normalization of the time segment of the vulnerability validation of the hosts of the target corporate network according to the expression (4.8).

Step 3. Construction of Bernstein polynomials to obtain initial analytical dependencies for basic characteristics (q_s, q_f, q_c) of the vulnerability validation quality.

Step 4. Calculation of more accurate performance indicators of the vulnerability validation mechanism: *A* – accuracy (4.16), *E* – error (4.17) and *Ce* – critical error (4.18).

Step 5. Evaluation of the performance of the mechanism for validating vulnerabilities of corporate networks based on the calculation of a single integral indicator (4.19).

Also, it should be noted that the dependencies (4.16)–(4.18) are generally functions of time

$$A\left(t_n\right)=\frac{\int_0^{t_n} q_s(\theta)d\theta}{\int_0^{t_n}(q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}, \tag{4.20}$$

$$E\left(t_n\right)=\frac{\int_0^{t_n} q_f(\theta)d\theta}{\int_0^{t_n}(q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}, \tag{4.21}$$

$$Ce\left(t_n\right)=\frac{\int_0^{t_n} q_c(\theta)d\theta}{\int_0^{t_n}(q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}, \tag{4.22}$$

the graphs of which were constructed and displayed in **Fig. 4.10** on the basis of an experimental study of the mechanism of validation of information system vulnerabilities of the db_autopwn plugin (**Table 4.1**).

In addition, taking the derivatives of the distribution functions of the quantitative indicators of the quality of the mechanism of validation of detected vulnerabilities (4.20), (4.21) and (4.22), let's obtain the distribution densities, which are determined as follows:

$$\alpha\left(t_n\right)=\frac{dA\left(t_n\right)}{dt_n}, \tag{4.23}$$

$$\xi\left(t_n\right)=\frac{dE\left(t_n\right)}{dt_n}, \tag{4.24}$$

$$\varsigma\left(t_n\right)=\frac{dCe\left(t_n\right)}{dt_n}. \tag{4.25}$$

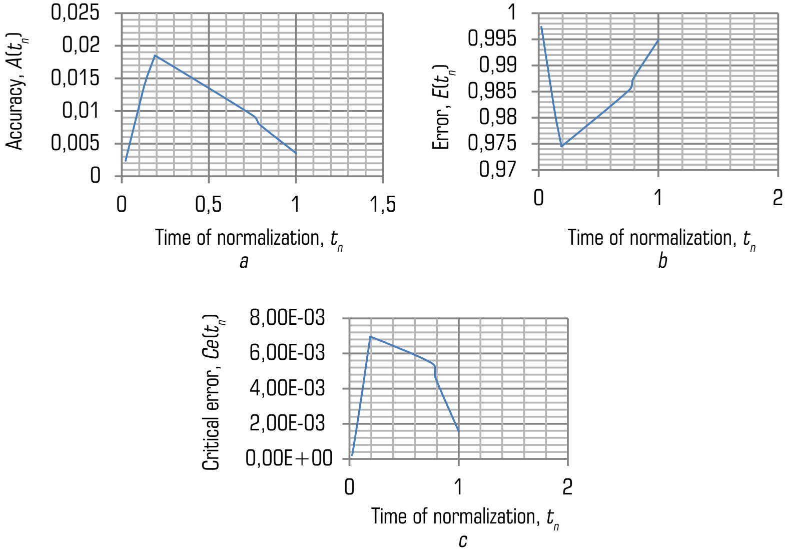

Fig. 4.10 Dependence of quantitative performance indicators of the vulnerability validation mechanism on the time of the rational cycle: *a* – accuracy $A(t_n)$; *b* – error $E(t_n)$; *c* – critical error $Ce(t_n)$

4.4 A METHOD OF CONSTRUCTING A FUZZY KNOWLEDGE BASE FOR DECISION-MAKING WHEN VALIDATING SOFTWARE AND HARDWARE PLATFORM VULNERABILITIES

Having analyzed the dependencies of the quality indicators of the corporate network vulnerability validation mechanism obtained in the previous subsection (**Fig. 4.10**), it can be seen that the maximum value of accuracy *A* takes the value 0,02 (a). At the same time, a minimal error *E* is within the limits from 0.97 to 0.98 (b), and the maximum critical error *Ce* is within the limits from $6\cdot10^{-3}$ to $8\cdot10^{-3}$ (c). This makes it possible to construct property functions for fuzzy sets, the elements of which are accuracy, error, and critical error.

Therefore, a decision was made to intellectualize the process of validating vulnerabilities of software and hardware platforms based on fuzzy technology [112], by creating a knowledge base for automatic decision-making when validating vulnerabilities during an active analysis of the security of corporate networks. This will make it possible to quickly, in real time, and with minimal risk, make appropriate decisions regarding attempts to implement specific exploits of vulnerabilities for their validation.

It should be noted that in order to build a knowledge base and further form decisive decision-making rules, first of all, it is necessary to select input and output parameters [80, 111].

Quantitative characteristics of the vulnerability validation process are used as input parameters, which include the number of successfully validated vulnerabilities $q_s(t)$, number of unvalidated vulnerabilities $q_f(t)$ and the number of instances of vulnerability validation that resulted in a critical error $q_c(t)$.

The output parameters are the distribution functions of quantitative indicators of the quality of the mechanism of validation of detected vulnerabilities (4.20)–(4.22).

As it was already established, from the conducted study of the vulnerability validation process, the initial parameters for building the knowledge base depend on the normalization time, which is a random variable that can take ambiguous values [112]. Based on this, with the use of statistical values obtained during the study of the mechanism of validation of information system vulnerabilities of the db_autopwn plugin (**Table 4.11**), the normalization time membership function was obtained (**Fig. 4.11**).

Table 4.11 The value of the number of successfully validated $q_s(t_n)$, unvalidated vulnerabilities $q_a(t_n)$ and cases of validations that led to critical errors $q_c(t_n)$

Normalized time – t_n	**0**	**0,022**	**0,03**	**0,126**	**0,129**	**0,145**	**0,148**	**0,188**	**0,191**	**0,756**	**0,788**	**1**	**0**
$q_s(t_n)$	0	0	1	3	1	0	3	1	3	3	0	3	0
$q_f(t_n)$	0	32	40	58	58	64	53	82	60	1442	1255	1908	0
$q_c(t_n)$	0	0	0	2	0	1	2	1	2	0	0	0	0

*Note. Compiled based on statistical data of an experimental study of the functioning of the Metasploit-autopwn automated tool for exploiting vulnerabilities (***Table 4.1***)*

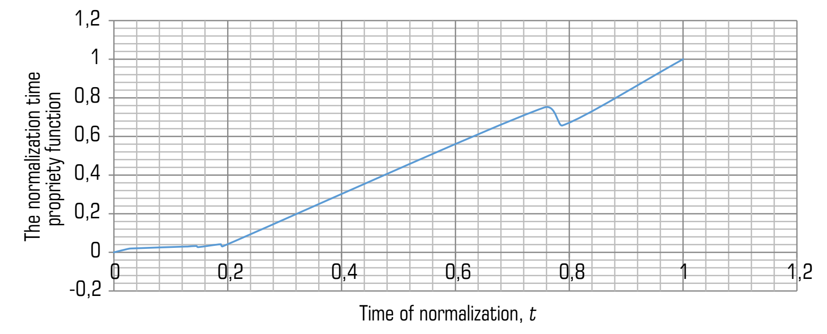

Fig. 4.11 Vulnerability validation analysis normalization time appropriateness function

Fig. 4.11 shows that the value of the membership function of the normalization time coincides with the error values *E*, which is determined by dependence (4.21). Thus, as the normalization time increases, the probability of an error in the implementation of specific exploits increases.

Each of the parameters determined by dependencies (4.20)–(4.22) has its own ranges of values independently of each other.

Therefore, there is a vague scale of the three values of these parameters, on the basis of which it is already possible to build a knowledge base for decision-making when validating corporate network vulnerabilities.

At the same time, a universal scale of linguistic variables (terms) was used for a vague scale of the quality of the vulnerability validation mechanism [80]:

$$T=\{Min, Low, Med, High, Max\}. \quad (4.26)$$

Thus, based on the results of two independent analyzes of the validation process of identified vulnerabilities, vague evaluations of the quality of the vulnerability validation mechanism were formed and described in **Tables 4.12** and **4.13**.

⬢ **Table 4.12** A knowledge base formed on the first experiment results using Armitage

A	***E***	***Ce***	Q_{mv}	**Description**
1	0	0	*Max*	A reliable vulnerability validation mechanism that does not disrupt the operation of target systems and does not allow wrong decisions regarding the use of exploits
[0,8;1)	(0:0,1)	(0:0,01]	*High*	The ability of the validation mechanism to successfully check and confirm the possibility of implementing vulnerabilities due to correctly made decisions regarding the use of selected exploits is high, with a fairly low number of wrong decisions
[0,5;0,8]	[0,1;0,35]	(0,01;0,2)	*Med*	The validation mechanism is quite stable
[0,1;0,5)	(0,35;0,7]	[0,2;0,5)	*Low*	For the most part, the vulnerability validation mechanism is inactive (inefficient) because it allows an unacceptable number of wrong decisions regarding the use of exploits
[0;0,1)	(0,7;1]	[0,5;1]	*Min*	The validation mechanism is unusable because it causes too many failures in the target systems

⬢ **Table 4.13** A knowledge base formed on the second experiment results using Metasploit-autopwn

A	*E*	*Ce*	Q_{mv}	**Description**
(0,8;1]	[0;0,1)	[0;0,1)	*Max*	A reliable vulnerability validation mechanism that practically does not lead to disruption of target systems, and also allows a minimum number of wrong decisions regarding the use of exploit
(0,7;0,8]	[0,1;0,2)	[0,1;0,15)	*High*	The ability of the validation mechanism to successfully check and confirm the possibility of implementing vulnerabilities due to correctly made decisions regarding the use of selected exploits is quite high, with a low number of false decisions
[0,5;0,7]	[0,2;0,3]	[0,15;0,2]	*Med*	The validation mechanism is quite stable, however, it allows correct validation of vulnerabilities in no more than 70 % of cases
[0,1;0,5)	(0,3;0,7]	(0,2;0,4]	*Low*	For the most part, the vulnerability validation mechanism is ineffective (inefficient) because it allows an unacceptable number of wrong decisions regarding the use of exploits
[0;0,1)	(0,7;1]	(0,4;1]	*Min*	The validation mechanism is not recommended for use because it leads to a large number of failures in the functioning of the target systems

It can be seen from both tables that when conducting two independent experiments with obtaining a large volume of statistical data, unclear estimates were formed Q_{mv} of the performance quality of the vulnerability validation mechanism, which is based on the introduction of set terms (2.10), does not differ significantly.

Based on this, and also taking into account the expression (4.27), according to which it was established that the knowledge base should contain 125 rules of logical inference of the form (4.28) [85, 92, 93], a generalized knowledge base was built, a fragment of which is presented in the form of **Table 4.14**.

$$N_{max}=l_1 \cdot l_2 \cdot \ldots \cdot l_n \tag{4.27}$$

where N_{max} – the number of terms for the evaluation of the i_{th} output variable $\left(i=\overline{1,n}\right)$; n – the number of output variables.

$$R(i): IF\left(A_i \in T \,\&\, E_i \in T \,\&\, Ce_i \in T\right) THEN\left(Q_{mv_i} \in T\right),\ i=\overline{1,k}. \tag{4.28}$$

Table 4.14 Knowledge base fragment

№	*A*	*E*	*Ce*	*Q_{mv}*	№	*A*	*E*	*Ce*	*Q_{mv}*
1.	*Max*	*Max*	*Max*	*Max*	7.	*Max*	*High*	*High*	*High*
2.	*Max*	*Max*	*High*	*Max*	8.	*Max*	*High*	*Med*	*High*
3.	*Max*	*Max*	*Med*	*Max*	9.	*Max*	*High*	*Low*	*Med*
4.	*Max*	*Max*	*Low*	*High*	10.	*Max*	*High*	*Min*	*Med*
5.	*Max*	*Max*	*Min*	*Med*	…	…	…	…	…
6.	*Max*	*High*	*Max*	*Max*	125.	*Min*	*Min*	*Min*	*Min*

The built knowledge base made it possible to form decisive decision-making rules (**Table 4.15**) regarding the implementation of one or another attacking action, taking into account the quality rank of the vulnerability exploit.

Table 4.15 Decisive decision-making rules regarding the implementation of vulnerability exploits

Rank/ *Q_{mv}*	Max	High	Med	Low	Min
Excellent	a_1	a_1	a_1	a_1	a_1
Great	a_1	a_1	a_1	a_1	a_1
Good	a_1	a_1	a_1	a_1	a_1
Normal	a_1	a_1	a_1	a_2	a_2
Average	a_1	a_1	a_2	a_2	a_2
Low	a_1	a_2	a_2	a_2	a_2
Manual	a_2	a_2	a_2	a_2	a_2

where Rank – exploit quality rank; a_1 – implement the selected vulnerability exploit; a_2 – skip the selected vulnerability exploit.

In turn, the formed decisive rules allow developing expert systems [6] for automating the decision-making process when validating identified vulnerabilities of target information systems and networks.

4.5 A METHOD OF AUTOMATIC ACTIVE ANALYSIS OF THE CORPORATE NETWORKS SECURITY BASED ON VULNERABILITIES INTELLIGENT VALIDATION

The proposed method of automatic active analysis of the security of corporate networks defines the main stages of using the developed: mathematical model for the analysis of quantitative characteristics of the vulnerability validation process, methods for analyzing the quality of work of the mechanism for validating detected vulnerabilities of the corporate network, and a method for building a fuzzy knowledge base for decision-making in the validation of software and hardware platform vulnerabilities. At the same time, the method can be divided into 4 main stages (**Fig. 4.12**): (I) preparatory stage, (II) initialization stage, (III) stage of adaptive validation of probable vulnerabilities, (IV) stage of processing and display of results (determination of the actual security level).

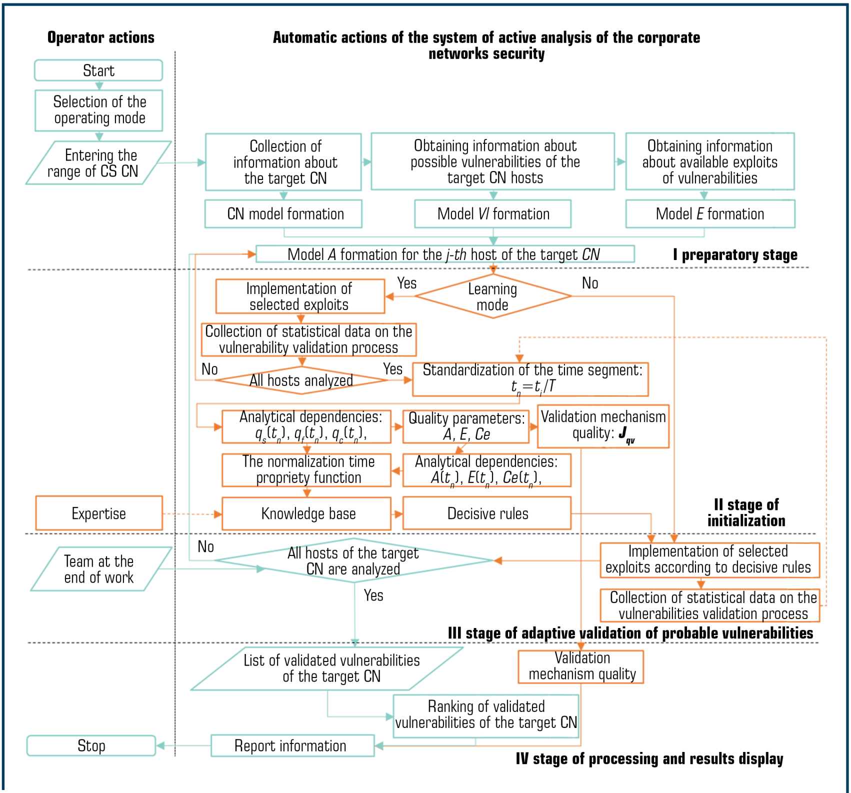
Operator actions
Automatic actions of the system of active analysis of the corporate networks security
Start
Selection of the operating mode
Entering the range of CS CN
Collection of information about the target CN
Obtaining information about possible vulnerabilities of the target CN hosts
Obtaining information about available exploits of vulnerabilities
CN model formation
Model *VI* formation
Model *E* formation
Model *A* formation for the *j-th* host of the target *CN*
I preparatory stage
Implementation of selected exploits
Yes
Learning mode
No
Collection of statistical data on the vulnerability validation process
No
All hosts analyzed
Yes
Standardization of the time segment: $t_n=t_i/T$
Analytical dependencies: $q_s(t_n)$, $q_f(t_n)$, $q_c(t_n)$,
Quality parameters: *A, E, Ce*
Validation mechanism quality: J_{qv}
The normalization time propriety function
Analytical dependencies: $A(t_n)$, $E(t_n)$, $Ce(t_n)$,
Expertise
Knowledge base
Decisive rules
II stage of initialization
No
Team at the end of work
All hosts of the target CN are analyzed
Yes
Implementation of selected exploits according to decisive rules
Collection of statistical data on the vulnerabilities validation process
III stage of adaptive validation of probable vulnerabilities
List of validated vulnerabilities of the target CN
Validation mechanism quality
Ranking of validated vulnerabilities of the target CN
Stop
Report information
IV stage of processing and results display

Fig. 4.12 Scheme of the method of automatic active analysis of the corporate network's security based on vulnerabilities intelligent validation

It should be noted that the proposed method includes two modes of operation, the first is training, during which all the above-mentioned scientific results are implemented for the construction and adaptation of the knowledge base, as well as decisive rules, that is, the training of the automatic system of active analysis of the security of corporate networks is carried out, and the second mode – directly the active analysis of the security of the corporate network.

In addition, on the basis of the analysis of approaches to conducting an active analysis of the security of corporate networks and methods and means of its automation, respectively, a number of models were formed that will allow the use of information about the target corporate network as input data, in particular information about all its components, found vulnerabilities and exploits available for their implementation.

The structure of these models is described using a theoretical-multiple approach.

Next, the sequence of steps of the method is considered in more detail:

Step 1. Gathering information about the target corporate network and forming a model *CN* according to (4.14). Any modern security scanner can be used as a source of all the necessary information, in particular information about the configuration of individual hosts of the target corporate network.

$$CN = \left\langle H, T_H, I_H \right\rangle, \tag{4.29}$$

where $H = \left\{ h_1, \ldots, h_j \right\}$ – finite set (f.s.) of hosts (nodes) of the corporate network; T_H – the type of the j_{th} host; I_H – key information about the target j_{th} host.

The host type is represented as [114]:

$$T_H = \left\{ CS, NH, M \right\}, \tag{4.30}$$

where CS – computer system; NH – network equipment; M – mobile platform.

Information about the target host:

$$I_H = \left\{ Pl, V_{Pl}, S, V_S, P \right\}, \tag{4.31}$$

where $Pl = \left\{ pl_1, \ldots, pl_j \right\}$ – f.s. of platforms (Windows, Linux, Android and other); $V_{Pl} = \left\{ v_{pl_1}, \ldots, v_{pl_j} \right\}$ – f.s. of probable platform versions; $S = \left\{ s_1, \ldots, s_j \right\}$ – f.s. of services; $V_S = \left\{ v_{s_1}, \ldots, v_{s_j} \right\}$ – f.s. of probable names and versions of the relevant services; $P = \left\{ p_1, \ldots, p_j \right\}$ – f.s. of ports on which services are running and running.

It should be noted that this description is built according to the Common Platform Enumeration (CPE) standard, which allows later, when making an assumption about the presence of vulnerabilities in the target system, to link the host configuration with data from the vulnerability database and to select appropriate exploits.

As an example: $\left\langle Linux,\ 2.6.x,\ ftp,\ \mathrm{Pr}\,oFTPD\ 1.3.1.2121 \right\rangle$.

Step 2. Obtaining information about possible vulnerabilities of the hosts of the target corporate network and forming a Vl model according to (4.32). The main source of information about vulnerabilities is open databases of vulnerabilities.

$$Vl = \left\langle ID_{Vl}, R_{Vl}, C_{Vl} \right\rangle, \tag{4.32}$$

where $ID_{Vl} = \left\{ id_{vl_1}, \ldots, id_{vl_n} \right\}$ – f.s. of identifiers of vulnerabilities presented in the CVE List; $R_{Vl} = \left\{ r_{vl_1}, \ldots, r_{vl_n} \right\}$ – f.s. of criticality assessments of vulnerabilities according to CVSS; $C_{Vl} = \left\{ c_{vl_1}, \ldots, c_{vl_n} \right\}$ – f.s. of known vulnerable configurations (identifiers issued using Common Platform Enumeration).

Step 3. Obtaining information about available exploits and forming a model E according to (4.33). Accordingly, the source of the necessary information is open and closed databases of exploits, ready-made exploit kits or integrated databases directly of the exploitation tools themselves.

$$E = \left\langle N_E, D_E, R_E, Rf_E \right\rangle, \tag{4.33}$$

where $N_E = \left\{ n_{E_1}, \ldots, n_{E_g} \right\}$ – f.s. of short names of available exploits (in fact, identifiers are represented by one or another means of exploitation); $D_E = \left\{ d_{E_1}, \ldots, d_{E_g} \right\}$ – f.s. of short descriptions of exploits (in which the name and version of the vulnerable service are indicated); $R_E = \left\{ excellent, \ldots, manual \right\}$ – f.s. of exploit quality ranks, $Rf_E = \left\{ rf_{E_1}, \ldots, rf_{E_g} \right\}$ – f.s. of links to identifiers of vulnerabilities that are implemented using an exploit.

Step 4. Selection of exploits of vulnerabilities for the jth host of the target network according to the cyber attack model A (using vulnerabilities), which is formed based on compliance with the main characteristics and vulnerabilities of the target system:

$$A = \left\{ a_1, \ldots, a_k \right\}, \tag{4.34}$$

where $a_k = F\left(\left(\left(Vl.C_{Vl}, I_H.S, I_H.V_S \right) \& \left(E.Rf_E, Vl.ID_{Vl} \right) \right) \middle| \left(E.D_E, I_H.S, I_H.V_S \right) \right)$.

Step 5 (in learning mode). The implementation of selected exploits and the collection of statistical data on the basic characteristics of the target host vulnerability validation process are carried out one by one. Based on the collected data, the quality of the vulnerability validation mechanism is evaluated according to the following sub-steps:

5.1. Standardization of the time segment for validation of host vulnerabilities of the target corporate network according to (4.8):

$$t_n = t_i / T;$$

5.2. Obtaining analytical dependencies for basic characteristics $\left(q_s, q_f, q_c \right)$ of vulnerability validation process (4.30):

$$q_s(t_n) = \sum_{i=0}^{n} q_s(t_n^{(i)}) b_{k,n}(t_n),\ q_f(t_n) = \sum_{i=0}^{n} q_f(t_n^{(i)}) b_{k,n}(t_n),\ q_c(t_n) = \sum_{i=0}^{n} q_c(t_n^{(i)}) b_{k,n}(t_n).$$

5.3. Calculation of quality indicators of the vulnerability validation mechanism (4.20)–(4.22): A – accuracy, E – error and Ce – critical error:

$$A=\frac{\int_0^1 q_s(\theta)d\theta}{\int_0^1 (q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}\ ;\quad E=\frac{\int_0^1 q_f(\theta)d\theta}{\int_0^1 (q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}\ ;$$

$$Ce=\frac{\int_0^1 q_c(\theta)d\theta}{\int_0^1 (q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}.$$

5.4. Evaluation of the quality of the validation mechanism according to the single integral quality indicator (4.23):

$$J_{qv}=\frac{A}{E}-Ce.$$

5.5. Obtaining analytical dependencies for the quality indicators of the vulnerability validation mechanism $A(t_n)$, $E(t_n)$, $Ce(t_n)$ (4.28)–(4.30):

$$A\left(t_n\right)=\frac{\int_0^{t_n} q_s(\theta)d\theta}{\int_0^{t_n} (q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta};\quad E\left(t_n\right)=\frac{\int_0^{t_n} q_f(\theta)d\theta}{\int_0^{t_n} (q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta};$$

$$Ce\left(t_n\right)=\frac{\int_0^{t_n} q_c(\theta)d\theta}{\int_0^{t_n} (q_s(\theta)+q_f(\theta)+q_c(\theta))d\theta}.$$

5.6. Construction of the normalization time membership function.

5.7. Formation of the knowledge base and decisive decision-making rules regarding the implementation of the attacking action in the form of (4.28) logical conclusion

$$R\left(i\right):IF\left(A_i\in T\ \&\ E_i\in T\ \&\ Ce_i\in T\right)THEN\left(Q_{mv_i}\in T\right),\ i=\overline{1,k},$$

having previously determined the required number of rules according to (4.27): $N_{\max}=l_1\cdot l_2\cdot\ldots\cdot l_n$.

Step 5 (in active analysis mode). Implementation of selected exploits in accordance with decisive decision-making rules and collection of statistical data regarding the vulnerability validation process, based on which, in accordance with Steps 5.1–5.4, the quality assessment of the vulnerability validation mechanism is carried out.

Step 6 (in active analysis mode). Ranking of validated vulnerabilities of the target corporate network and generating a report of the conducted active security analysis. After analyzing all the hosts of the target corporate network, a general list of validated, i.e., confirmed vulnerabilities is formed, at the same time, they are ranked according to the level of their criticality, which is determined by the CVSS base assessment and the level of prevalence of this vulnerability L_{v_i} in the corporate network according to the expression (4.35), otherwise, there is a return to Step 5 (in active analysis mode). As a result, a report of the conducted active security analysis is generated, containing a ranked list of confirmed vulnerabilities of the target corporate network in descending order, from vulnerabilities with the highest levels of criticality and prevalence to vulnerabilities with the lowest levels, as well as the quality level of the mechanism for validating the identified vulnerabilities.

$$L_{v_i} = \frac{h_v}{h_T} \cdot 100, \tag{4.35}$$

where h_v – number of vulnerable hosts to the validated vulnerability *v*; h_T – the total number of analyzed hosts of the target corporate network, $h_T > 0$.

Based on the report, the expert decides on the necessity of retraining the automated system of active security analysis, as well as on the priority elimination of one or another validated vulnerability.